

Closed-loop quantitative verification of rate-adaptive pacemakers

NICOLA PAOLETTI, University of Oxford, Department of Computer Science

ANDREA PATANÈ, University of Catania, Department of Mathematics and Computer Science

MARTA KWIATKOWSKA, University of Oxford, Department of Computer Science

Rate-adaptive pacemakers are cardiac devices able to automatically adjust the pacing rate in patients with chronotropic incompetence, i.e. whose heart is unable to provide an adequate rate at increasing levels of physical, mental or emotional activity. These devices work by processing data from physiological sensors in order to detect the patient's activity and update the pacing rate accordingly. Rate-adaptation parameters depend on many patient-specific factors, and effective personalisation of such treatments can only be achieved through extensive exercise testing, which is normally intolerable for a cardiac patient. In this work, we introduce a data-driven and model-based approach for the automated verification of rate-adaptive pacemakers and formal analysis of personalised treatments. To this purpose, we develop a novel dual-sensor pacemaker model where the adaptive rate is computed by blending information from an accelerometer, and a metabolic sensor based on the QT interval. Our approach enables personalisation through the estimation of heart model parameters from patient data (electrocardiogram), and closed-loop analysis through the online generation of synthetic, model-based QT intervals and acceleration signals. In addition to personalisation, we also support the derivation of models able to account for the varied characteristics of a virtual patient population, thus enabling safety verification of the device. To capture the probabilistic and non-linear dynamics of the heart, we define a probabilistic extension of timed I/O automata with data and employ statistical model checking for quantitative verification of rate modulation. We evaluate our rate-adaptive pacemaker design on three subjects and a pool of virtual patients, demonstrating the potential of our approach to provide rigorous, quantitative insights into the closed-loop behaviour of the device under different exercise levels and heart conditions.

ACM Reference Format:

Nicola Paoletti, Andrea Patanè and Marta Kwiatkowska, 2017. Closed-loop quantitative verification of rate-adaptive pacemakers. *ACM Trans. Cyber-Phys. Syst.* V, N, Article A (January YYYY), 45 pages.
ttDOI: <http://dx.doi.org/10.1145/0000000.0000000>

1. INTRODUCTION

Cardiac pacemakers are small, life-saving medical devices that stimulate the heart tissue through artificial electrical impulses in order to maintain an adequate heart rhythm in patients with cardiac conditions. Current pacemakers include mechanisms for rate adaptation, i.e., for automatically adjusting the frequency of electrical stimuli, or *pacing rate*, depending on the levels of physical, mental or emotional stress of the patient. This feature is crucial to ensure a good quality of life for patients with chronotropic incompetence, i.e. whose heart is unable to provide by itself a rate commensurate with the ideal metabolic demand. Rate-adaptive pacemakers work by processing data

This work is supported by the ERC AdG VERIWARE, the ERC PoC VERIPACE and AFFECTech ITN. Nicola Paoletti is partially supported by the NSF Grant CNS-1446832. Andrea Patanè carried out the research during his internship at the University of Oxford funded by VERIWARE and VERIPACE projects.

Author's addresses: N. Paoletti, (Current address) Department of Computer Science, Stony Brook University, NY 11794, USA; A. Patanè and M. Kwiatkowska, Department of Computer Science, University of Oxford, OX1 3QD, UK.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© YYYY ACM. 2378-962X/YYYY/01-ARTA \$15.00

ttDOI: <http://dx.doi.org/10.1145/0000000.0000000>

from physiological sensors in order to detect the patient's activity and update the pacing rate accordingly [Dell'Orto et al. 2004].

Rate adaptation and device safety have been investigated in a number of clinical studies (see e.g. [Candinas et al. 1997; Lamas et al. 2007; Abi-Samra et al. 2013]), but such studies require extensive testing under varying levels of physical activity, including maximal exercise tests (e.g. treadmill), which is often intolerable for cardiac patients, especially in the elderly. Similarly, exercise testing is necessary for effective personalisation of the treatment, given that rate-adaptation parameters depend on many patient-specific factors such as age, lifestyle and tolerance to rapid pacing.

An alternative approach put forward in this paper is rigorous *in silico* analysis of the designs, while accounting for the specific electrophysiological characteristics of the patient and how these characteristics vary at different activity levels.

We introduce a data-driven, model-based approach for the automated, closed-loop verification of rate-adaptive pacemakers. We consider the VVIR¹ pacemaker design, that is, a single-chamber pacemaker that senses and paces the (right) ventricle, and supports rate adaptation. To this purpose, we develop a novel dual-sensor VVIR pacemaker model where the adaptive rate is computed by a so-called *sensor blending algorithm* that combines information coming from two sensors: an accelerometer, and a metabolic sensor based on the *QT interval (QTI)*, i.e. the time needed for ventricle depolarisation and re-polarisation. QTIs are extracted from the pacemaker electrogram, i.e. the electrical signal recorded by the electrode.

The proposed sensor blending algorithm exploits the strengths of both sensors: the accelerometer has a quick response to exercise but is inaccurate, while the QTI provides a slow but very accurate response. The QT sensor relies on the fact that physical and mental stress shortens the QTI. In particular, there is a natural, patient-specific relationship between QTI and heart rate (HR) in healthy subjects. The blending algorithm leverages this relationship through the estimation of regression functions between QTI and HR, used to predict an adequate pacing rate based on the QTI.

A key feature of our approach is that it supports rigorous analysis of patient-specific treatments through the estimation of personalised heart models from electrocardiogram (ECG) data [Barbot et al. 2015a]. Importantly, our estimation method can also handle ECG data from multiple patients, by combining the parameters (i.e., probability distributions extracted from data) across all patients in a given population. In this way we can estimate models able to fully capture the varied characteristics of the virtual patient population, thus enabling safety verification.

To account for the probabilistic nature of cardiac dynamics and ECG features detected from data, heart and pacemaker models are specified in a probabilistic extension of timed I/O automata with priorities and data [Kwiatkowska et al. 2015; Barbot et al. 2016]. This formalism can represent networked systems with real-time constraints and discrete control actions, characteristic of the pacemaker, as well as the probabilistic, hybrid and non-linear dynamics of the cardiac conduction system. For formal verification, we resort to statistical model checking (SMC) [Ballarini et al. 2015], an approximate verification technique based on the statistical inference of quantitative properties from a set of executions, well-suited to our formalism for which no precise model checking method exists.

A crucial requirement for the analysis *in silico* of rate adaptation is closing the gap between the heart simulation and the physiological sensor data used by the pacemaker to update the pacing rate, in this way modelling the characteristic feedback interaction between the physiological signal used and the heart dynamics [Ellenbogen et al. 2016,

¹The code VVIR is based on the standard NASPE/BPEG nomenclature [Bernstein et al. 2002].

ch. 5]. To this aim, we introduce a *closed-loop design* for rate adaptation that builds on the online generation of model-based QTIs and synthetic accelerometer signals.

We evaluate our approach over four different heart model parametrisation, respectively describing three distinct virtual patients and a pool of ten virtual patients, and employ SMC for quantitative verification in a variety of scenarios. These include the analysis of rate control under realistic exercises curves and clinical stress tests, and under increasing degrees of chronotropic incompetence (i.e. worsening heart conditions). Results demonstrate that our approach can provide rigorous and quantitative insights into the safety and performance of the device, enabling both personalised and population-level analysis of cardiac therapies, with the potential to drastically reduce the need for actual exercise testing with cardiac patients.

Paper structure. In the remainder of the introduction we discuss related work. In Section 2 we introduce our main modelling formalism. Statistical model checking is presented in Section 3. In Section 4, we introduce our closed-loop heart-pacemaker model, the virtual patients used for our analyses, and the methods for the processing and generation of physiological sensor data. In Section 5, we report simulation and verification results. Conclusions are given in Section 6.

1.1. Related work

This paper is a major extension of [Kwiatkowska et al. 2014], which introduces the first formal and executable model of the VVIR pacemaker. In that work, the authors realise a single-sensor and open-loop design for rate-adaptation, i.e., such that the pacemaker detects activity from static, offline physiological data and just from one sensor (the QT). In contrast, our work implements a dual-sensor and closed-loop design, which provides faster and more accurate activity detection through the combination of QT and accelerometer sensors, and uses dynamically generated physiological information. Compared to the open-loop design, in Section 5.2.2 we show that the closed-loop design can reproduce with superior accuracy phenomena of sensor-induced endless-loop tachycardia, triggered by the feedback interaction between the pacemaker, the heart and the resulting physiological sensor data. Furthermore, in this work we employ the probabilistic heart model of [Barbot et al. 2015a] that supports personalisation from patient data. This improves on [Kwiatkowska et al. 2014], where the authors use the deterministic heart model of Ye et al. [2005], which, albeit providing more detailed action potential dynamics, does not support personalisation.

Formal modelling and analysis of cardiac dynamics and devices is a very active field of research (see [Macedo et al. 2008; Gomes and Oliveira 2009; Bartocci et al. 2009; Tuan et al. 2010; Grosu et al. 2011; Jiang et al. 2012; Pajic et al. 2012; Méry et al. 2014; Chen et al. 2014; Kwiatkowska et al. 2015; Barker et al. 2015; Ai et al. 2017] for a non-exhaustive list of references). However, modelling of the rate-adaptive pacemaker has received limited attention so far. Besides [Kwiatkowska et al. 2014] (discussed above), to the best of our knowledge, the only other work on rate adaptation is that of Méry and Singh [2009], where the authors consider a single-sensor accelerometer-based pacemaker, but only provide a high-level, non-executable specification of the rate modulation mechanism. Related research also includes the definition of algorithms for integrating signals from multiple physiological sensors (see e.g. [Shin et al. 2001; Amigoni et al. 2006]).

A complementary technique is model-based testing (MBT), employed by Ai et al. [2016] for the validation of cardiac devices. Similarly to SMC, MBT relies on evaluating executions of the system, but its focus is more on guaranteeing high coverage and efficient bug finding rather than deriving quantitative measures (e.g. satisfaction probability) for a given property as in SMC.

2. MODELLING FORMALISM

We introduce our main modelling language, *Probabilistic Timed I/O Automata with priorities and data (PTIOA)*, which extends Timed I/O Automata with priorities and data (TIOA) [Kwiatkowska et al. 2015; Barbot et al. 2016] with probabilistic delays.

TIOAs are well suited for modelling networked systems with real-time constraints and discrete control actions, as well as hybrid dynamics through continuous variables and non-linear update functions, and thus provide an adequate level of representation for cardiac pacemaker models in closed loop with hybrid models of the human heart [Kwiatkowska et al. 2015; Barbot et al. 2016]. Importantly, this kind of automata can be expressed as MATLAB Stateflow diagrams (as shown by Barbot et al. [2015a; 2016]), thus enabling effective tool support.

In addition, probabilistic features are essential to achieve personalisation of heart models from data, as we will see in Section 4.2. By supporting arbitrary distributions to specify time delays, PTIOAs allow describing patient's features accounting for statistical information from the data.

We now provide a formal account of PTIOA and its semantics. In the following, we denote with $dist(A)$ the set of probability distributions whose support is A .

Variables. A PTIOA includes a set of variables $V = \mathcal{X} \cup \mathcal{D}$, where \mathcal{X} and \mathcal{D} are the set of *clocks* and *data*, respectively. Clocks record the passage of time, while data variables can be updated to arbitrary real values. A variable valuation $\eta : V \rightarrow \mathbb{R}$ is a function that maps data variables to the reals and clocks to the non-negative reals. For a set \mathcal{Y} , we denote with $\mathcal{V}(\mathcal{Y})$ the set of all valuations over \mathcal{Y} . For $\eta \in \mathcal{V}(V)$, $\eta^{\mathcal{X}} \in \mathcal{V}(\mathcal{X})$ and $\eta^{\mathcal{D}} \in \mathcal{V}(\mathcal{D})$ denote the valuation η restricted to clocks and data variables, respectively. The valuation η after time $t \in \mathbb{R}_{\geq 0}$ has elapsed is denoted with $\eta+t$ and is such that $(\eta+t)(v) = \eta(v) + t$ if $v \in \mathcal{X}$ and $(\eta+t)(v) = \eta(v)$ otherwise. This captures the fact that all clocks proceed at the same speed and data variables are not affected by the passage of time.

Updates. In a PTIOA, variable valuations are manipulated through update functions. The update of a set of variables $V' \subseteq V$ is a real-valued function $r : V' \times \mathcal{V}(V) \rightarrow \mathbb{R}$. A valuation $\eta \in \mathcal{V}(V)$ is changed by the update function r into the valuation $\eta[r] = \{v \mapsto r(v, \eta) \mid v \in V'\} \cup \{v \mapsto \eta(v) \mid v \notin V'\}$ that applies the update r to the variables in V' and leaves the others unchanged. We denote with \mathcal{R} the set of update functions.

Guards. We denote with $\mathcal{B}(V)$ the set of guard constraints over V , which describe the probabilistic firing conditions of automata edges. Specifically, guard constraints specify probabilistic delays of the form $g = \bigwedge_i x_i \geq t_i$, where $x_i \in \mathcal{X}$ is a clock and $t_i \sim D_i$ is a non-negative random variable² describing the delay, and distributed according to the probability density function (PDF) $D_i : \mathcal{V}(\mathcal{D}) \rightarrow dist(\mathbb{R}_{\geq 0})$, which possibly depends on data variable valuations. Where possible, we alternatively use the shortcut $g = \bigwedge_i x_i \geq D_i$ in place of $g = \bigwedge_i x_i \geq t_i$, where $t_i \sim D_i$. Let $D_i(\eta^{\mathcal{D}})$ be the PDF under valuation $\eta^{\mathcal{D}}$. We denote with $D_i^{\leq}(\eta^{\mathcal{D}})$ the corresponding cumulative distribution function, i.e. such that for all $t \in \mathbb{R}_{\geq 0}$, $D_i^{\leq}(\eta^{\mathcal{D}})(t) = \int_{-\infty}^t D_i(\eta^{\mathcal{D}})(u) du$.

Remark 2.1 (Independence). We assume that the probabilistic delays in the guards of a PTIOA are mutually independent.

²Not to be confused with the data variables.

Under the above assumption, the *satisfaction probability* of guard g under valuation $\eta \in \mathcal{V}(V)$ can be computed by factorising the joint probability as follows:

$$Pr(\eta \models g) = \prod_i D_i^{\leq}(\eta^{\mathcal{D}})(\eta(x_i)). \quad (1)$$

The random waiting time, $t(g, \eta)$, for g to be satisfied under $\eta \in \mathcal{V}(V)$ is given by

$$t(g, \eta) = \max(0, \max\{t_i - \eta(x_i)\}_i) \quad (2)$$

Deterministic delays can be expressed by choosing a Dirac distribution for $D_i(\eta^{\mathcal{D}})$. Note that, since D_i can depend only on data variables, it is not affected by the passage of time, i.e. $\forall t \geq 0. \forall \eta \in \mathcal{V}(V). D_i((\eta + t)^{\mathcal{D}}) = D_i(\eta^{\mathcal{D}})$.

Example 2.2 (Satisfaction probability). Let $g = x \geq \mathcal{U}(1, 2) \wedge y \geq \mathcal{U}(0, 2)$, where $\mathcal{U}(a, b)$ is the uniform distribution on $[a, b]$, characterised by the cumulative distribution function $\mathcal{U}^{\leq}(a, b)(x) = \{0 \text{ if } x < a, \frac{x-a}{b-a} \text{ if } x \in [a, b], 1 \text{ if } x > b\}$.

Following Equation 1, the probability that valuation $\eta = \{x \mapsto 0, y \mapsto 0\}$ satisfies g is 0, since $\mathcal{U}^{\leq}(1, 2)(0) = 0$ and $\mathcal{U}^{\leq}(0, 2)(0) = 0$. The same holds for valuation $\eta' = \{x \mapsto 1, y \mapsto 1.5\}$, since $\mathcal{U}^{\leq}(1, 2)(1) = 0$. If from η' we let 0.5 time units pass, the corresponding valuation $\eta' + 0.5 = \{x \mapsto 1.5, y \mapsto 2\}$ yields a satisfaction probability $\mathcal{U}^{\leq}(1, 2)(1.5) \cdot \mathcal{U}^{\leq}(1, 2)(2) = \frac{1}{2} \cdot 1$.

Remark 2.3 (Discrete distributions). To simplify the presentation, probabilistic delays are defined over continuous distributions, even if the above definitions can be easily adapted to support discrete distributions.

Actions. Let A be a set of action symbols. We consider the set of actions $\Sigma = \Sigma_{\text{in}} \cup \Sigma_{\text{out}}$, composed of input actions $\Sigma_{\text{in}} = \{?\alpha \mid \alpha \in A\}$ and output actions $\Sigma_{\text{out}} = \{!\alpha \mid \alpha \in A\}$. Each edge has an action associated, describing the event performed by that edge. As we shall see, automata components communicate by synchronising on matching actions. For $a \in \Sigma_{\text{in}}$ and $b \in \Sigma_{\text{out}}$, we say that a *matches* b (or vice-versa) if $a = ?\alpha$ and $b = !\alpha$ for some $\alpha \in A$.

Definition 2.4 (PTIOA). A *probabilistic timed I/O automaton (PTIOA)* with *priorities and data* $\mathcal{A} = (\mathcal{X}, \mathcal{D}, Q, q_0, \Sigma, \rightarrow)$ consists of:

- A finite set of *clocks* \mathcal{X} and *data variables* \mathcal{D} .
- A finite set of *locations* Q , with *initial location* $q_0 \in Q$.
- A finite sets of *input and output actions* $\Sigma = \Sigma_{\text{in}} \cup \Sigma_{\text{out}}$.
- A finite set of *edges* $\rightarrow \subseteq Q \times \Sigma \times \mathbb{N} \times \mathcal{B}(V) \times \mathcal{R} \times Q$. 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 , an *update* r and a *target* q' . Priorities define a total ordering of the edges out of any location, and are such that lower pr values imply higher priorities.

Note that, unlike classical timed automata [Alur 1999], PTIOA locations do not include invariants in order to avoid time non-determinism: as explained in Section 2.2, an enabled edge is fired as soon as possible, in a manner compatible with priority ordering, and no additional time can pass.

In addition to probabilistic dynamics, we remark that PTIOAs can express hybrid dynamics, and specifically hybrid automata whose differential equations admit explicit solutions that can be effectively computed. Indeed, we can use non-linear update functions to encode such explicit solutions.

2.1. Networks of PTIOAs

To facilitate modular designs, PTIOAs are able to synchronise on matching input and output actions, thus forming *networks of communicating automata*.

Definition 2.5 (Network of PTIOAs). A network of PTIOAs with m components is a tuple $\mathcal{N} = (\{\mathcal{A}^1, \dots, \mathcal{A}^m\}, \mathcal{X}, \mathcal{D}, \Sigma)$, where

- for $j = 1, \dots, m$, $\mathcal{A}^j = (\mathcal{X}, \mathcal{D}, Q^j, q_0^j, \Sigma, \rightarrow^j)$ is a PTIOA,
- \mathcal{X} , \mathcal{D} and Σ are the common sets of clocks, data variables and actions, respectively.

We define the set of *network modes* by $\vec{Q} = Q^1 \times \dots \times Q^m$, with initial mode $\vec{q}_0 = (q_0^1, \dots, q_0^m)$ and the initial variable valuation η_0 . For mode $\vec{q} = (q^1, \dots, q^m) \in \vec{Q}$ and $j \in \{1, \dots, m\}$, we denote with $\vec{q}_j = q^j$ the j -th component of \vec{q} , that is, the location of the j -th automaton. A *state of the network* is a pair (\vec{q}, η) , where $\vec{q} \in \vec{Q}$ is the vector of active locations and $\eta \in \mathcal{V}(V)$ is the variable valuation.

We use the notation $\mathcal{N} = (\{\mathcal{A}^1, \dots, \mathcal{A}^m\}, \mathcal{X}, \mathcal{D}, \Sigma)$ to stress the fact that \mathcal{X} , \mathcal{D} and Σ are shared across $\mathcal{A}^1, \dots, \mathcal{A}^m$.

Example 2.6 (PTIOA network for SA node and atrium). Figure 1 shows a sub-network of PTIOAs of the heart model, illustrated in Section 4.2. The automaton of Fig. 1a describes the sinoatrial (SA) node, the component of the right atrium that generates intrinsic electrical stimuli. Fig. 1b depicts the PTIOA for the atrium.

In the SA node component, after the firing period of the SA node has elapsed, modelled by the normally distributed delay $\text{SA_d} \sim \mathcal{N}(1, 0.1)$ and guard $x \geq \text{SA_d}$, the SA node transitions from the initial location Wait4ABeat to Wait4ASynch , performing an output action $!A_{\text{beat}}$ to notify the other components of the network that the stimulus occurred. Data variable a_dV maintains the action potential of the signal and is updated to SA_dV , which is specific to stimuli generated from the SA node. In Wait4ASynch , the automaton waits until it can synchronise on action NextAtrBeat , at which point it resets the clock x and returns to location Wait4ABeat .

In the atrium component, after a refractory period where the atrium cannot be stimulated, governed by the uniformly distributed delay $\text{Atr_refrD} \sim \mathcal{U}(0.04, 0.06)$, the atrium becomes excitable and can receive three types of impulses: an intrinsic stimulus from the SA node, through synchronisation on A_{beat} ; paced stimulus from the pacemaker (AP); or a retrograde signal from the ventricles (AtrRetroln). 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 $!A_{\text{trAnteOut}}$. When the stimulus is not artificial, the component emits an A_{get} action for the pacemaker to read. Before reaching the Refractory location again, an intermediate action is required ($!A_{\text{NextAtrBeat}}$) to synchronise the SA node component.

Note that, while the VVIR pacemaker under study stimulates only the ventricle, the Atrium component supports atrial pacing, and thus can be also connected to models of atrial or dual-chamber pacemakers.

In this example, we arbitrarily chose the random delay SA_d to produce a heart period with mean 1 s (i.e. rate of 60 BPM). In our experiments, however, SA_d depends on the ideal rate demand, and thus on the kind of activity performed, as explained in Section 4.2.3. On the other hand, the atrial refractory period Atr_refrD is based on the original parameterisation of Barbot et al. [2015b].

2.2. Semantics

Enabled components and synchronisation. Let $\mathcal{N} = (\{\mathcal{A}^1, \dots, \mathcal{A}^m\}, \mathcal{X}, \mathcal{D}, \Sigma)$ be a PTIOA network and (\vec{q}, η) be its current state. We say that a PTIOA component \mathcal{A}^j

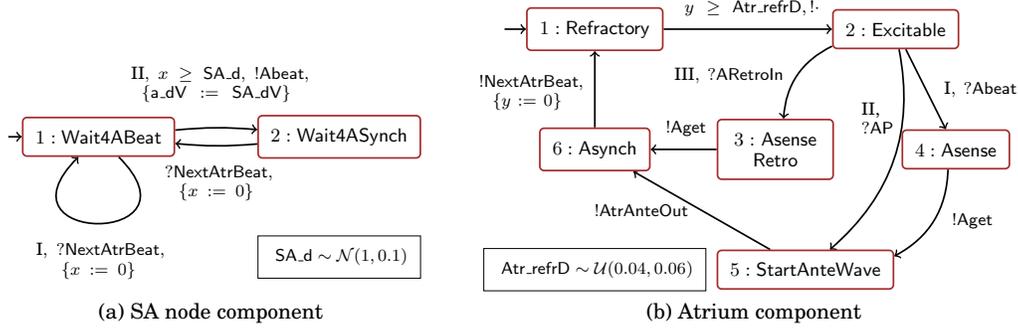


Fig. 1: Portion of the PTIOA network of the heart model. Edge priorities are indicated with Roman numbers (I,II, III). Empty updates, trivially true guards and priorities of edges with no alternative choices are omitted for clarity. Output action !: is used when no synchronisation is sought for an edge.

of \mathcal{N} is enabled from (\vec{q}, η) if, from its current location \vec{q}_j , it has at least one enabled edge. An output edge is enabled when the associated guard is satisfied under valuation η^3 , while an input edge is enabled when its guard holds under η and, at the same time, it can synchronise with a matching output action fired by another component of the network. This means that, unlike output edges, input edges can fire only by synchronising with a matching output action, i.e. an edge labelled with $!\alpha$ can fire without other components firing $?\alpha$, but not vice-versa. In turn, we allow for multi-party synchronisation, that is, multiple input edges can synchronise with a single matching output edge.

Product PTIOA. Formally, the semantics of a PTIOA network is defined over the corresponding *product PTIOA* (see Appendix A for details on its construction). The product PTIOA of a network \mathcal{N} is a PTIOA $\mathcal{N}_{\otimes} = (\mathcal{X}, \mathcal{D}, \vec{Q}, \vec{q}_0, \mathcal{P}(\Sigma_{\text{out}}), \rightarrow)$ where locations correspond to the set of network modes \vec{Q} , and edges in \rightarrow are defined by applying the above synchronisation rules. To reflect the simultaneous firing of multiple components, an edge $e = (\vec{s}, a, pr, g, r, \vec{t}) \in \rightarrow$ is characterised by: an action $a \in \mathcal{P}(\Sigma_{\text{out}})$ given by the set of output actions fired by the enabled components; a vector of priorities $pr = (pr^1, \dots, pr^m) \in \mathbb{N}^m$, where, for $j = 1, \dots, m$, pr^j is the priority of the edge fired by \mathcal{A}^j or $+\infty$ if \mathcal{A}^j is not enabled; a guard g and an update r obtained by combining guards and updates, respectively, of the fired edges.

Below, we describe the semantics of PTIOA networks in a way that facilitates their encoding into Discrete Event Stochastic Processes [Ballarini et al. 2015], for which we can leverage efficient statistical model checking algorithms, as described in Section 3.

Semantics. Given a PTIOA network \mathcal{N} and its product PTIOA $\mathcal{N}_{\otimes} = (\mathcal{X}, \mathcal{D}, \vec{Q}, \vec{q}_0, \mathcal{P}(\Sigma_{\text{out}}), \rightarrow)$, the semantics of \mathcal{N} is described by timed paths of the form $\rho = (\vec{q}_0, \eta_0) \xrightarrow{e_0, t_0} (\vec{q}_1, \eta_1) \xrightarrow{e_1, t_1} \dots$, where, for each i , $\rho[i] = (\vec{q}_i, \eta_i)$ is a state of the network, t_i is the time spent in that state and $e_i \in \rightarrow$ is the edge fired in \mathcal{N}_{\otimes} .

Given the probabilistic nature of the model, $\rho[i] = (\vec{q}_i, \eta_i)$, t_i and e_i are families of random variables, characterised for each step i as follows (see Appendix A for a formal definition). Let $t(e, \eta_i)$ be a random variable describing the waiting time for

³To simplify the presentation, here guard satisfaction is assumed under some realisation (sampling) of the random delays associated to the guard constraint.

$e \in \rightarrow$ to be enabled under valuation η_i (see Equation 2). Then, the waiting time t_i is the shortest waiting time among the outgoing edges of \vec{q}_i , and $e_i \in \rightarrow$ is the edge with shortest waiting time and highest priority. Finally, the next state $\rho[i+1] = (\vec{q}_{i+1}, \eta_{i+1})$ is determined by the target location and the update of e_i .

3. STATISTICAL MODEL CHECKING WITH HASL PROPERTIES

For the verification of PTIOA models we employ statistical model checking (SMC), a technique for the verification of probabilistic properties based on simulating the system for finitely many runs [Younes et al. 2006]. This provides an effective, yet approximate, verification method for PTIOA networks, for which no precise (numerical) model checking method exists. As we shall see, another advantage of SMC is that it supports rich specification languages, including quantitative and automata-based properties.

In this work, we consider properties specified in the *Hybrid Automata Stochastic Language (HASL)* [Ballarini et al. 2015], implemented in the statistical model checker COSMOS⁴. An HASL formula consists of two parts:

- (1) a *Linear Hybrid Automaton (LHA)* that keeps a set of data variables and synchronises with executions of the system, in our case paths of a PTIOA network; and
- (2) the actual quantity to evaluate, given as an expression over the data variables of the LHA.

In the following, we give a brief overview of these two components. For a detailed explanation, we refer to [Ballarini et al. 2015] and Appendix B.

Linear Hybrid Automata. We outline the main differences between LHAs and PTIOAs (introduced in the previous section). Since an LHA is meant to synchronise with the paths of the PTIOA network under analysis, it also contains a *final location* that determines when to terminate and accept the current path. If at any point the LHA cannot synchronise with the PTIOA network, the path is terminated and rejected. In particular, an LHA edge can either synchronise with a set of PTIOA actions⁵ or proceed autonomously, i.e. it can fire without synchronisation (denoted by action #). Unlike PTIOAs, LHA guards are deterministic and are specified as linear constraints over LHA variables. In addition to update functions, the evolution of LHA variables is controlled by *flows*, i.e. linear functions describing their change rate at each location.

HASL expressions. We introduce the HASL fragment of interest for our properties (see Appendix B for the full syntax). A HASL expression is of the form $E[Y]$ and allows for evaluating the expectation of a so-called *path variable* $Y = \text{LAST}(y)$, i.e. a random variable denoting the value of y at the final state of the synchronised execution.

The term y is an arithmetic expression over LHA variables, built using the operations $\{+, \cdot, -, /\}$. A derived HASL expression is $\text{PDF}(Y, h, a, b)$ that computes a discrete approximation of the probability density function (PDF) of path variable Y in the interval $[a, b]$ and using h subdomains.

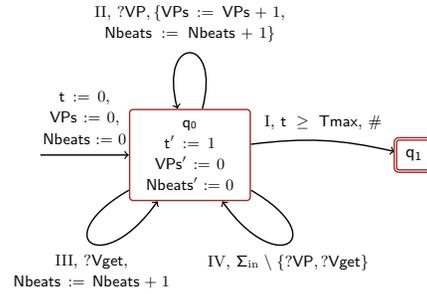


Fig. 2: LHA for estimating the fraction of paced beats. Final locations are denoted with double-bordered boxes.

⁴Available at <http://www.lsv.ens-cachan.fr/Software/cosmos/>

⁵Since they synchronise with the output actions fired by the PTIOA network, all LHA actions are implicitly input actions.

In the COSMOS tool, statistical model checking of a HASL expression Z is based on *confidence interval estimation*: given a confidence level $\alpha \in (0, 1)$ and number of executions n , the method returns a confidence interval CI_{μ_Z} for the expected value of Z , μ_Z , such that the actual value of μ_Z lies within CI_{μ_Z} with probability $1 - \alpha$. In particular, the midpoint of CI_{μ_Z} provides an estimator for μ_Z and is computed as the mean of the values of Z obtained from the above n executions.

Example 3.1 (Fraction of paced beats). Consider the LHA in Figure 2. Its purpose is to record the number of paced and total ventricular beats occurring within T_{\max} time units. These quantities are kept respectively in data variables VPs and $Nbeats$. Clock t stores the current time. All variables are initialised to 0. Flows indicate that VPs and $Nbeats$ have associated a change rate of 0, while t has a change rate of 1. The LHA is intended to synchronise with the heart and pacemaker network presented in Section 4.2, where intrinsic and paced ventricular impulses are implemented through actions $Vget$ and VP , respectively.

The automaton has two locations: an initial location q_0 and a final location q_1 , reached from q_0 with an autonomous edge as soon as $t \geq T_{\max}$.

Consider the following path, describing the antegrade propagation of the action potential from the atrium to the ventricle, as shown in the high-level schema of the heart model of Figure 4⁶:

$$\begin{aligned} (\bar{q}_0, \eta_0) &\xrightarrow{!Abeat, 0.9} (\bar{q}_1, \eta_1) \xrightarrow{!Aget, 0} (\bar{q}_2, \eta_2) \xrightarrow{!AtrAnteOut, 0} (\bar{q}_3, \eta_3) \xrightarrow{!NextAtrBeat, 0} (\bar{q}_4, \eta_4) \xrightarrow{!AVJAnteln, 0.03} \dots \\ \dots (\bar{q}_5, \eta_5) &\xrightarrow{!AVJAnteOut, 0.02} (\bar{q}_6, \eta_6) \xrightarrow{!AVVAnteln, 0.04} (\bar{q}_7, \eta_7) \xrightarrow{!VtrAnteln, 0.05} (\bar{q}_8, \eta_8) \xrightarrow{!Vget, 0} (\bar{q}_9, \eta_9) \end{aligned}$$

Let $T_{\max} = 1.5$. From the initial location q_0 , the LHA synchronises with path transitions $\xrightarrow{e_0, t_0} \dots \xrightarrow{e_7, t_7}$ by firing the loop labelled with $\Sigma_{in} \setminus \{?VP, ?Vget\}$. This edge has the least priority, a trivially true guard, empty update and can match any output action fired by the PTIOA network apart from $?VP$ and $?Vget$. Without such an edge, the LHA would fail to synchronise at the first transition (labelled with $\{!Abeat\}$), and the path would be immediately rejected. Along transition $\xrightarrow{e_8, t_8}$, the PTIOA network fires $\{!Vget\}$, leading to a synchronisation on the LHA edge labelled with $?Vget$. The resulting variable valuation is $t = 1.04$, $Nbeats = 1$ and $VPs = 0$.

Consider the continuation $\rho = \dots (\bar{q}_9, \eta_9) \xrightarrow{!VP, 0.3} (\bar{q}_{10}, \eta_{10})$. Here, the LHA synchronises with $\xrightarrow{e_9, t_9}$ on the LHA edge labelled with $?VP$, leading to valuation $t = 1.34$, $Nbeats = 2$ and $VPs = 1$. After time $T_{\max} - t = 0.16$, the LHA can fire the autonomous edge to the final location. Assuming no $!VP$ or $!Vget$ is emitted by the PTIOA network during this period, the final valuation is $t = 1.5$, $Nbeats = 2$ and $VPs = 1$.

Let $y = \frac{VPs}{Nbeats}$ denote the fraction of paced beat over the total number of beats. For path ρ , $LAST(y) = \frac{1}{2}$, because $Nbeats = 2$ and $VPs = 1$ at the final state. A property that we will analyse in Section 5 is the PDF of the fraction of paced beats at the end of the execution, i.e. after time T_{\max} :

$$\phi_{PDF(VP)} = PDF\left(LAST\left(\frac{VPs}{Nbeats}\right), h, a, b\right), \quad (3)$$

where $a = 0$ and $b = 1$, given that the ratio $\frac{VPs}{Nbeats}$ ranges in the interval $[0, 1]$. We further set $h = 0.01$, which corresponds to dividing the domain of the PDF in 100 sub-domains.

⁶With abuse of notation, we denote transition $\xrightarrow{e, t}$ by $\xrightarrow{a(e), t}$, where we replace the edge e with the corresponding set of fired actions $a(e)$.

4. MODELS AND METHODS

In this section, we introduce our closed-loop design for the rate-adaptive pacemaker, and describe the heart and pacemaker models, the algorithms for the processing of physiological data, and how the extracted features are combined to compute an adequate adaptive rate. The design, illustrated in Figure 3, realises a dual-sensor VVIR pacemaker, that is, a pacemaker that senses and paces only the ventricle and supports rate adaptation. The heart model (explained in Section 4.2) and the pacemaker model (Section 4.3) are connected into a closed-loop system and implemented as a PTIOA network, where these two components communicate with each other to detect intrinsic heart beats (*sensing*) and deliver artificial beats (*pacing*). We include a so-called *sensor blending* algorithm (Section 4.3.1) that computes the adaptive pacing rate by combining QT-based and accelerometer sensors. The QT sensor exploits the fact that physical and mental stress shortens the QTI. The blending algorithm estimates patient-specific mathematical laws between the QTI and the heart period (also called RR interval) in order to predict an adequate pacing rate based on the QTI measured by the pacemaker electrogram.

To close the loop between the heart dynamics and the sensor data used by the blending algorithm, we generate online, synthetic QTIs (Section 4.2.3) that reflect the state of the heart model simulation. We further devise a method for generating synthetic accelerometer signals, whose parameters can be either estimated from offline recordings in order to reproduce patient-specific characteristics, or configured to reproduce specific physical activities, e.g. running or walking.

Through the parameter estimation method of Section 4.2.1, our approach crucially enables the derivation of personalised heart models from patient ECG data, resulting in *patient-specific virtual patient models*. Importantly, if we use ECG data from a set of multiple patients instead of a single patient, we can apply the same method to estimate *virtual populations of patients*, i.e. models able to fully capture the varied characteristics of the input population. Therefore, the synthetic QTIs received by the blending algorithm in turn reflect the features of the input patient/population, since they are generated from the corresponding virtual patient/population.

We remark that parameter estimation is performed offline, prior to the closed-loop analysis, and that this results in a probabilistic heart model, which motivates the application of quantitative verification techniques.

Open-loop variant. To assess the effectiveness of our closed-loop design, we also consider an open-loop variant, where the QTIs are not generated from the model but are extracted from the input (offline) ECG data. Such obtained QTIs are fixed, implying that the adaptive rate does not reflect the dynamic state of the heart model. For this reason, compared to the open-loop variant, the closed-loop design can reproduce with superior accuracy phenomena of sensor-induced endless-loop tachycardia, as we will see in the experimental evaluation section.

We first introduce the virtual patients over which we evaluate our design, and the corresponding input data for parameter estimation.

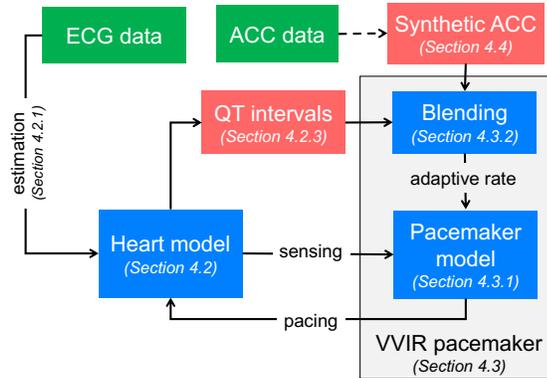


Fig. 3: Closed-loop architecture for rate-adaptation.

4.1. Virtual Patients

We consider two kinds of heart models: patient-specific models for the analysis of personalised treatments (*Subjects 1–3* below); and population-level models, which can capture the varied features of a pool of patients, and thus are suitable for safety verification (*Virtual population* below).

Subject 1. The data consists of 1-lead ECG and 1-axis accelerometer data obtained from a volunteer (one of the authors) and recorded through a BITalino board⁷. We performed recordings at rest and at varying levels of activity, for a total of 15 minutes.

Subject 2. The data corresponds to record 16272 of the MIT-BIH Normal Sinus Rhythm Database [Goldberger et al. 2000], consisting of 1-hour, 2-lead ECG recordings of patients without any particular arrhythmia. Since we do not have accelerometer data for this subject, we generate it following the synthetic generation method discussed in Section 4.4, and using data from [Anguita et al. 2013].

Subject 3. Signals are extracted from the data published by Alan Dix⁸, collected during his 1000+ miles circumlocution of Wales. Public data consists of 35 sequences of 40 hours ECG (1-lead) and 3-axis accelerometer signals, recorded through wearable devices during resting, activity and sleep. Of this database, we consider an extract of day one.

Virtual population. We estimate a virtual population of patients based on 10 subjects taken from the MIT-BIH ST Change Database [Albrecht 1983]⁹, which comprises ECG recordings of different patients during exercise stress tests. Similarly to Subject 2, we generate accelerometer signals based on the data from [Anguita et al. 2013].

In Table I, we also report age and HR at rest of the above four patients. As we will see in Section 4.3.3, these parameters are used within the rate-adaptation algorithm.

We remark that ECG recordings are only used to estimate the parameters of the heart model, while the closed-loop simulation just relies on synthetic data (QTIs and accelerometer).

4.2. Heart model

Below, we illustrate the heart model, including the method for estimation from ECG data, the modelling of relevant arrhythmias, and the online generation of synthetic QTIs. The model reproduces the electrical conduction system of the heart, that is, the propagation of the cardiac action potential (AP) from the atria (the upper chambers) to the ventricles (the lower chambers) through the atrio-ventricular (AV) node. Here, we only provide a high-level view of the model, which is depicted in Figure 4. For a comprehensive description of the model components, we refer to the original publication [Barbot et al. 2015a] and Appendix F.

The model comprises 10 conduction nodes and two main conduction pathways: the *antegrade* conduction, that is, the normal situation where an electrical impulse generated by the SA node stimulates the atria and is conducted to the ventricles through the

	Age	HR _{rest}
Subject 1	21	80
Subject 2	34 ^(*)	75
Subject 3	52	67
V. population	34 ^(*)	77

Table I: Age (y) and HR at rest (HR_{rest}, BPM) for the the virtual patients of our study. ^(*)Due to the anonymity of the record, the age is set to the median of the ages of the participants in [Anguita et al. 2013].

⁷<http://www.bitalino.com/>

⁸<http://alanwalks.wales/data/>

⁹IDs: 300, 301, 302, 304, 307, 308, 310, 311, 312, 313.

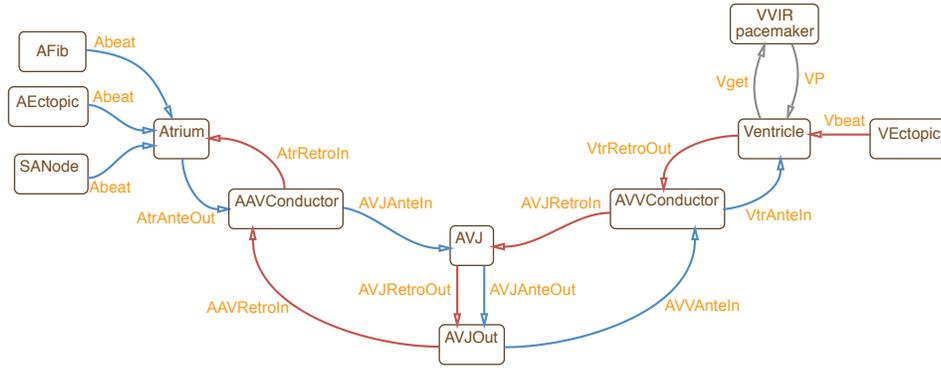


Fig. 4: High-level view of the PTIOA components of the heart model. Arrows indicate synchronisations, with the target component synchronising with the output action performed by the source component, and are labelled with the action name. The anterograde conduction (blue arrows) passes from the Atrium to the Ventricle component. The retrograde conduction (red arrows) follows the opposite path. Grey arrows indicate the connection with the VVIR pacemaker.

AV node (component AVJ in Fig. 4); 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 originates 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 the figure, we also illustrate the connection with the pacemaker component: the device sends impulses to the right ventricle through action VP and senses intrinsic impulses from the right ventricle by synchronising on action Vget.

We have already inspected the Atrium and SANode PTIOAs in Example 2.6.

Component AEctopic generates the so-called atrial ectopic beats, which originate from the spontaneous excitation of different portions of the atrial tissue, mainly from the pulmonary veins, but are not as frequent as the stimuli from the SA node. A dedicated component, AFib, is included to model atrial fibrillation, as we will see in Section 4.2.2.

The Ventricle and VEctopic components describe the AP dynamics in the ventricles, and are modelled in a similar way to the Atrium and AEctopic components, respectively. There are two structurally equivalent conductor components, AAVConductor (Atrium-AV node conductor) and AVVConductor (AV node-Ventricle conductor), whose role is to apply a propagation delay to the transmission of the action potential in both directions. From the physiological viewpoint, the AAVConductor represents 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. The AV node is modelled through components AVJ and AVJOut. Unlike the other components that use delays to implement the conduction of cardiac waves, AVJ provides a detailed representation of the action potential dynamics, whose conformation is reproduced through non-linear update functions. This enables the modelling of complex conditions such as the AV block, as explained below. Component AVJOut applies additional delays depending on the state of the AV node action potential.

Delays in the heart network are mainly probabilistic, as a result of the estimation from ECG data described below and in detail in [Barbot et al. 2015a].

4.2.1. *Personalisation.* The ECG is composed of five main waves, P, Q, R, S and T, which correspond to specific heart events, as illustrated in Figure 5 and explained in [Malmivuo and Plonsey 1995, Chapter 15]. In particular, time-domain ECG features can be mapped into specific actions and parameters in our heart model:

- The P wave corresponds to the activation and propagation of the AP in the atria. In the heart model, the event !Aget captures the moment in which an atrial pacemaker lead would sense the signal, which happens as the ECG signal rises toward the P peak. We assume that !Aget coincides with the P peak.
- The Q wave corresponds to the AP propagating into the Purkinje fibers and the inner walls of the ventricles. In our model, this is equivalent to the event !AVVAnteln, which indicates the beginning of the AP propagation into the AVV conductor component.
- The QRS complex corresponds to the propagation of the de-polarisation wave into the ventricles. Similarly to the P wave in the atria and action !Aget, we assume that action !Vget (denoting AP sensing by the ventricular lead) corresponds exactly to the R peak.
- Finally, the T wave is generated by the ventricle re-polarisation wave. The T peak corresponds to the end of the ventricular absolute refractory period, which is indicated by the delay parameter, Vtr_refrD, in our model.

Note that the PTIOA heart model does not distinguish between left and right ventricle, meaning that it does not have actions or timing delays mapping into the S wave of the ECG, which roughly corresponds to the depolarisation of the left ventricle.

Importantly, through this mapping between the heart model and the ECG, the synthetic QTI (used as a sensor for rate-adaptation) is derived as the time elapsed between the event !AVVAnteln and the ending of the corresponding ventricular refractory period. Details of the generation of synthetic QTIs are given in Section 4.2.3.

Our parameter estimation method relies on the generation of synthetic ECG signals (following the method of [McSharry et al. 2003; Barbot et al. 2015b; Eberz et al. 2017]) that resemble the behaviour of the electrical propagation in the heart during the execution of the model. Such synthetic ECGs are synchronised with the heart model simulation according to the above-explained mapping, and are obtained by summing, for each wave kind, a Gaussian curve that reproduce the morphology of the wave.

In particular, estimation of patient-specific models from ECG data consists of the following two steps:

1) *Extraction of ECG features.* This step results in a set of discrete probability distributions $\{D_f\}_f$, one for each ECG feature f . The features considered are those of Figure 5. Following the mapping described above, some of the model parameters can be directly estimated at this stage, in terms of probabilistic delays distributed according to the relevant feature

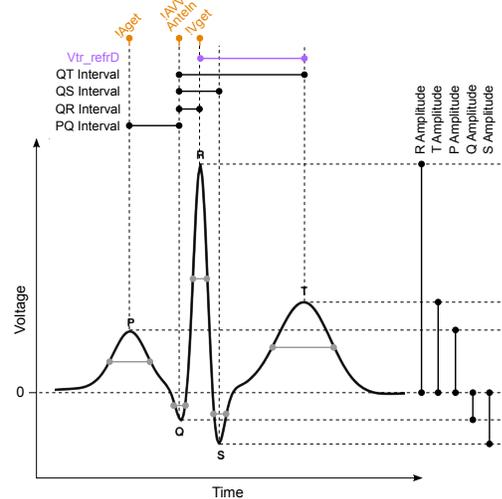


Fig. 5: Example of ECG for one cardiac cycle and corresponding features. Top: duration features, and mappings with actions (orange) and delay parameters (purple) of the heart model. Right: amplitude features. For each wave, the grey solid line indicates the width at half amplitude. Vtr_refrD is the ventricular refractory period.

distribution D_f . For feature extraction, we combine the Pan-Tomkin peak detection algorithm [Pan and Tompkins 1985] to identify R peaks with local search to identify the other peaks.

2) *Estimation of “hidden” parameters.* We estimate the heart parameters that cannot be directly derived from any ECG feature, hence called hidden. This boils down to finding the hidden parameter values that minimise a suitable statistical distance between the input ECG and the synthetic ECG generated from the model simulation. In addition to the explicit parameters of step 1, we estimate two hidden parameters, the AV node threshold voltage AV_Vt and the minimum conduction time in the AV node AV_DMin , because these were the conduction parameters that produced the highest sensitivity for the synthetic ECG. We consider the statistical distance of Eberz et al. [2017], which is computed as the mean of the statistical distances between the feature distributions, and solve the optimisation problem using the pattern search algorithm [Kolda et al. 2003], even though our method supports arbitrary (black-box) optimisation algorithms and distances.

Estimation for virtual population models. In order to estimate the parameters of virtual population models, we proceed by performing steps (1) and (2) for each patient of the input data set. This results in a set of random variables $x_{i,p} \sim D_{i,p}$, describing the values of each estimated parameter i for each patient p . Then, the virtual population model is obtained by combining, for each parameter, the patient-specific distributions across all patients into a single distribution: parameter i is characterized by a random variables $x_i \sim D_i$, where $D_i = \uplus_p D_{i,p}$ is the combined distribution¹⁰. Virtual population models can thus represent all the dynamics that could result from each of their constituent patients, as well as new dynamics that may arise from the “cross-talking” among the individually estimated patient-specific features.

Figure 6 compares the input ECG signals used for the estimation and the corresponding synthetic ECG produced by a simulation trace of the model, after the parameter estimation is performed. We observe good agreement between the two, especially as far as time-domain features (i.e. time between peaks) are concerned, which is most relevant for our PTIOA model. The main difference is in the shape of the T wave, which is naturally asymmetric, while our synthetic ECG waves are generated as symmetric Gaussian curves. We remark, however, that this lack of asymmetry does not affect the estimated time delays, and that it can be easily accounted for by introducing an additional width feature, as done in [Eberz et al. 2017].

4.2.2. *Heart conditions.* The VVIR pacing mode is generally recommended for patients suffering from both AV block and SA node dysfunctions, or chronic atrial impairment (e.g. atrial fibrillation or flutter) [Iaizzo 2009, Fig. 30.7-8]. Therefore, in our model we reproduce AV block and atrial fibrillation (AF) as illustrated below.

AV block. we consider the heart disorder called *Wenckebach AV block* [Hampton and Adlam 2013], a conduction failure causing the loss of ventricular beats due to the progressive prolongation of the AV conduction time. The Wenckebach AV block is a form of chronotropic incompetence typically addressed by setting the pacemaker to the rate-adaptive mode. As illustrated in [Barbot et al. 2015a], this condition can be reproduced in our heart model by increasing parameter AV_Vt , i.e. the depolarisation threshold potential of the AV node. We stress that, although we arbitrarily introduce AV block, this does not compromise the original electrophysiological characteristics of the patient. Indeed, the Wenckebach AV block affects only the duration of the PQ

¹⁰For simplicity, the combined distribution is defined as the multiset sum of the underlying samples.

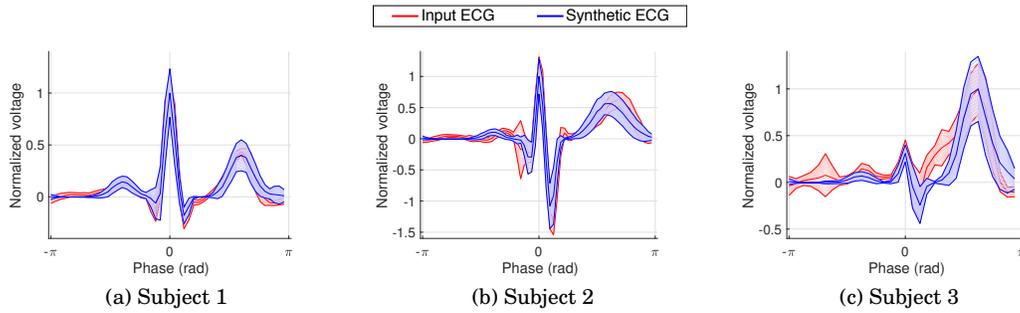


Fig. 6: Comparison of mean input ECG and mean synthetic ECG for the estimated patient-specific models. Shaded areas span ± 1 standard deviation around the mean. Mean ECGs are computed after a linear phase assignment [Sameni et al. 2007], assigning a periodic phase value to each sample in the ECG, starting from one R-peak (phase 0) and ending with the next R-peak (phase 2π). For each heart cycle, amplitudes are normalized by the amplitude of the corresponding highest peak.

interval [Hampton and Adlam 2013, Chapter 4], and thus has no effect on the other ECG features, including the QTI. In the heart model, the AV block prolongs the timing delays solely of the AVJ component, meaning that all the other components maintain the subject-specific behaviour learnt from ECG data.

Atrial Fibrillation. AF is the most common cardiac arrhythmia [Markides and Schilling 2003] and is characterised by a highly disorganised atrial electrical activity, which is thought to be caused by the interaction of rapidly firing ectopic extra-atrial foci and abnormal atrial tissue. We model AF through the AFib component (depicted in Figure 7), which is responsible for the generation of random AF impulses. Following [Lian et al. 2006], the irregular rates of the electrical signals are governed by an exponentially distribution (with mean 0.3 s), and the corresponding action potential (AF.dV) is not strong enough to stimulate, by itself, the AV node.

For the purpose of model personalisation from ECG data, we assume that AF is either recurrent (i.e. few AF episodes occurred to the patient), paroxysmal (i.e. repeated self-terminating AF episodes that last less than 7 days) or persistent (i.e. long lasting AF episodes that can be terminated by external intervention) [Markides and Schilling 2003]. Indeed, in all these cases we can obtain ECG segments showing no AF episodes, and use them to estimate the model parameters and the subject-specific relationship between QT and RR (see Sections 4.3.2 and 4.2.3). For permanent AF (i.e. long lasting AF episodes that cannot be terminated by external intervention), we cannot obtain useful ECG segments to learn the relationship between QT and RR intervals, which is at the basis of rate adaptation. In this case, however, we can still use a different rate adaptation model that works for virtual populations and does not require patient-specific data, as explained in Section 4.3.2.

4.2.3. Synthetic QT intervals. As previously explained, there is a clear patient-specific correlation between the QT and the RR intervals in every heart beat, which is exploited by the blending algorithm to compute the adaptive rate (see Section 4.3.1). Therefore, it is crucial to account for the QT-RR relationship also in our heart model, so that the synthetic QTIs used for rate adaptation reflect this correlation in an accurate way and adapt to the dynamical HR changes due to e.g. exercise, pacing or conduction

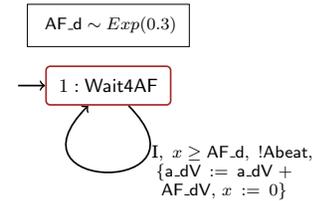


Fig. 7: AFib component.

defects. To this purpose, we derive from the input ECG data a joint discrete probability distribution, denoted by $D_{QT,RR} : \mathbb{R}^2 \rightarrow [0, 1]$, which characterises the observed QT and RR intervals at each heart cycle. For virtual population models, the distribution is obtained by combining the set of samples across the multiple input data sets.

The actual dynamics of the QTI shortening depend on both the effective HR and the autonomic tone [Ahnve and Vallin 1982; Davey and Bateman 1999]. The autonomic tone resembles the ideal rate demand, which is represented in our model by the firing period of the SA node SA.d (see the SA node model of Figure 1a). However, the electrical stimuli produced by the SA node are overwhelmed by the effect of AF, and are also affected by the AV conduction defects, thus producing a discrepancy between the ideal rate demand and the effective HR in cardiac patients. On the basis of the experimental results obtained by Magnano et al. [2002], we assume that the QT interval correlates with both the ideal rate demand and effective HR.

Consider a generic heart cycle t , and let \overline{HR}_t be the corresponding ideal rate demand (in BPM), which depends on the current activity level. The synthetic QTI at t , QT_t , is generated as follows:

- (1) We update the SA node firing period based on the ideal rate: $SA.d_t = 60/\overline{HR}_t$.
- (2) To consider the combined effect of the effective and ideal HR, we define \overline{RR}_t as the weighted sum $\overline{RR}_t = k \cdot SA.d_t + (1 - k) \cdot RR_{t-1}$, where $k \in (0, 1)$ and RR_{t-1} is the effective heart period at $t - 1$, i.e., computed as the time between the last two ventricular events.
- (3) QT_t is finally sampled from distribution $D_{QT,RR}$, conditioned on $RR = \overline{RR}_t$.

Recall that, in the open-loop variant of the VVIR design, QTIs are extracted from offline ECG data and not generated from the heart model.

4.3. VVIR pacemaker model

In this section we introduce the pacemaker model, the blending algorithm, and the estimation of QT-RR laws for rate adaptation.

VVIR pacemaker. The PTIOA network of the VVIR pacemaker is shown in Figure 8. The model is a simplified version of the dual-chamber pacemaker model [Jiang et al. 2012], from which we only retain the components responsible for sensing and pacing the ventricle. We remark that, in the pacemaker network, delays are deterministic. The *Lower Rate Interval (LRI)* component ensures a given lower HR, which in our case corresponds to the adaptive rate. The adaptive pacing period is denoted with t_{AR} and is updated at regular intervals by the sensor blending algorithm (see Section 4.3.1), typically every 2.5 seconds. If there is no sensed ventricular event (represented by the synchronisation on action VS) before t_{AR} expires, the component paces the ventricle, which is implemented by output action !VP.

On the other hand, the *Ventricular Refractory Period (VRP)* component is responsible for the sensing of intrinsic ventricular events. These are generated by the heart network through action Vget. After a Vget is detected, the VRP component notifies the LRI component of the sensed impulse by sending a VS action. After a sensed or

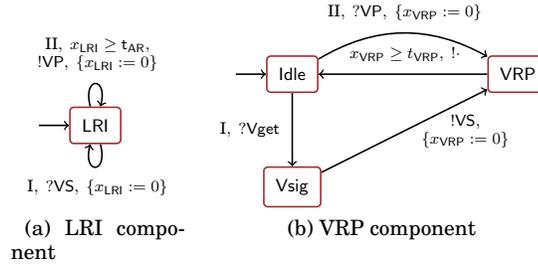


Fig. 8: Components of the VVIR pacemaker model

paced ventricular event, corresponding respectively to firing the edge $V_{sig} \rightarrow V_{RP}$ and $Idle \rightarrow V_{RP}$, the pacemaker enters a refractory period (location V_{RP}), during which sensing is disabled, i.e. no synchronisation on V_{get} is possible. This mimics the natural refractory period of the ventricles, and thus allows for filtering out sensing noise. After the refractory period t_{VRP} has elapsed, the component returns to the $Idle$ location where sensing is re-activated.

4.3.1. Blending algorithm. The blending algorithm combines data from the QT and accelerometer sensors in order to change the adaptive pacing rate t_{AR} such that the pacemaker mimics the behaviour of a healthy conduction system in terms of mental and physical stress. The algorithm accounts for the fact that the accelerometer quickly reacts at the onset of exercise, but lacks precision in the longer run. On the other hand, the QT sensor gives very accurate and specific HR predictions, but has a slower response to exercise [Lau et al. 2007].

We propose a blending algorithm that generalises the rate-adaptation approaches presented in [Lau et al. 2007]. Let t_{AR}^{QT} and t_{AR}^{ACC} be the adaptive pacing intervals computed by the QT and the accelerometer sensors, respectively, as detailed in Sections 4.3.2 and 4.3.3. Let RR_{rest} denote the heart period at rest. For the computation of the adaptive pacing interval t_{AR} , our algorithm distinguishes four cases:

$$t_{AR} = \begin{cases} RR_{rest} & \text{if } t_{AR}^{QT} \geq RR_{rest} \wedge t_{AR}^{ACC} \geq RR_{rest} & \text{(i)} \\ t_{AR}^{QT} & \text{if } t_{AR}^{QT} < RR_{rest} \wedge t_{AR}^{ACC} < RR_{rest} & \text{(ii)} \\ \max(t_{AR}^{QT}, ADL) & \text{if } t_{AR}^{QT} < RR_{rest} \wedge t_{AR}^{ACC} \geq RR_{rest} & \text{(iii)} \\ w(\Delta t) t_{AR}^{ACC} + (1 - w(\Delta t)) RR_{rest} & \text{if } t_{AR}^{QT} \geq RR_{rest} \wedge t_{AR}^{ACC} < RR_{rest} & \text{(iv)} \end{cases}$$

Case (i) corresponds to rest conditions, i.e. when none of the sensors indicate activity. (ii) describes sustained activity, i.e. when both sensors indicate activity, in which case t_{AR} is updated to the pacing period suggested by the QT sensor. (iii) describes either mental or isometric activity, and occurs when only the QT sensor suggests activity. Also in this case, t_{AR} is updated to t_{AR}^{QT} but the pacing period is limited by parameter $ADL = 0.66 s$ (Activities of Daily Living) to avoid excessive pacing [Lau et al. 2007].

In (iv) only the accelerometer detects activity, indicating either an early stage of activity (to which the QTI has not yet adapted), or a false detection. To reflect this duality, the blending algorithm opens a time window during which t_{AR} is set to a weighted combination of t_{AR}^{ACC} and RR_{rest} , where the contribution of the accelerometer gradually decreases according to the parameter $w(\Delta t)$, defined as follows:

$$w(\Delta t) = \begin{cases} -\frac{w_0}{t_{max}} \Delta t + w_0 & \text{if } \Delta t < t_{max} \\ 0 & \text{if } \Delta t \geq t_{max} \end{cases}$$

where w_0 is the initial weight, Δt indicates the time elapsed (in seconds) since case (iv) was entered and t_{max} is the maximum time window. Following the blending algorithms reported in [Lau et al. 2007], we set $w_0 = 0.8$ and $t_{max} = 60$.

Importantly, our algorithm can be easily adapted to reproduce the sensor blending algorithms of commercially available rate-adaptive pacemakers [Lau et al. 2007].

4.3.2. QT sensor. In this section, we discuss methods to effectively compute such QT-RR laws, that is, to estimate a function f such that:

$$QT \approx f(RR). \quad (4)$$

This equation is used within the blending algorithm to update the adaptive pacing rate period t_{AR}^{QT} from QT values. Below, we illustrate the methods used to derive f for patient-specific models, and for virtual population models.

Patient-specific QT-RR laws. Several attempts have been made to formalise the QT-RR relationship, even if experimental and statistical evidence shows that no single law exists that optimally fits every patient [Malik et al. 2002]. We perform an in-depth analysis and comparison of five established mathematical models for describing patient-specific QT-RR relationship, namely:

- (1) Linear model: $QT = a_0 + a_1 RR$
- (2) Quadratic model: $QT = a_0 + a_1 \sqrt{RR}$
- (3) Cubic model: $QT = a_0 + a_1 \sqrt[3]{RR}$
- (4) Hyperbolic model: $QT = a_0 + \frac{a_1}{RR}$
- (5) Exponential model: $QT = a_0 - a_1 \exp(-a_2 RR)$,

We further investigate, for the first time in this context, the application of *Multigene Symbolic Regression* (MSR) [Searson 2015]. MSR consists in the automatic computation of a regression law from a set of basis functions, and thus it notably reduces the need of *trial-and-error* fitting that results from fixed choices of regression functions. We employ the MSR algorithm of Searson [2015] that crucially supports multi-objective optimisation to evaluate the trade-off between the goodness of fit and the expressional complexity of a fitting function. In this way, MSR allows for a Pareto-oriented analysis of different response functions. To ensure a fair comparison, MSR was evaluated by choosing models (1) – (5), $\{+, \cdot, -, /, \sqrt{\cdot}, \sqrt[3]{\cdot}, \exp\}$, as the set of basis functions.

Curve fitting results (plotted in Figure 9) evidence that: (i) QT-RR laws are highly patient-specific; (ii) MSR produces models with superior accuracy, outperforming the above classical mathematical laws; (iii) among models (1-5), the exponential law yields overall the best fitting scores; (iv) unlike the MSR models, the hyperbolic and exponential laws are characterised by good extrapolation accuracy, i.e., they provide satisfactory predictions also for points outside the training dataset.

In general, MSR represents an effective alternative to the current practice of choosing a fixed a-priori mathematical model. Indeed, the “one-fits-all” approach can lead to inaccurate computation of the pacing rate period t_{AR}^{QT} computed by the QT sensor. Nevertheless, relatively to our input data, the exponential law showed a satisfactory overall performance. Further details on MSR and fitting results are in Appendix C.2.

Population-level QT-RR laws. As we just showed, the QT-RR relationship is patient-specific, and thus, for the case of virtual population models, we have to resort to different, more general techniques.

To this purpose, we follow the work of Poore and Mann [1991], where a general mathematical law for computing the adaptive pacing rate is proposed that does not require patient-specific information, but only minimum and maximum values for the RR and the physiological sensor.

Let QT_{max} (QT_{min}) and RR_{max} (RR_{min}) be the maximum (minimum) QT and RR interval observed for the pool of patients considered. This law assumes linear updates from the minimum to the maximum values, i.e. a linear model with parameters:

$$a = \frac{QT_{max} - QT_{min}}{RR_{max} - RR_{min}} \quad \text{and} \quad b = QT_{min} - a \cdot RR_{min}.$$

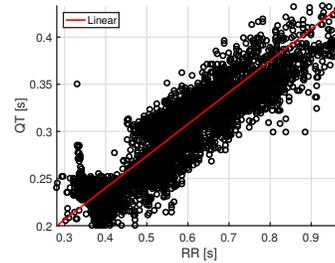


Fig. 10: Estimation of linear QT-RR law for the virtual population.

The estimated linear model for the our virtual population is shown in Fig. 10. Compared to the results of patient-specific estimation, we observe that, for each RR interval, observed QTIs have much higher variability, and that the overall trend is more markedly linear. We remark that this law can be also used for patients affected by permanent atrial impairment, i.e. whose ECG contains no useful QT-RR pairs for the above patient-specific estimation, since it relies on maximal and minimal HR that can be estimated from population-wide models, as per Equation 5 below.

4.3.3. Accelerometer sensor. We process three-axial accelerometer signals in order to detect the physical activity of the patient. Specifically, we derive a real-time approximation of the patient *Metabolic Equivalent of Task (MET)*, an index of physical activity intensity defined as ratio of the work metabolic rate to the resting metabolic rate. Thus, a MET value of 1 indicates resting, and typically, light physical activities have associated a MET value < 3 , while MET values for intense activities are > 6 .

The MET computation is achieved by a decision tree model, estimated by Ohkawara et al. [2011] from the statistical analysis of patient data, and its value is updated at runtime by inspecting a time window of 5 seconds. Let X be the mean value of the accelerometer signal during the time window, and \hat{X} be the mean of the corresponding filtered signal, obtained after removing the base-line drift. The model distinguishes among sedentary activity, light activity and heavy activity, and derives the MET based on X and \hat{X} as follows:

$$\text{MET} = \begin{cases} 1 & \text{if } \hat{X} < k_0 \quad (\text{sedentary activity}) \\ a_0 + a_1 \cdot \hat{X} & \text{if } \frac{X}{\hat{X}} \geq k_1 \quad (\text{light activity}) \\ b_0 + b_1 \cdot \hat{X} & \text{otherwise} \quad (\text{heavy activity}) \end{cases}$$

where $k_0, k_1, a_0, a_1, b_0, b_1$ are the parameters estimated in [Ohkawara et al. 2011].

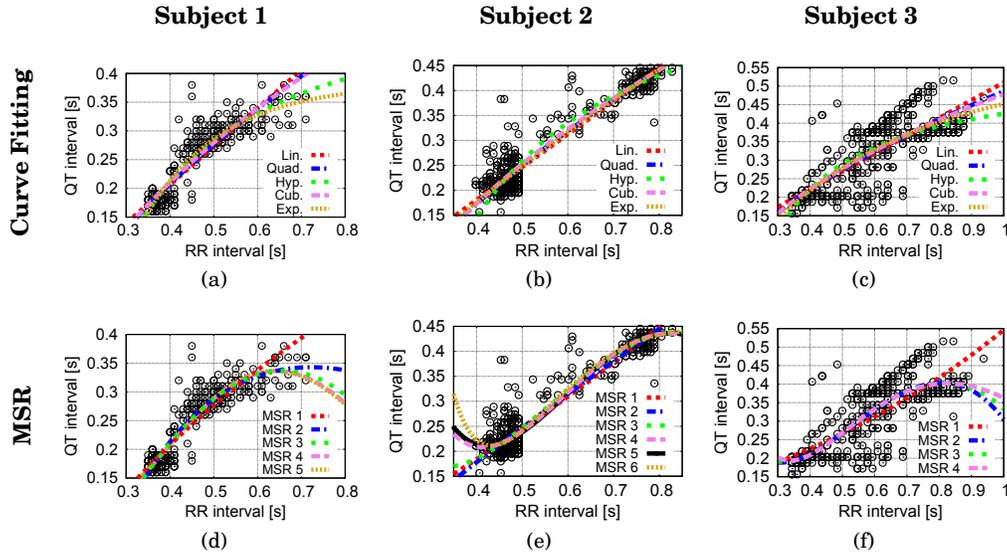


Fig. 9: Estimation of QT-RR laws for the three subjects of our study, and comparison between classical curve fitting (top row) and multigene symbolic regression (bottom row). For the latter, we show the Pareto-optimal models found.

Finally, the adaptive pacing period suggested by the accelerometer, t_{AR}^{ACC} , is derived using Wilkoff's patient-dependent chronotropic response model [Wilkoff et al. 1989], which computes, for a given MET value, the appropriate HR:

$$HR(MET) = \frac{(220 - \text{age} - HR_{\text{rest}})(MET - 1)}{MET_{\text{max}} - 1} + HR_{\text{rest}} \quad \text{and} \quad t_{AR}^{ACC} = \frac{60}{HR(MET)} \quad (5)$$

where age is the age of the patient, HR_{rest} is their HR at rest, and MET_{max} is the maximum MET value. Values of age and HR_{rest} for our virtual patients are given in Table I of Section 4.1. Note that the minimum HR is obtained at minimum MET ($MET = 1$) and is equal to HR_{rest} , while for $MET = MET_{\text{max}}$ we recover the well-known equation for maximum HR: $220 - \text{age}$. Alternative laws for the maximum HR have been proposed [Tanaka et al. 2001], but are outside the scope of our study.

4.4. Generation of synthetic accelerometer signals

We introduce a new method for the generation of synthetic, three-axial accelerometer signals that are either *patient-specific*, i.e. they resemble a given input signal, or *activity-dependent*, i.e. they describe a specific physical activity (e.g. walking or running). Our method is based on a Poisson process model, whose events represent the onset of activity in the generated signal, e.g. climbing stairs.

Let \bar{t} be a time bound and $T_1(\beta), \dots, T_N(\beta)$ be random variables describing the absolute arrival time up to time \bar{t} of N activity events, such that for $i = 1, \dots, N$, $T_i(\beta) - T_{i-1}(\beta)$ is exponentially distributed with mean arrival time β ¹¹. The durations of the activity segments are modelled as independent and identically distributed Gaussian random variables $L_1(\mu, \sigma), \dots, L_N(\mu, \sigma)$, where μ and σ are, respectively, the mean and standard deviation of the distribution. The typical shape of the accelerometer signal during these segments is reproduced by means of a white Gaussian noise generator [Wang et al. 2011]. For each time point $t \in [0, \bar{t}]$ and interval $[t_1, t_2] \subseteq [0, \bar{t}]$, we define the white Gaussian noise generator $WGN_{t_1}^{t_2}(t; \sigma_p)$ as a random variable that is normally distributed with zero mean and standard deviation equals to σ_p if $t \in [t_1, t_2]$, and identically zero otherwise. The synthetic accelerometer signal at time t is hence a random variable defined as:

$$\text{synthAcc}(t; \beta, \mu, \sigma, \sigma_p) = \sum_{i=1}^N WGN_{T_i(\beta)}^{T_i(\beta)+L_i(\mu, \sigma)}(t; \sigma_p) \quad (6)$$

We finally add noise to the signal in order to mimic background noise effects. Parameters β , μ , σ and σ_p are among the most used features for accelerometer signal processing [Chernbumroong et al. 2011] and can be automatically estimated from accelerometer recordings, or tuned to reproduce a prescribed, activity-dependent morphology. For the former, we have implemented a detection algorithm able to identify the regions of interest in the signal, and extract the relevant statistical features from it. An example of synthetic accelerometer signal is shown in Figure 11, using as input accelerometer data taken from [Anguita et al. 2013]. We can observe that the key features of the input signal are preserved by our algorithm and are clearly visible in the generated synthetic signal.

Validation. To validate our algorithm, we need to show that the input accelerometer signal and the corresponding synthetic signals are consistent with respect to the decision-tree model used for the accelerometer, that is, the two signals must be classi-

¹¹We assume $T_0(\beta) = 0$.

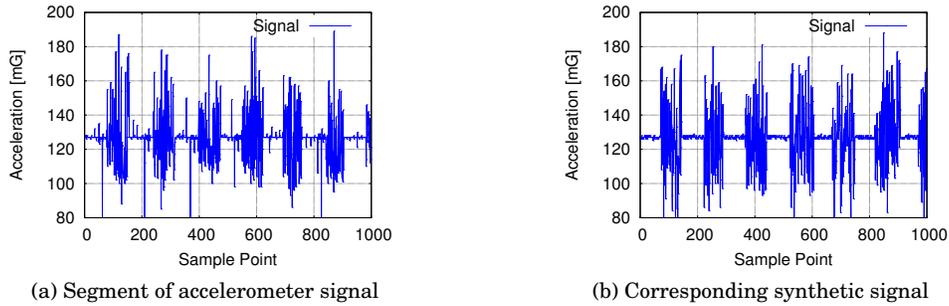


Fig. 11: Example of the results produced by our algorithm for the generation of patient-specific, activity-dependent, synthetic accelerometer signals.

fied to the same activity level (sedentary, light or heavy activity). We are also interested in how the predicted metabolic activity differs between the two.

To this end, we consider one hundred recordings from the training set of [Anguita et al. 2013] and generate, for each, one hundred synthetic signals. Our synthetic signal generator produced a satisfactory 93% of correct classifications, with a mean deviation of only 0.17 MET in the predicted metabolic activity.

5. RESULTS

In Section 5.1, we analyse the behaviour of the VVIR pacemaker along a single simulation, and how it compares with alternative pacemaker designs. In Section 5.2 we illustrate the verification results obtained through statistical model checking.

Unless otherwise stated, the following experiments assume a condition of AV block and AF (modelled as per Section 4.2), which is typically addressed by VVIR pacing [Iaizzo 2009, Fig. 30.7-8].

Implementation. COSMOS statistical model checker works by executing efficient C++ code generated from Stochastic Petri Net (SPN) models. Therefore, we apply existing methods [Barbot et al. 2016] to translate the heart-pacemaker PTIOA network into an SPN. The blending and signal processing algorithms described in Sections 4.3.1–4.4 are implemented as C++ code, called by COSMOS during execution. Note that these algorithms have only the effect of updating variable values of the PTIOA network. An implementation is available at www.veriware.org/heart_pm_methods.php#heartverify.

5.1. Simulation and comparison with fixed-rate pacemaker

In this section, we compare our VVIR pacemaker design with its fixed-rate variant (called VVI) during sustained activity, and further compare the dual-sensor blending algorithm with the corresponding single-sensor variants, i.e. obtained by disabling one of the two sensors. For a deeper look at the behaviour of the VVIR pacemaker, here we consider a single simulation trace, parameterised with an extract of the ECG and accelerometer data from subject 1 during activity (see Section 4.1), and chose an exponential QT-RR law in the blending algorithm.

Given that subject 1 has a healthy ECG, we use the corresponding sequence of RR intervals to construct the ideal rate demand curve. Recall that in our model the ideal rate demand is reflected by the firing rate of the SA node (see Section 4.2.3), which propagates impulses into the atrium. Therefore, to obtain a clear illustration of the atrial activity, only for this experiment, we assume no AF (only AV block).

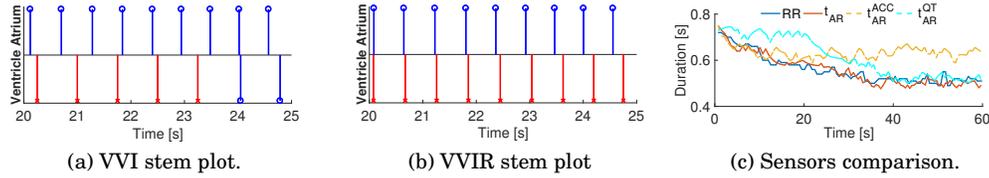


Fig. 12: Results of model simulation. Left and centre figures depict the stem plots of a 5-second simulation (upper half, atrial activity; lower half, ventricular). Blue stems indicate intrinsic atrial impulses, or atrial impulses conducted in the ventricle; red stems indicate paced ventricular beats. **Right:** pacing period of different rate-adaptive sensors: blended (red), accelerometer (yellow) and QT-based (aqua), compared to the ideal demand (blue).

Analysis of pacing modes. The results are summarised in Figures 12a and 12b, which give a detailed view of the atrial and ventricular events during a 5 seconds simulation. We observe that, in the VVI mode, the fixed ventricular rate imposed by the pacemaker is not commensurate to the atrial activity. There are only a few impulses successfully conducted from the atrium to the ventricle, which happens at regular intervals in the Wenckebach AV block. This behaviour does not occur in the VVIR mode, where the adaptive pacing rate ensures that the ventricular rhythm follows the ideal demand through an increased pacemaker activity.

Analysis of blending algorithms. In Figure 12c we compare the pacing rates computed by our blending algorithm under exercise with the single-sensor pacing rates. We observe that, if we consider only the accelerometer, the rate-adaptive pacemaker reacts appropriately to the onset of exercise (visible by the decreased RR duration) and sets an adequate pacing rate for the first ≈ 20 seconds, after which it cannot account for the increased rate demand. On the other hand, the QT sensor alone fails to detect the onset of activity in a timely manner, but after the first ≈ 25 seconds provides more and more accurate predictions. Our blending algorithm takes advantage of both sensors: it successfully detects the onset of patient activity and predicts the right chronotropic response in the longer run, with the resulting adaptive rate closely following the ideal demand at any time.

5.2. Statistical model checking experiments

We employ statistical model checking (SMC) to provide formal and quantitative guarantees on the behaviour of the VVIR pacemaker. Our study involves two broad kinds of analyses, in order to investigate: 1) effects of pacemaker malfunctions on cardiac dynamics; 2) adequacy of pacing treatment under varying physiological conditions.

The experiments of Sections 5.2.2 and 5.2.3 belong to the first kind of analyses. In the former experiment, we investigate phenomena of sensor-induced endless-loop tachycardia, comparing closed- and open-loop designs. In the latter experiment, we analyse how the heart rate regularity is affected by a faulty pacemaker lead that causes oversensing. This defect is typically random and thus, can be meaningfully investigated through SMC.

The experiments of Sections 5.2.4 and 5.2.5 belong to the second kind of analyses. In the former experiment, we simulate exercise curves and stress tests in order to assess which QT-RR law provides the most adequate rate at every activity stage. In the latter experiment, we analyse how the pacemaker responds to different conditions of the cardiac conduction system, namely, at different degrees of AV block.

Where not otherwise specified, the following analyses are obtained by instantiating the QT sensor of the rate-adaptive algorithm with the exponential law. For SMC, we only consider time-bounded HASL properties¹².

5.2.1. Verification properties.

Absence of Induced Tachycardia. This HASL property evaluates the probability that, in resting conditions, the pacemaker does not wrongly switch to activity mode, thus inducing an excessive HR (tachycardia). The corresponding LHA (illustrated in Appendix E) is designed to count, through variable $nOKs$, the number of adequate rate updates, i.e. inducing an effective HR that does not exceed the HR at rest of the subject. Variable $nUpd$ keeps track of the total number of rate updates. The HASL formula is defined as:

$$\phi_{NO_TC} = \mathbb{E} \left[\text{LAST} \left(\frac{nOKs}{nUpd} \right) \right], \quad (7)$$

corresponding to the expectation of the final value of the ratio of adequate updates. Since the adequacy/non-adequacy of each update forms a Bernoulli process, ϕ_{NO_TC} actually estimates the probability that the rate is updated correctly, in a way that it does not induce tachycardia.

PDF of Fraction of Paced Beats. We consider the HASL property $\phi_{PDF(VP)}$ introduced in Example 3.1. The property evaluates the PDF of the fraction of paced beats, or, simply put, how much artificial pacing is needed to ensure a safe heart rhythm.

Heart Rate Regularity. We introduce a measure of HR regularity for a patient under pacing conditions, defined as the average deviation between consecutive RR intervals. Let n be the number of beats along a given execution, and RR_i be the RR associated to the i -th beat. We define the regularity of the path as

$$\text{Reg} = \frac{1}{n-1} \sum_{i=1}^{n-1} |RR_{i+1} - RR_i|.$$

Low Reg values are indicative of paths where the RRs change smoothly, whereas high Reg values indicate abrupt HR changes, potentially introducing excessive stress on the patient's cardiac tissue. Due to the intrinsic HR variability, a value of zero for Reg is impossible to achieve.

In this regard, we can compute a baseline level for Reg, corresponding to the regularity of a perfectly healthy subject in a particular activity stage. This can be obtained by assuming that the RR delays of a healthy subject in a particular stage of activity are normally distributed (as done in [Lian et al. 2010]) and is defined by:

$$\text{Reg}^* = 2\sqrt{\frac{\sigma^2}{\pi}}. \quad (8)$$

where σ is the standard deviation of the estimated Gaussian distribution for RR intervals at the given activity stage. The full derivation of (8) is in Appendix E.

The LHA for computing Reg over a simulation path is depicted in Figure 22 of the Appendix. It contains a variable Reg_Σ for storing the cumulative sum of the deviations between consecutive RRs, and a variable $Nbeats$, tracing the number of ventricular

¹²These are characterised by an LHA with a global clock t and unique final state whose incoming edges are autonomous and have guard $g = t \geq T_{max}$, where T_{max} is the time bound.

beats. The HASL formula is:

$$\phi_{\text{Reg}} = \mathbb{E} \left[\text{LAST} \left(\frac{\text{Reg}_{\Sigma}}{\text{Nbeats}} \right) \right]. \quad (9)$$

5.2.2. Endless-loop tachycardia: open- vs closed-loop. We analyse phenomena of *Pacemaker Mediated Tachycardia (PMT)*, also called endless-loop tachycardia, and compare the accuracy of the open- and closed-loop VVIR architectures in reproducing this malfunction. PMT is typical of (dual-chamber) DDD pacemakers [Monteil et al. 2015], but has been found to occur also in VVIR pacemakers (see e.g. [Bohm et al. 2010]). In this case, the event is triggered by a sensor malfunction causing the pacemaker to wrongly detect activity, thus inducing a self-sustained increase of the ventricular pacing rate, as illustrated in Figure 13.

To intentionally induce PMT in our model, we consider a naïve blending algorithm where cross-checking of multiple sensors is disabled and thus, any of the two sensors can set the pacing rate as it predicts. The PMT episode is initiated by simulating a fault in the accelerometer sensor that detects activity when the subject is actually at rest. This results in a wrongly increased pacing rate and, as a side effect, in a shortening of the QTI [Ahnve and Vallin 1982]. In turn, shorter QTIs are translated by the QT sensor into higher pacing frequencies, leading to endless-loop tachycardia.

We verify property $\phi_{\text{NO_TC}}$ (see Equation 7) to estimate the probability that, after an induced sensor malfunctioning of 30 seconds, the pacemaker does not induce tachycardia in the following 90 seconds. Detailed experimental setup and numerical results are reported in Appendix E. In a nutshell, we obtain that in the closed-loop architecture, PMT is from 18% to 59% more likely to happen than in the open-loop case across the three virtual patients, for which tachycardia occurs with relatively low probability (ranging from 1% to 30%).

Therefore, SMC results confirm that the feedback-loop nature of PMT can be only reproduced in the closed-loop model and cannot be well captured by the open-loop model, since this uses offline data that do not reflect the runtime activity of the pacemaker. We remark, however, that when using the complete blending algorithm of Section 4.3.1, PMT is terminated.

5.2.3. Regularity under over-sensing. We analyse how over-sensing, i.e. the misinterpretation of artifacts or other waves as R peaks by the pacemaker, affects the HR regularity (see Equation 5.2.1). The occurrence of an over-sensing event is modelled as an exponentially distributed variable, with mean waiting time given by the parameter β_{over} , ranging in the interval $[0.5, 10]$ seconds. We evaluate regularity in both resting and activity conditions through SMC of the HASL formula ϕ_{Reg} (see Equation 9).

Results for ϕ_{Reg} are illustrated in Figure 14. Recall that smaller ϕ_{Reg} values imply a more regular heart rhythm. As expected, the worst regularity is registered when severe over-sensing is in action, and this monotonically improves as the rate of mis-detected beats decreases.

We obtain similar results among the three patient-specific models and between resting and activity conditions, suggesting that the pacemaker is able to maintain good levels of regularity independently of the activity stage and the differences intrinsic to the three subjects.

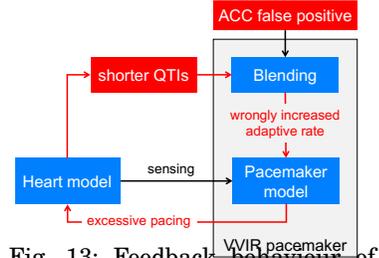


Fig. 13: Feedback behaviour of sensor-induced tachycardia

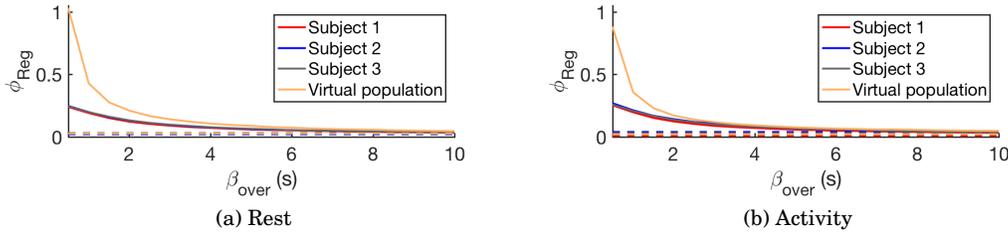


Fig. 14: HR regularity (ϕ_{Reg} , y-axis) at different over-sensing rates (β_{over} , x-axis). Mean and 99% confidence intervals for ϕ_{Reg} are indicated with lines and shaded areas, respectively. Dashed lines represent the corresponding ideal regularity values (Reg^* , see Equation 8). For SMC, we use 1000 simulations for each subject.

For the virtual population model, this analysis yields considerably higher values of ϕ_{Reg} (less regular HR) for high values of over-sensing (i.e. low values of β_{over}). This is explained by the higher width of the parameters distributions estimated by the population-wide approach. Note, however, that values of ϕ_{Reg} comparable to those obtained for the patient-specific approach are restored for high values of β_{over} .

We further observe that the obtained confidence intervals for ϕ_{Reg} are very narrow, meaning that SMC results are precise and, more importantly, that the regularity measure is robust with respect to the intrinsic heart variability and the random occurrence of over-sensing events.

5.2.4. Adequacy of Pacing Rate During Exercise. We evaluate how the HR under AV block and AF is corrected by the rate-adaptive pacemaker during exercise, and in particular, how the effective HR induced by the pacemaker fits the required ideal rate demand (I.D.). Recall that, in our model, the I.D. is set by updating the firing period of the SA node (see Section 4.2.3).

We consider three experiments, respectively describing:

- (1) a vigorous, 20-minute exercise consisting of four activity stages: activity onset, sustained activity, decay, rest;
- (2) a more intense, 10-minute exercise consisting of three stages: activity onset, sustained activity, decay; and
- (3) the *Bruce protocol* [Candinas et al. 1997], one of the most common clinical stress tests. It is composed of seven activity stages performed at increasing intensity, with duration of 3 minutes each. The I.D. at each stage is described in MET, from which we derive the corresponding ideal HR by using Equation 5 for each virtual patient (see Table X in the Appendix for the ideal HR of each patient).

Regarding accelerometer data, we generate synthetic signals using parameters estimated from recordings at rest (for rest and decay stages) and during activity (for onset and sustained activity stages). As explained in Section 4.1 we use accelerometer data from [Anguita et al. 2013] for subject 2 and for the virtual population.

Results are shown in Figure 15 and compare the behaviour of different QT-RR laws in the blending algorithm: linear model, exponential model, and lowest mean-square error MSR model. We generally observe that the effective HR qualitatively and quantitatively agrees with the I.D., demonstrating that the blending algorithm is able to provide an adequate heart behaviour during exercise.

In particular, for the Bruce protocol, the rate-adaptive algorithm achieves an almost ideal behaviour, accounting for the sudden I.D. changes in a timely manner when

transitioning between activity stages, and successfully maintaining the effective HR constant during each stage. The only exception is the linear model for subject 1, which underestimates the I.D. during stages 4 – 6. The cause of this behaviour is to be found in the decreased fitting performance of the linear model for the corresponding range of RR values (see Fig. 9a), even though the discrepancy between I.D. and effective rate is never above 20 BPM.

Noteworthy are the narrow confidence intervals in the patient-specific experiments (Subject 1-3), obtained with just 100 simulations, independently of the subject or the used QT-RR law. This proves the robustness of our pacemaker design, able to mitigate the intrinsic stochasticity of the heart model.

Results obtained for the virtual population demonstrate that, also in this case, the rate-adaptive pacemaker is able to follow the overall trend of the exercise curves, with a very similar performance to the patient-specific experiments. At sustained activity rates, when the pacing rate is mainly determined by the QT sensor, we notice an increased width of the confidence interval, which mirrors the increased variability of the QTIs estimated from multiple patients.

Further note that the quick initial HR increase, as well as the quick decrease at the decay stage (for exercise curves 1 and 2), is basically independent from the QT-RR law used in the QT sensor and of the virtual patient. This is in fact mostly driven by the accelerometer sensor that successfully detects the onset/offset of activity.

5.2.5. AV block degree. We perform a parametric analysis of AV_Vt, i.e. the AV node depolarization threshold that, as explained in Section 4.2, can be tuned to reproduce AV block conditions, such that higher AV_Vt values imply worse heart conduction. In particular, we let AV_Vt deviate from its default (healthy) value of -4.0 mV [Barbot et al. 2015a] by an exponentially distributed random variable with mean β_{vt} . We consider property $\phi_{PDF(VP)}$, defined in Example 3.1, to evaluate the probability density function of the fraction of paced beats, with β_{vt} spanning from 0.1 (normal AV conduction) to 2.5 (severe AV block).

The heat-maps in Figure 16 depict, for each subject and for both resting and sustained activity conditions, how the distribution of the fraction of paced beats is affected by parameter β_{vt} . We observe that as the condition of the conduction system worsens, the number of paced beats increases, demonstrating that the pacemaker can well adapt to different degrees of AV block. This increase slowly saturates around the value of 0.9. This indicates that the co-existence of AF and severe AV block in a subject makes the pacemaker almost essential, with only a small percentage of atrial impulses propagating into the ventricles. At high β_{vt} values, we further notice a substantial decrease of the width of the distributions (redder area, implying a higher probability density). This suggests a scenario where heart cycles are primarily initiated by the pacemaker, effectively reducing the intrinsic variability of the patient (diseased) heart dynamics.

For the virtual population, we report a fraction of paced beats always greater than 0.9, independently of the AV block degree and of the activity stage. This suggests that the QT-RR law implemented for the virtual population is more conservative, causing the device to pace the heart even when only AF and no AV block is present.

6. CONCLUSION

Safety assurance of rate-adaptive pacemakers demands rigorous verification methods able to account for the high specificity of the patient's electrophysiology, and how this changes under different levels of physical or mental stress. In this work, we addressed this challenge through the development of a formal, data-driven and model-based approach to the automated, closed-loop verification of dual-sensor rate-adaptive pacemakers. Our approach uniquely combines methods for the personalisation of heart

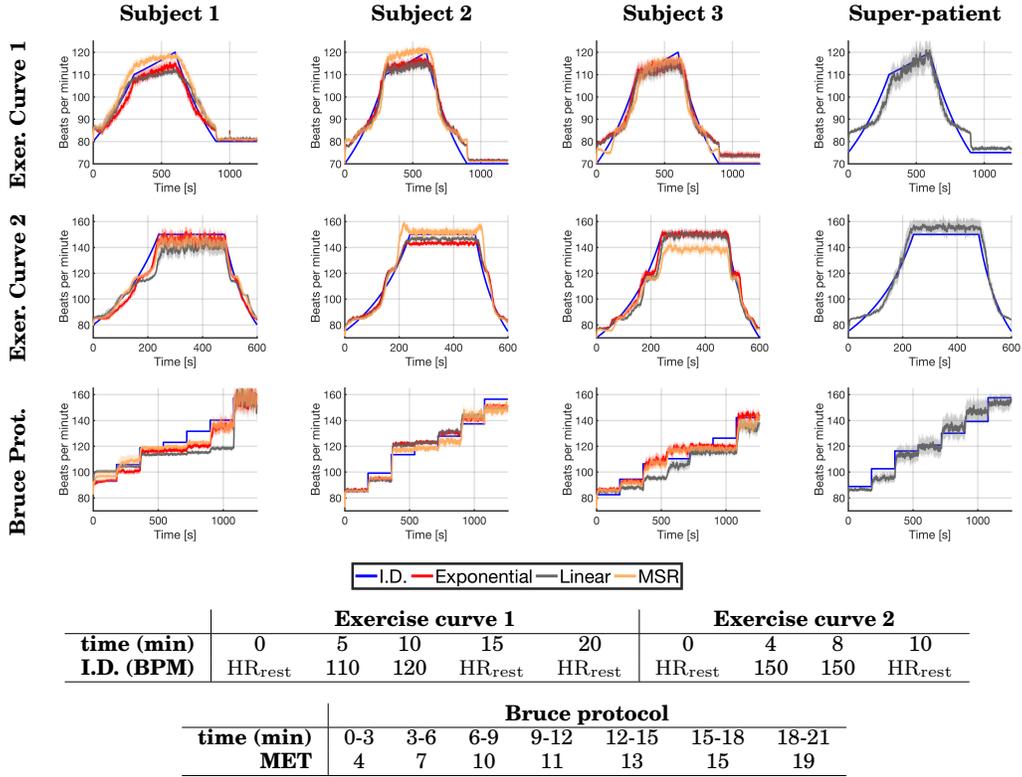


Fig. 15: HR during simulated exercise curves, obtained through 100 simulations for each subject. The ideal rate demand (I.D., blue) is compared to the HR under different QT-RR laws for the blending algorithm: exponential, linear and best fitting MSR model. Shaded areas indicate 99% confidence intervals. Tables describe the different stages of each test.

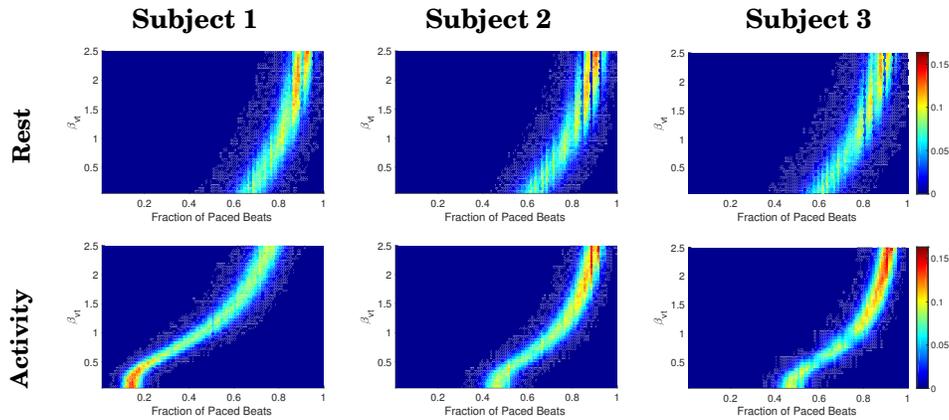


Fig. 16: Probability density function of the fraction of paced beats (x-axis) under different degrees of AV block severity, obtained by increasing the AV node threshold potential by parameter β_{vt} (y-axis). Colors indicate the SMC results for formula $\phi_{PDF(VP)}$ (see Example 3.1), evaluated for each value of β_{vt} using a total of 50,000 simulations.

models from patient data with the online generation of synthetic, model-based sensor data used for rate adaptation, seamlessly supporting patient-specific models for the analysis of personalised treatments and virtual population models targeted to safety verification. We performed extensive statistical model checking experiments to assess the device's behaviour under a variety of conditions, leading to quantitative insights that would be difficult, if not impossible, to obtain with testing on real patients. By considerably reducing exercise testing, our techniques offer great potential to enhance the design and development process of rate-adaptive pacemakers, and to provide clinicians with appropriate and patient-specific treatment indications.

REFERENCES

- Freddy M Abi-Samra, Narendra Singh, Benjamin L Rosin, Jerome V Dwyer, Crystal D Miller, CLEAR Study Investigators, and others. 2013. Effect of rate-adaptive pacing on performance and physiological parameters during activities of daily living in the elderly: results from the CLEAR (Cynos Responds with Physiologic Rate Changes during Daily Activities) study. *Europace* 15, 6 (2013), 849–856.
- S Ahnve and H Vallin. 1982. Influence of heart rate and inhibition of autonomic tone on the QT interval. *Circulation* 65, 3 (1982), 435–439.
- Weiwei Ai, Nitish Patel, and Partha Roop. 2016. Requirements-centric closed-loop validation of implantable cardiac devices. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2016*. IEEE, 846–849.
- W. Ai, N. Patel, P. Roop, A. Malik, S. Andalamp, E. Yip, N. Allen, and M. Trew. 2017. A Parametric Computational Model of the Action Potential of Pacemaker Cells. *IEEE Transactions on Biomedical Engineering* PP, 99 (2017), 1–1.
- Paul Albrecht. 1983. *ST segment characterization for long term automated ECG analysis*.
- Rajeev Alur. 1999. Timed automata. In *International Conference on Computer Aided Verification*. Springer, 8–22.
- Francesco Amigoni, Alessandro Beda, and Nicola Gatti. 2006. Combining rate-adaptive cardiac pacing algorithms via multiagent negotiation. *Information Technology in Biomedicine, IEEE Transactions on* 10, 1 (2006), 11–18.
- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. 2013. A public domain dataset for human activity recognition using smartphones. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN*.
- Paolo Ballarini, Benoît Barbot, Marie Dufлот, Serge Haddad, and Nihal Pekergin. 2015. HASL: A new approach for performance evaluation and model checking from concepts to experimentation. *Performance Evaluation* 90 (2015), 53–77.
- Benoît Barbot, Marta Kwiatkowska, Alexandru Mereacre, and Nicola Paoletti. 2015a. Estimation and verification of hybrid heart models for personalised medical and wearable devices. In *Computational Methods in Systems Biology (LNCS)*, Vol. 9308. Springer, 3–7.
- Benoît Barbot, Marta Kwiatkowska, Alexandru Mereacre, and Nicola Paoletti. 2015b. *Estimation and Verification of Hybrid Heart Models for Personalised Medical and Wearable Devices*. Technical Report CS-RR-15-05. Department of Computer Science, University of Oxford.
- Benoît Barbot, Marta Kwiatkowska, Alexandru Mereacre, and Nicola Paoletti. 2016. Building Power Consumption Models from Executable Timed I/O Automata Specifications. In *Hybrid Systems: Computation and Control (HSCC '16)*. ACM, 195–204.
- Chris Barker, Marta Kwiatkowska, Alexandru Mereacre, Nicola Paoletti, and Andrea Patanè. 2015. Hardware-in-the-loop simulation and energy optimization of cardiac pacemakers. In *IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 7188–7191.
- Ezio Bartocci, Flavio Corradini, Maria Rita Di Berardini, Emilia Entcheva, Scott A Smolka, and Radu Grosu. 2009. Modeling and simulation of cardiac tissue using hybrid i/o automata. *Theoretical Computer Science* 410, 33 (2009), 3149–3165.
- Alan D Bernstein and others. 2002. The Revised NASPE/BPEG Generic Code for Antibradycardia, Adaptive-Rate, and Multisite Pacing. *Pacing and clinical electrophysiology* 25, 2 (2002), 260–264.
- Adam Bohm, Robert Kiss, Paul Dorian, and Arnold Pinter. 2010. Single-chamber, rate-responsive pacemaker-mediated tachycardia. *Canadian Journal of Cardiology* 26, 9 (2010), e340.
- Reto Candinas, Markus Jakob, Thomas A Buckingham, Heidi Mattmann, and F Wolfgang Amann. 1997. Vibration, acceleration, gravitation, and movement: activity controlled rate adaptive pacing during tread-

- mill exercise testing and daily life activities. *Pacing and clinical electrophysiology* 20, 7 (1997), 1777–1786.
- Taolue Chen, Marco Diciolla, Marta Kwiatkowska, and Alexandru Mereacre. 2014. Quantitative verification of implantable cardiac pacemakers over hybrid heart models. *Information and Computation* 236 (2014), 87–101.
- Saisakul Chernbumroong, Anthony S Atkins, and Hongnian Yu. 2011. Activity classification using a single wrist-worn accelerometer. In *Software, Knowledge Information, Industrial Management and Applications (SKIMA), 2011 5th International Conference on*. IEEE, 1–6.
- Patrick Davey and Jeffrey Bateman. 1999. Heart rate and catecholamine contribution to QT interval shortening on exercise. *Clinical cardiology* 22, 8 (1999), 513–518.
- Simonetta Dell’Orto, Paolo Valli, and Enrico Maria Greco. 2004. Sensors for rate responsive pacing. *Indian pacing and electrophysiology journal* 4, 3 (2004), 137.
- Simon Eberz, Nicola Paoletti, Marc Roeschlin, Andrea Patanè, Marta Kwiatkowska, and Ivan Martinovic. 2017. Broken Hearted: How To Attack ECG Biometrics. *Network and Distributed System Security Symposium* (2017).
- Kenneth A Ellenbogen, Bruce L Wilkoff, G Neal Kay, Chu Pak Lau, and Angelo Auricchio. 2016. *Clinical cardiac pacing, defibrillation and resynchronization therapy*. Elsevier Health Sciences.
- Ary L Goldberger and others. 2000. Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals. *Circulation* 101, 23 (2000), e215–e220.
- Artur Oliveira Gomes and Marcel Vinicius Medeiros Oliveira. 2009. Formal specification of a cardiac pacing system. In *FM 2009: Formal Methods*. Springer, 692–707.
- Radu Grosu, Gregory Batt, Flavio H Fenton, James Glimm, Colas Le Guernic, Scott A Smolka, and Ezio Bartocci. 2011. From cardiac cells to genetic regulatory networks. In *Computer Aided Verification*. Springer, 396–411.
- John R Hampton and David Adlam. 2013. *The ECG in practice*. Elsevier Health Sciences.
- Paul A Iaizzo. 2009. *Handbook of cardiac anatomy, physiology, and devices*. Springer Science & Business Media.
- Zhihao Jiang, Miroslav Pajic, Salar Moarref, Rajeev Alur, and Rahul Mangharam. 2012. Modeling and verification of a dual chamber implantable pacemaker. In *Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 188–203.
- Tamara G Kolda, Robert Michael Lewis, and Virginia Torczon. 2003. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review* 45, 3 (2003), 385–482.
- Marta Kwiatkowska, Harriet Lea-Banks, Alexandru Mereacre, and Nicola Paoletti. 2014. Formal modelling and validation of rate-adaptive pacemakers. In *Healthcare Informatics (ICHI), 2014 IEEE International Conference on*. IEEE, 23–32.
- Marta Kwiatkowska, Alexandru Mereacre, Nicola Paoletti, and Andrea Patanè. 2015. Synthesising robust and optimal parameters for cardiac pacemakers using symbolic and evolutionary computation techniques. In *Hybrid Systems and Biology (HSB 2015) (LNCS/LNBI)*, Vol. 9271. Springer, 119–140.
- Gervasio A Lamas, J David Knight, Michael O Sweeney, Marcus Mianulli, Vinod Jorapur, Koroush Khalighi, James R Cook, Russell Silverman, Lawrence Rosenthal, Nancy Clapp-Channing, and others. 2007. Impact of rate-modulated pacing on quality of life and exercise capacity: evidence from the Advanced Elements of Pacing Randomized Controlled Trial (ADEPT). *Heart Rhythm* 4, 9 (2007), 1125–1132.
- Chu-Pak Lau, Hung-Fat Tse, A. John Camm, and Serge S. Barold. 2007. Evolution of pacing for bradycardias: sensors. *European Heart Journal Supplements* 9, suppl I (2007), I11–I22.
- Jie Lian, Hannes Krätschmer, Dirk Müssig, and L Stotts. 2010. Open source modeling of heart rhythm and cardiac pacing. *Open Pacing Electrophysiol Ther J* 3 (2010), 4.
- Jie Lian, D Müssig, and Volker Lang. 2006. Computer modeling of ventricular rhythm during atrial fibrillation and ventricular pacing. *IEEE transactions on biomedical engineering* 53, 8 (2006), 1512–1520.
- Hugo Daniel Macedo, Peter Gorm Larsen, and John Fitzgerald. 2008. Incremental Development of a Distributed Real-Time Model of a Cardiac Pacing System Using VDM. In *Formal Methods*. 181.
- Anthony R Magnano, Steve Holleran, Rajasekhar Ramakrishnan, James A Reiffel, and Daniel M Bloomfield. 2002. Autonomic nervous system influences on QT interval in normal subjects. *Journal of the American College of Cardiology* 39, 11 (2002), 1820–1826.
- Merak Malik, P Färhom, V Batchvarov, K Hnatkova, and AJ Camm. 2002. Relation between QT and RR intervals is highly individual among healthy subjects: implications for heart rate correction of the QT interval. *Heart* 87, 3 (2002), 220–228.
- Jaakko Malmivuo and Robert Plonsey. 1995. *Bioelectromagnetism: principles and applications of bioelectric and biomagnetic fields*. Oxford University Press, USA.

- Vias Markides and Richard J Schilling. 2003. Atrial fibrillation: classification, pathophysiology, mechanisms and drug treatment. *Heart* 89, 8 (2003), 939–943.
- Patrick E McSharry, Gari D Clifford, Lionel Tarassenko, Leonard Smith, and others. 2003. A dynamical model for generating synthetic electrocardiogram signals. *Biomedical Engineering, IEEE Transactions on* 50, 3 (2003), 289–294.
- Dominique Méry, Bernhard Schätz, and Alan Wasssyng. 2014. The Pacemaker Challenge: Developing Certifiable Medical Devices (Dagstuhl Seminar 14062). *Dagstuhl Reports* 4, 2 (2014).
- Dominique Méry and Neeraj Kumar Singh. 2009. *Pacemaker's Functional Behaviors in Event-B*. Rapport de recherche. <http://hal.inria.fr/inria-00419973>
- Benjamin Monteil, Sylvain Ploux, Romain Eschalier, Philippe Ritter, Michel Haissaguerre, Jayanthi N Koneru, Kenneth A Ellenbogen, and Pierre Bordachar. 2015. Pacemaker-Mediated Tachycardia: Manufacturer Specifics and Spectrum of Cases. *Pacing and Clinical Electrophysiology* 38, 12 (2015), 1489–1498.
- Kazunori Ohkawara, Yoshitake Oshima, Yuki Hikiyara, Kazuko Ishikawa-Takata, Izumi Tabata, and Shigeo Tanaka. 2011. Real-time estimation of daily physical activity intensity by a triaxial accelerometer and a gravity-removal classification algorithm. *British Journal of Nutrition* 105 (6 2011), 1681–1691.
- Miroslav Pajic, Zhihao Jiang, Insup Lee, Oleg Sokolsky, and Rahul Mangharam. 2012. From verification to implementation: A model translation tool and a pacemaker case study. In *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2012 IEEE 18th*. IEEE, 173–184.
- Jiapu Pan and Willis J Tompkins. 1985. A real-time QRS detection algorithm. *IEEE transactions on biomedical engineering* 3 (1985), 230–236.
- J.W. Poore and B.M. Mann. 1991. Self-adjusting rate-responsive pacemaker and method thereof. (Dec. 24 1991). <https://www.google.co.uk/patents/US5074302> US Patent 5,074,302.
- Reza Sameni, Mohammad B Shamsollahi, Christian Jutten, and Gari D Clifford. 2007. A nonlinear Bayesian filtering framework for ECG denoising. *Biomedical Engineering, IEEE Transactions on* 54, 12 (2007), 2172–2185.
- Dominic P Searson. 2015. GPTIPS 2: an open-source software platform for symbolic data mining. *Handbook of Genetic Programming Applications* (2015).
- JW Shin, JH Yoon, and YR Yoon. 2001. Rate-adaptive pacemaker controlled by motion and respiratory rate using neuro-fuzzy algorithm. *Medical and Biological Engineering and Computing* 39, 6 (2001), 694–699.
- Hirofumi Tanaka, Kevin D Monahan, and Douglas R Seals. 2001. Age-predicted maximal heart rate revisited. *Journal of the American College of Cardiology* 37, 1 (2001), 153–156.
- Luu Anh Tuan, Man Chun Zheng, and Quan Thanh Tho. 2010. Modeling and verification of safety critical systems: A case study on pacemaker. In *Secure Software Integration and Reliability Improvement (SSIRI), 2010 Fourth International Conference on*. IEEE, 23–32.
- Jin Wang, Ronghua Chen, Xiangping Sun, Mary FH She, and Yuchuan Wu. 2011. Recognizing human daily activities from accelerometer signal. *Procedia Engineering* 15 (2011), 1780–1786.
- Bruce L Wilkoff, Joseph Corey, and Gordon Blackburn. 1989. A mathematical model of the cardiac chronotropic response to exercise. *Journal of Electrophysiology* 3, 3 (1989), 176–180.
- Pei Ye, Emilia Entcheva, Radu Grosu, and Scott A Smolka. 2005. Efficient modeling of excitable cells using hybrid automata. In *Proceedings of CMSB*, Vol. 5. 216–227.
- Håkan LS Younes, Marta Kwiatkowska, Gethin Norman, and David Parker. 2006. Numerical vs. statistical probabilistic model checking. *International Journal on Software Tools for Technology Transfer* 8, 3 (2006), 216–228.

A. PROBABILISTIC TIMED I/O AUTOMATA WITH DATA AND PRIORITIES

Formally, the semantics of a PTIOA network is defined over the corresponding *product PTIOA*. Before illustrating the definition of product PTIOA, we introduce the notion of *mutually consistent updates*, required to combine multiple update functions.

Definition A.1 (Mutually consistent updates). Let $\mathcal{R}' \subseteq \mathcal{R}$ be a set of update functions. Updates in \mathcal{R}' are *mutually consistent* if either they are defined over disjoint sets of variables, or they map common variables to the same values:

$$\forall r, r' \in \mathcal{R}'. \forall (v, \eta) \in \text{Dom}(r) \cap \text{Dom}(r'). r(v, \eta) = r'(v, \eta).$$

In this way, the combined application of a set of mutually consistent update functions \mathcal{R}' can be obtained as the union $\bigcup_{r \in \mathcal{R}'} r$ of the functions.

Example A.2. Consider the update functions $r = \{x := x + 1, y := 0\}$ and $r' = \{y := 1\}$. The valuation $\eta = \{x \mapsto 10, y \mapsto 1\}$ is updated by r and r' as $\eta[r] = \{x \mapsto 11, y \mapsto 0\}$ and $\eta[r'] = \{x \mapsto 10, y \mapsto 1\}$. The combined application $r \cup r'$ is not a function because of the inconsistent updates of y .

In the definition below, for action symbol $\alpha \in A$, we denote with $\overline{?}\alpha = !\alpha$ the co-action of $?\alpha$. Similarly, $\overline{!}\alpha = ?\alpha$ and $\overline{a} = a$, for $a \in \Sigma_{\text{in}} \cup \Sigma_{\text{out}}$.

Definition A.3 (Product PTIOA). The *product PTIOA* of a network $\mathcal{N} = (\{\mathcal{A}^1, \dots, \mathcal{A}^m\}, \mathcal{X}, \mathcal{D}, \Sigma = \Sigma_{\text{in}} \cup \Sigma_{\text{out}})$ is the PTIOA $\mathcal{N}_{\otimes} = (\mathcal{X}, \mathcal{D}, \vec{Q}, \vec{q}_0, \mathcal{P}(\Sigma_{\text{out}}), \rightarrow)$ where

- \mathcal{X} and \mathcal{D} are the sets of clocks and data variables of \mathcal{N} , respectively;
- the set of locations corresponds to the set of network modes \vec{Q} , with initial location $\vec{q}_0 \in \vec{Q}$;
- actions correspond to non-empty sets of output actions of \mathcal{N} ;
- $\rightarrow \subseteq \vec{Q} \times \mathcal{P}(\Sigma_{\text{out}}) \times \mathbb{N}^m \times \mathcal{B}(V) \times \mathcal{R} \times Q$ is the set of edges. It is composed of all elements $e = (\vec{s}, a, pr, g, r, \vec{t})$ for which there exists a non-empty subset of *synchronising components* $J \subseteq \{1, \dots, m\}$ and, for each $j \in J$, a *synchronising edge* $e^j = (s^j, a^j, pr^j, g^j, r^j, t^j) \in \rightarrow^j$ such that:
 - (1) source and target locations of e reflect those of synchronising edges: for all $j \in J$, $\vec{s}_j = s^j$ and $\vec{t}_j = t^j$;
 - (2) non-synchronising components remain in their source locations: for all $k \in \{1, \dots, m\} \setminus J$, $\vec{s}_k = \vec{t}_k$;
 - (3) actions of synchronising edges meet synchronisation rules: for all $j \in J$, either $a^j \in \Sigma_{\text{out}}$, or $a^j \in \Sigma_{\text{in}}$ and there exists $j' \in J$ such that $a^j = \overline{a^{j'}}$;
 - (4) a is given by the set of output actions fired by the synchronising edges: $a = \{a^j\}_{j \in J} \cap \Sigma_{\text{out}}$;
 - (5) the priority of e depends on those of synchronising edges and associates the lowest priority to non-synchronising components: $pr = (pr^1, \dots, pr^m) \in \mathbb{N}^m$, where, for $k \in \{1, \dots, m\} \setminus J$, $pr^k = +\infty$;
 - (6) g corresponds to the conjunction of the guard conditions of synchronising edges: $g = \bigwedge_{j \in J} g^j$; and
 - (7) r is the combined application of the updates of synchronising edges¹³: $r = \bigcup_{j \in J} r^j$.

Remark A.4. By condition (5), non-synchronising components are mapped to the lowest priority ($+\infty$) based on the assumption that components should always choose to fire one of the enabled edges over firing none and staying in the current location.

¹³We assume that updates $\{r_j\}_{j \in J}$ are mutually consistent.

Remark A.5 (Product PTIOA priorities ordering). By condition (5), the priorities of a product PTIOA are vectors of PTIOA priorities. The natural ordering on such vectors would be the component-wise order, i.e. for $pr_1 = (pr_1^1, \dots, pr_1^m)$ and $pr_2 = (pr_2^1, \dots, pr_2^m)$, $pr_1 \leq pr_2$ iff $\bigwedge_{j=1}^m pr_1^j \leq pr_2^j$. However, the component-wise order is not total (take e.g. $pr_1 = (1, 3)$ and $pr_2 = (2, 1)$). To ensure total ordering (needed in the PTIOA semantics), we consider instead the (total) lexicographic order \preceq . $pr_1 \preceq pr_2$ iff: $pr_1^1 < pr_2^1 \vee (pr_1^1 = pr_2^1 \wedge pr_1^2 < pr_2^2 \vee (pr_1^2 = pr_2^2 \wedge \dots \wedge pr_1^{m-1} < pr_2^{m-1} \vee (pr_1^{m-1} = pr_2^{m-1} \wedge pr_1^m \leq pr_2^m) \dots))$

Following the lexicographic order, $pr_1 = (1, 3) \preceq pr_2 = (2, 1)$.

Remark A.6. By condition (6), the satisfaction probability of a guard g is the product of the satisfaction probabilities of the guards g^j of the synchronising edges (see Equation 1), since we assume independence of distributions (see Remark 2.1):

$$Pr(\eta \models g) = \prod_{j \in J} Pr(\eta \models g^j). \quad (10)$$

Given a product PTIOA guard $g = \bigwedge_{j \in J} g^j$, where, for $j \in J$, $g^j = \bigwedge_i x_{i,j} \geq t_{i,j}$, the random waiting time, $t(g, \eta)$, for g to be satisfied under valuation $\eta \in \mathcal{V}(V)$ is given by

$$t(g, \eta) = \max(0, \max\{t_{i,j} - \eta(x_{i,j})\}_{i,j}). \quad (11)$$

Semantics. Let \mathcal{N} be a PTIOA network, $\mathcal{N}_\otimes = (\mathcal{X}, \mathcal{D}, \vec{Q}, \vec{q}_0, \mathcal{P}(\Sigma_{\text{out}}), \rightarrow)$ its product PTIOA and $\rho = (\vec{q}_0, \eta_0) \xrightarrow{e_0, t_0} (\vec{q}_1, \eta_1) \xrightarrow{e_1, t_1} \dots$ be a path of \mathcal{N} , as explained in Section 2. Below, we formally describe the probabilities of random variables $\rho[i] = (\vec{q}_i, \eta_i)$, t_i and e_i .

— For $e = (\vec{s}, a, pr, g, r, \vec{t}) \in \rightarrow$ and $d \in \mathbb{R}^{\geq 0}$,

$$Pr(t(e, \eta_i) \leq d \mid \vec{s} = \vec{q}_i) = Pr(\eta_i + d \models g) \quad \text{and} \quad Pr(t(e, \eta_i) = +\infty \mid \vec{s} \neq \vec{q}_i) = 1,$$

where $t(e, \eta_i)$ is a random variable describing the waiting time for e to be enabled under valuation η_i (see Equations 2 and 11). If e is not an outgoing edge of \vec{q}_i , it cannot be fired from the current state, and thus, $t(e, \eta_i) = +\infty$ with probability 1.

— The waiting time t_i in $\rho[i]$ is the shortest waiting time among the edges of \mathcal{N}_\otimes : $t_i = \min_{e \in \rightarrow} t(e, \eta_i)$.

— Let $E_i = \{e \in \rightarrow \mid t(e, \eta_i) = t_i\}$ be the set of edges with shortest delay. Then, the fired edge $e_i = (\vec{s}, a, pr, g, r, \vec{t})$ is the element of E_i with maximum priority: $e_i = \arg \min_{e \in E_i} pr(e)$, where $pr(e)$ is the priority of e . Finally, the next state is consistent with the target and the update function of e_i : $\vec{q}_{i+1} = \vec{t}$ and $\eta_{i+1} = (\eta_i + t_i)[r]$.

Example A.7 (Semantics of a PTIOA network). Figure 17 illustrates the product PTIOA of the network of Example 2.6. Synchronisations between the SA node and the atrium automata occur from location (SA_1, A_2) to (SA_2, A_4) (action !Abeat), and from (SA_2, A_6) to (SA_1, A_1) (action !NextAtrBeat). From (SA_1, A_1) to (SA_2, A_2) , the two automata proceed independently without synchronisation. In all the other edges, one of the two automata is inactive, which is also visible by the associated $+\infty$ priority. Recall that edges of a product PTIOA are labelled with the set of output actions performed by the PTIOA components.

The regular heart behaviour, where periodically the atrium becomes excitable and later receives the stimulus from the SA node, is described by the cycle $(SA_1, A_1) \rightarrow (SA_1, A_2) \rightarrow (SA_2, A_4) \rightarrow (SA_2, A_5) \rightarrow (SA_2, A_6) \rightarrow (SA_1, A_1)$. There is one deadlock location, (SA_2, A_2) , reached when the SA node sends the impulse before the atrium has become excitable. However, unlike the regular heart cycle, the deadlock location has a

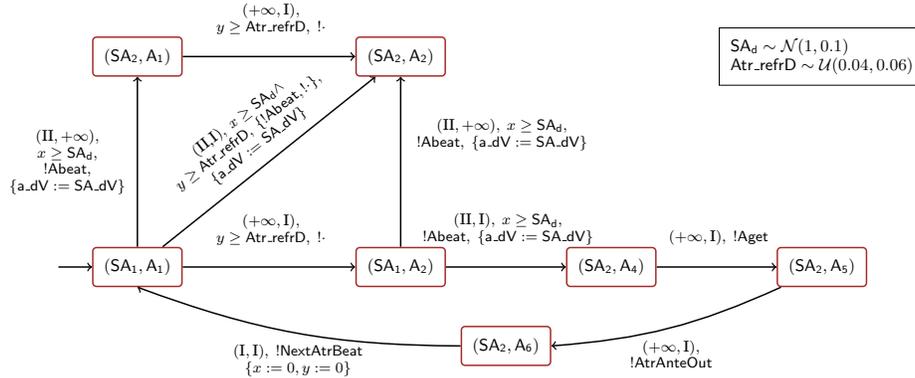


Fig. 17: Product PTIOA of the network in Figure 1. Empty updates and trivially true guards are omitted. Action singletons $\{\alpha\}$ are just indicated as $!\alpha$. Location (SA_i, A_j) corresponds to the network mode where the i -th location of the SA node and the j -th location of the atrium are active.

very small probability to be reached. We show this by inspecting the probabilities of the edges out of the initial location (SA_1, A_1) .

- $t \leq 0.04$: the probability that the edges to (SA_1, A_2) and (SA_2, A_2) are enabled within time t is 0, because $\mathcal{U}^{\leq}(0.04, 0.06)(0.04) = 0$. The probability of stepping to (SA_2, A_1) is only slightly higher, and is given by $\mathcal{N}^{\leq}(1, 0.1)(0.04) \approx 4 \times 10^{-22}$.
- $t \leq 0.06$: in this case, the probability that the edge to (SA_1, A_2) is enabled within time t is 1, since $\mathcal{U}^{\leq}(0.04, 0.06)(0.06) = 1$. The probability for the other two edges remains insignificant, being equal to $\mathcal{N}^{\leq}(1, 0.1)(0.06) \approx 2.72 \times 10^{-21}$.

In principle, the deadlock location can be also reached from (SA_1, A_2) if the SA node chooses not to synchronise on $!Abeat$. But note that this edge has the same random delay but strictly lower priority than the edge leading to (SA_2, A_4) , meaning that can never be taken. Also note that the deadlock location is a side-effect of considering the product of just two components. In the full heart model never ends up in a deadlock location, since (SA_2, A_2) is exited with the arrival of a retrograde wave (e.g. paced impulse).

B. HYBRID AUTOMATA STOCHASTIC LANGUAGE (HASL)

B.1. Linear Hybrid Automata

The first component of an HASL formula is a kind of hybrid automaton called *synchronised Linear Hybrid Automaton (LHA)*. Below, we introduce the main features of LHAs, especially in relation to the PTIOA formalism introduced in Section 2.

Locations. In addition to the initial location, an LHA also contains a final location. Since the LHA synchronises with infinite paths of the PTIOA network under analysis, the final location determines when to terminate and accept the current path. On the other hand, if at any point the LHA cannot synchronise with the PTIOA network, the current path is terminated and rejected.

Variables. Similarly to PTIOAs, LHAs are characterised by clock and data variables. Unlike PTIOAs where data variables can only be modified through updates, in LHAs data variables can evolve over time according to rate functions defined at each location and called *flows*. In an LHA, flows are restricted to linear functions. Similarly, LHA

updates are linear functions over the variables, while PTIOAs support arbitrary real-valued functions.

Guards. LHA guards are deterministic and defined by linear constraints of the form $\sum_i a_i x_i + c \bowtie 0$, where $a_i, c \in \mathbb{R}$, x_i is a clock or data variable and $\bowtie \in \{=, \leq, \geq, <, >\}$. In contrast, PTIOA guards are probabilistic and restricted only to clock variables.

Actions and synchronisation. Given that an LHA must synchronise with the output actions fired by the PTIOA network, all LHA actions are implicitly input actions and are drawn from the same set of action symbols of the PTIOA network. Specifically, each LHA edge is labelled with a set of actions and synchronisation happens as follows. Let $\rho = (\vec{q}_0, \eta_0) \xrightarrow{e_0, t_0} (\vec{q}_1, \eta_1) \xrightarrow{e_1, t_1} \dots$ be the current path of the network. Consider an LHA edge with guard g and set of actions E . Starting from position i of ρ , the LHA edge can fire if the following conditions hold:

- (1) the guard is satisfied: $g \models \eta' + t_i$, where η' is the current valuation of the LHA variables; and
- (2) E contains actions that can synchronise with those fired by the PTIOA network: $\exists a \in E. \exists a' \in a(e_i). a = \overline{a'}$, where e_i is the fired edge of the underlying product PTIOA and $a(e_i)$ denotes the corresponding set of actions.

Finally, some LHA edges are allowed to proceed autonomously, that is, without synchronisation. Such edges are labelled with the symbol $\#$.

Priorities. Similarly to PTIOAs, LHA edges are labelled with priorities that establish a total ordering, thus avoiding non-determinism when multiple LHA edges are allowed to fire.

Remark B.1. In the original LHA definition of [Ballarini et al. 2015], LHA edges are labelled with sets of actions of the underlying system and have no priorities. Determinism is ensured by mutually exclusive guards and action sets. In our case, an action of a PTIOA network is the set of PTIOA actions fired by some network components, meaning that LHA edges would have to be labelled with sets of sets of PTIOA actions. Our simplification considers instead just sets of PTIOA actions and introduces priorities to resolve non-determinism.

B.2. HASL syntax

The syntax of an HASL expression Z is defined by the following grammar:

$$\begin{aligned} Z &::= c \mid P \mid \mathbb{E}[Y] \mid Z \circ Z \\ Y &::= c \mid Y \circ Y \mid \text{LAST}(y) \mid \text{MIN}(y) \mid \text{MAX}(y) \mid \text{INT}(y) \mid \text{AVG}(y) \\ y &::= c \mid x \mid y \circ y \end{aligned}$$

where $c \in \mathbb{R}$ and $\circ \in \{+, -, \times, /\}$ and x is a variable of the associated LHA. y is an arithmetic expression over LHA variables. Expression Y is called a *path variable*, and is defined from the following path-dependent random variables: $\text{LAST}(y)$, the value of y at the final state; $\text{MIN}(y)$ ($\text{MAX}(y)$), the minimum (maximum) value of y along the path; $\text{INT}(y)$, the integral over time of y along the path; and $\text{AVG}(y)$, the average value of y along the path. Finally, an HASL formula Z is an arithmetic expression built on top of $\mathbb{E}[Y]$ (the expectation of Y) and P (the probability that a path is accepted).

C. PATIENT-SPECIFIC QT-RR RELATIONSHIPS

C.1. Expressional complexity of MSR functions

Functions estimated through MSR are of the form:

$$f(\mathbf{x}) = a_0 + a_1 f_1(\mathbf{x}) + \dots + a_k f_k(\mathbf{x}) \quad (12)$$

and are encoded into the optimization problem as trees [Searson 2015]. The expressional complexity of f , $ec(f)$, is given by

$$ec(f) = \sum_{i=1}^k \sum_{N \in \text{nodes}(f_i)} |\text{nodes}(\text{subtree}(N))|, \quad (13)$$

where $\text{nodes}(f_i)$ is the set of nodes of the tree encoding for f_i , and $\text{subtree}(N)$ is the subtree of f_i with root N . In other words, $ec(f)$ is obtained by summing, for each tree f_i , the node count of itself and its possible sub-trees. In this way, functions characterised by deep trees have higher complexity than functions with flatter and more balanced trees. The complexity of each function f_i is controlled by the user-defined parameter dep , i.e. the maximum tree depth.

Example C.1 (Expressional complexity). Consider the two trees depicted in Figure 18. They encode for the single variate function:

$$f(x) = f_1(x) + f_2(x) = \sqrt[3]{2x} + \exp(\sqrt[2]{x} + x^3).$$

To compute the expressional complexity of f , we need to sum the number of nodes for each subtree of f_1 and f_2 . It is easy to verify that the node count is 9 for f_1 and 17 for f_2 , yielding $ec(f) = 26$.

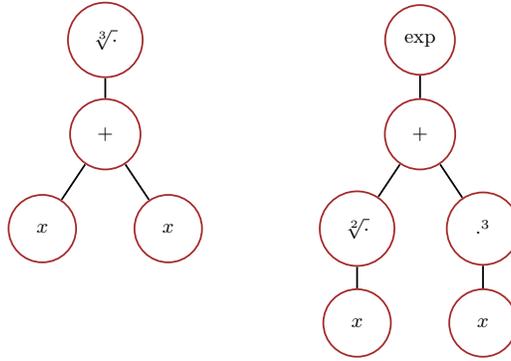


Fig. 18: Example of multi-gene tree encoding for functions $f_1(x) = \sqrt[3]{2x}$ (left) and $f_2(x) = \exp(\sqrt[2]{x} + x^3)$ (right).

C.2. Estimation of QT-RR relationships

Patient-specific estimation. We perform an in-depth analysis of optimal QT-RR laws for the three subjects introduced in Section 4.1. For each subject, we built training and validation datasets, extracting 500 QT-RR pairs for the former, and 100 for the latter. Regression for the classical QT-RR models (linear, quadratic, hyperbolic, cubic and exponential) was performed using the MATLAB curve fitting toolbox. For multigene symbolic regression (MSR), we used the MATLAB package GPTIPS [Searson 2015].

In the Tables II–VII, we report numerical data for goodness of fit along with the analytical form and expressional complexity of the Pareto-optimal models found through symbolic regression. Plots of the Pareto fronts are shown in Figure 19.

Curve fitting results (see Figure 9) demonstrate the patient-specificity of the optimal QT-RR laws. For subject 1, the exponential and the hyperbolic models provide the best fit among the five classical regression models, since they adapt well to the concave shape of the subject’s QT-RR relationship (see plot (a)). This agrees with the Pareto-optimal models found through MSR, which are all characterised by a concave behaviour (plot (d)). On the other hand, the linear law is optimal for subject 2, even though the quadratic, cubic and exponential models produces comparable results (see plot (b)). Similarly, the shapes of all Pareto-optimal models exhibit small convexity, visible in plot (e). In plot (c) we observe that QT-RR couples for subject 3 are optimally fitted by the exponential and the cubic laws. In contrast, Pareto-optimal models 2-4 for this subject (plot (f)) exhibit convex behaviour in the first half of the plot, and a concave shape in the second-half.

Importantly, for all subjects MSR is able to find at least one Pareto-optimal model outperforming the five classical regression models for both training and validation sets, hence proving greater adaptability to data and better accuracy. However, the main drawback of these models is their lack of extrapolation accuracy, since predictions outside the training set may turn out to be unreliable. For instance, in models 4-6, subject 2 (plot (e)), RR intervals below 0.4 seconds are mapped to excessive QT values. Similarly, models 3-5, subject 1 (plot (d)) and models 2-4, subject 3 (plot (f)) produce poor predictions for RR intervals above 0.8 seconds. In this regard, the hyperbolic and the exponential models are the most robust, since they both quickly saturate when approaching resting RR intervals. Indeed, the QT interval should ideally reach a plateau at resting rate RR intervals. This behaviour can be easily reproduced for all the other models, by choosing appropriate upper and lower bounds for the predicted RR interval.

In general, the advantage of MSR is the ability to derive, in a fully automatic fashion, highly accurate and patient-specific QT-RR laws, albeit suffering from poor extrapolation accuracy. Therefore, it represents an effective alternative to the current practice of choosing a fixed a-priori mathematical model, which cannot yield accurate predictions for all patients.

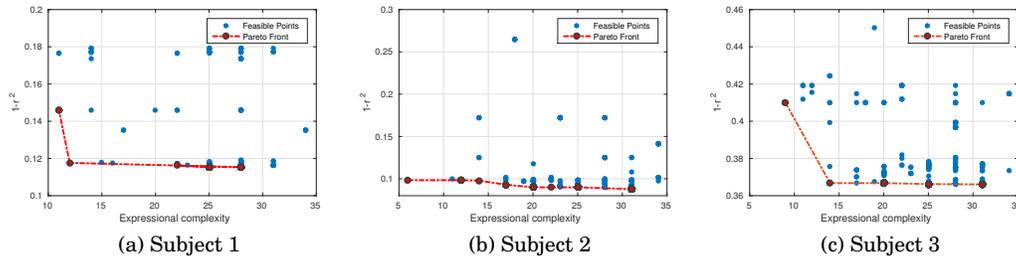


Fig. 19: Pareto analysis of trade-off among expressions complexity and r^2 (computed on the *training* set) for the points explored in MSR optimization for the three subjects. Blue dots represents all the feasible models explored by the algorithm, whereas red dots are the projection of the (observed) Pareto-optimal models into the objective space.

Model	Training set		Validation set	
	RMSE	r^2	RMSE	r^2
Linear	0.0244	0.8359	0.0247	0.8243
Quadratic	0.0232	0.8522	0.0234	0.8417
Cubic	0.0228	0.8566	0.0230	0.8467
Hyperbolic	0.0211	0.8772	0.0210	0.8723
Exponential	0.0208	0.8810	0.0204	0.8800
Pareto 1	0.0230	0.8541	0.0232	0.8439
Pareto 2	0.0206	0.8824	0.0200	0.8842
Pareto 3	0.0205	0.8838	0.0196	0.8891
Pareto 4	0.0204	0.8846	0.0192	0.8932
Pareto 5	0.0204	0.8846	0.0192	0.8938

Table II: Results of curve fitting and symbolic regression for subject 1. In bold, the best RMSE and r^2 values.

Model	Training set		Validation set	
	RMSE	r^2	RMSE	r^2
Linear	0.0290	0.9014	0.0378	0.8521
Quadratic	0.0290	0.9012	0.03839	0.8475
Cubic	0.0291	0.9007	0.0386	0.8454
Hyperbolic	0.0308	0.8881	0.0418	0.8191
Exponential	0.0289	0.9017	0.0381	0.8500
Pareto 1	0.0291	0.9002	0.0378	0.8525
Pareto 2	0.0289	0.9012	0.0380	0.8502
Pareto 3	0.0280	0.9075	0.0366	0.8617
Pareto 4	0.0276	0.9104	0.0343	0.8780
Pareto 5	0.0275	0.9105	0.0340	0.8806
Pareto 6	0.0272	0.9126	0.0335	0.8836

Table III: Results of curve fitting and symbolic regression for subject 2. In bold, the best RMSE and r^2 values.

Model	Training set		Validation set	
	RMSE	r^2	RMSE	r^2
Linear	0.0593	0.6091	0.0586	0.6267
Quadratic	0.0586	0.6185	0.0578	0.6367
Cubic	0.0585	0.6204	0.0576	0.6388
Hyperbolic	0.0590	0.6127	0.0581	0.6324
Exponential	0.0583	0.6238	0.0573	0.6427
Pareto 1	0.0606	0.5900	0.0600	0.6078
Pareto 2	0.0573	0.6332	0.0565	0.6525
Pareto 3	0.0573	0.6338	0.0565	0.6528
Pareto 4	0.0573	0.6339	0.0565	0.6527

Table IV: Results of curve fitting and symbolic regression for subject 3. In bold, the best RMSE and r^2 values.

Model	Complexity	Analytic form
Pareto 1	11	$0.525 \sqrt[3]{RR} + 0.525 \sqrt[2]{RR} - 0.511$
Pareto 2	12	$2.37 \sqrt[3]{RR} + 2.37 \sqrt[2]{RR} + 2.38RR - 2.08$
Pareto 3	22	$-2.7RR^2 - 2.7\sqrt{RR} + 0.263$
Pareto 4	25	$12e^{RR} + 12RR - 18.3RR^2 - 18.3\sqrt{RR} - 8.06$
Pareto 5	28	$14.4e^{RR} + 13.4RR - 21.3RR^2 - 21.3\sqrt{RR} - 9.69$

Table V: Pareto-optimal models found through symbolic regression for subject 1.

Model	Complexity	Analytic form
Pareto 1	6	$0.976\sqrt{e^{RR}} - 1.01$
Pareto 2	14	$0.38RR + 0.38\sqrt[3]{RR} + 0.157\sqrt{e^{RR}} - 0.445$
Pareto 3	17	$26.7\sqrt{e^{RR}} - 7.54e^{RR} - 7.54\sqrt[3]{RR} - 15.6$
Pareto 4	20	$35.6RR - 22.6\sqrt[3]{RR} - \sqrt{e^{RR}} + 46.4$
Pareto 5	22	$8.73 - 8.41RR^2 - 30.9\sqrt{RR} + 30.9RR$
Pareto 6	31	$-1.91\frac{e^{RR}}{RR} + 0.45\frac{1}{RR^2} - 0.39\frac{1}{RR} + 5.55$

Table VI: Pareto-optimal models found through symbolic regression for subject 2.

Model	Complexity	Analytic form
Pareto 1	9	$0.262e^{RR} - 0.165$
Pareto 2	14	$13.6e^{RR} - 9.33\sqrt{RR}e^{RR} - 11.2$
Pareto 3	25	$9.59RR - 10.4\sqrt[3]{RR} - 3.45RR^2 + 4.57$
Pareto 4	31	$18.5RR - 18.5\sqrt[3]{RR} - 9.73RR^2 + 2.22RR^3 + 7.87$

Table VII: Pareto-optimal models found through symbolic regression for subject 3.

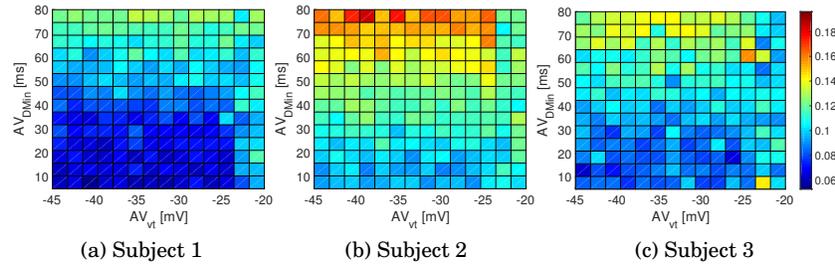


Fig. 20: Analysis of the objective function profile for different values of the hidden parameters considered.

D. ESTIMATION OF HIDDEN PARAMETERS

We perform a parametric analysis of the objective function used for the estimation of hidden parameters of the heart model. The objective function describes the statistical distance between the input ECG signal and the synthetic ECG signal. Even though for simulation purposes we value having some good parameter estimation and are not interested in knowing all admissible estimations, we investigate the profile of the objective function for the presence of multiple minima, which could potentially lead to ambiguity in the estimation problem.

The hidden parameters we aim to estimate are AV_{anteDMin} , $AV_{\text{retroDMin}}$ and AV_{Vt} . The former two defines the minimum propagation time in the AV node for signals following the antegrade and retrograde conduction pathways, respectively. In the formulation of the optimization we assume these two conduction delay two be the same [Barbot et al. 2015b] and collectively call them AV_{DMin} . Parameter AV_{Vt} represents the depolarisation voltage threshold of the AV node.

Figure 20 depicts the heat-maps for the three subjects. These were computed on a uniformly spaced grid of the two hidden parameters we aim to estimate as a solution of the minimisation problem. In all three cases we have a significantly better fitness (i.e. lower values of the objective function) in the lower part of the figures. For subject 3 we observe the only local minimum away from the global one. This, however, yields almost double the objective function value of the observed global minimum.

E. VERIFICATION PROPERTIES

Absence of Induced Tachycardia. We describe the LHA associated to the HASL property $\phi_{\text{NO_TC}}$ (see Equation 7). The LHA is depicted in Figure 21.

Recall that the rate-adaptive pacemaker updates the value of the pacing rate at a fixed frequency. Let *window* be the update period, typically set to 2.5 seconds in the blending algorithm (see Section 4.3.1). The aim of the LHA is to count, through variable *nOKs*, the number of adequate rate updates, i.e. inducing an effective heart rate that does not exceed the resting rate (parameter RR_{rest}). Variable *nUpd* keeps track of the total number of rate updates.

For this purpose, the LHA computes the average RR interval during each updating window through variables *Avg*, storing the cumulative sum of the RR intervals and *Nbeats*, i.e., the number of ventricular beats. The LHA uses two clocks, t_1 and *RR*, representing the time elapsed since the last update and since the last ventricular beat, respectively. These variables are updated accordingly, every time a ventricular beat occurs, captured by firing the bottom-left edge (*?Vget* denotes an intrinsic beat, *?VP* a paced beat).

At the end of the update period ($t_1 \geq \text{window}$), the current adaptive rate is considered adequate and *nOKs* is increased if $\text{Avg} \geq \text{Nbeats} \cdot \text{RR}_{\text{rest}}$, that is, if the average RR does not exceed RR_{rest} , corresponding to the firing of the top-right edge. Otherwise, only *nUpd* is increased (top-left edge).

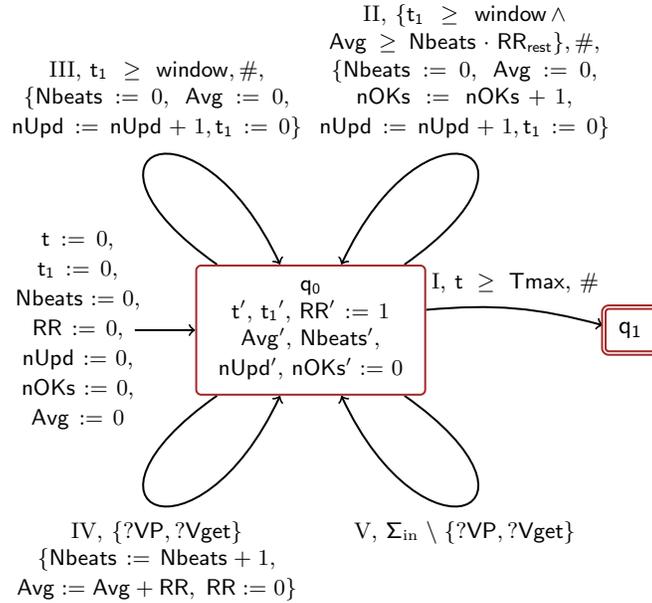


Fig. 21: LHA for evaluation of absence of sensor induced tachycardia.

Regularity. The LHA for estimating the HASL formula ϕ_{Reg} heart rate regularity (see Equation 9) is illustrated in Figure 22.

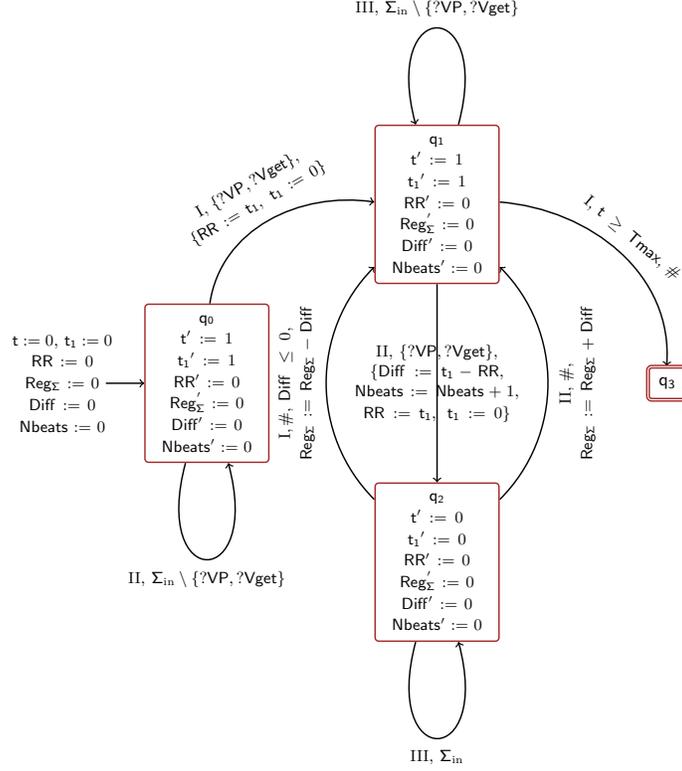


Fig. 22: LHA for the estimation of heart rate regularity.

Derivation of Reg^ .* Let $\mathcal{N}(\mu, \sigma^2)$ be the Gaussian distribution of the RR intervals estimated for an healthy subject specific to a particular activity stage. We hence have that, for each beat i , the RR associated to the i -th and $i + 1$ -th beats, RR_i and RR_{i+1} respectively, are independent and identically distributed Gaussian random variables. Hence, the difference variable $D_i = RR_{i+1} - RR_i$ follows a Gaussian distribution with zero mean and variance $2\sigma^2$. In turn, variables $|D_i|$ are independently and identically distributed half-normal random variables of scale parameter $2\sigma^2$. These are characterised by expectation $\mathbb{E}[|D_i|] = 2\sqrt{\frac{\sigma^2}{\pi}}$, for all i .

We define Reg^* as the expected value of the regularity measure Reg . By above considerations and by the definition of Reg of Equation 5.2.1, we obtain:

$$\text{Reg}^* = \mathbb{E}[\text{Reg}] = \mathbb{E}\left[\frac{1}{n-1} \sum_{i=1}^{n-1} |D_i|\right] = \frac{1}{n-1} \sum_{i=1}^{n-1} \mathbb{E}[|D_i|] = 2\sqrt{\frac{\sigma^2}{\pi}}. \quad (14)$$

Endless-loop tachycardia: open- vs closed-loop. We report the detailed setup and full numerical results of the endless-loop experiment. We consider paths of 2 minutes, with a faulty accelerometer sensor in the first 30 seconds. We evaluate the probability that in the remaining 90 seconds the pacemaker does not induce tachycardia, encoded by the HASL formula $\phi_{\text{NO.TC}}$ of Equation 7.

	Subject 1			Subject 2			Subject 3		
	Exp. 1	Exp. 2	Exp. 3	Exp. 1	Exp. 2	Exp. 3	Exp. 1	Exp. 2	Exp. 3
$\phi_{\text{NO_TC}}$	0.857	0.878	0.695	0.686	0.628	0.267	0.825	0.799	0.306
99% C.I.	/	± 0.002	± 0.012	/	± 0.002	± 0.013	/	± 0.003	± 0.007
Min	0.857	0.857	0.029	0.686	0.625	0	0.825	0.790	0
Max	0.857	0.885	0.886	0.686	0.633	0.686	0.825	0.813	0.771

Table VIII: Comparison between open-loop (experiments 1 and 2) and closed-loop (experiment 3) architectures in simulated endless-loop tachycardia. $\phi_{\text{NO_TC}}$ (see Equation 7) estimates the probability that the pacemaker does not induce tachycardia. Minimum and maximum observed values for $\phi_{\text{NO_TC}}$ are reported, including 99% confidence intervals. Results for experiments 2 and 3 are obtained with 1000 trajectories for each subject. Experiment 1 is deterministic and thus considers one trajectory.

Since the pacing rate is adjusted smoothly by the blending algorithm, it cannot be set to resting levels immediately after the initial 30 seconds of sensor malfunction. This implies that $\phi_{\text{NO_TC}}$ cannot reach probability 1 even when no PMT episodes occur. Specifically, the pacing rate period t_{AR} is updated every 2.5 seconds, corresponding to a total of 36 updates in a 90 seconds window. Since the pacemakers sets the value of t_{AR} to the median of the last 5 rates suggested by the blending algorithm, we have that the effect of the sensor malfunction (leading to wrongly high pacing rates) will completely fade out after 4 rate updates, leading to an upper bound of 0.889 for $\phi_{\text{NO_TC}}$. To account for this effect, the results reported in Section 5.2 are scaled by such obtained upper bound. Below, we report instead the unscaled probability values and use the value 0.889 as a baseline level representing total absence of pacemaker-induced tachycardia.

For each subject, we performed the following three experiments:

- Experiment 1: open-loop model using fixed data from an ECG signal recorded at rest;
- Experiment 2: open-loop model using synthetic QT data generated offline using resting parameters;
- Experiment 3: closed-loop model using synthetic QT data generated online using resting parameters.

For all the experiments, we generate synthetic accelerometer signals using activity parameters in the first 30 seconds (to reproduce the fault) and resting parameters in the remaining 90 seconds. Results are listed in Table VIII. Experiments 1 and 2 yield very similar results. For subjects 1 and 3, probabilities are very close to the upper-bound 0.889, whereas they are slightly lower (-20%) for subject 2. Nevertheless, both the 99% confidence interval and the range of the distributions are very narrow, suggesting that PMT hardly occurs in the open-loop model.

On the other hand, results for Experiment 3 (closed-loop) indicate that PMT is successfully reproduced in all subjects. Indeed, we obtain substantially smaller mean values for $\phi_{\text{NO_TC}}$, that is, higher probabilities of pacemaker-induced tachycardia. However, the wide ranges of observed $\phi_{\text{NO_TC}}$ values (rows Min and Max in Table VIII) hint at the randomness of this phenomenon: PMT occurs and is not terminated for some trajectories, while it never occurs for some others. In general, SMC results agree with the evidence that PMT episodes are rare in VVIR pacemakers and are strongly patient-specific.

F. MODELLING OF STATEFLOW CHARTS

The formalism of PTIOAs is captured by a subset of the Simulink/Stateflow modelling language. As described in [Barbot et al. 2015b], we do not consider the state hierarchy feature of Stateflow, 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. PTIOAs 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, PTIOAs do not support junctions and multiple actions associated to a single edge. Unlike PTIOAs, Stateflow diagrams do not support the definition of arbitrary clocks and clock updates. In particular, each Stateflow component only possesses an implicit clock, which is reset to 0 whenever an edge is taken, and guards are specified through Stateflow temporal operators. Specifically, we use the operator `after(t)` in place of the guard $x \geq t$, `before(t)` for $x \leq t$ and `at(t)` for $x = t$, where x is the implicit clock and t is a time value that can be specified as a function over data variables and parameters.

We implement random variables and delays in Stateflow via the use of local variables. The value of each local variable is freshly sampled every time it is used in the Stateflow model. Sampling is implemented via the use of MATLAB Function blocks.

F.1. Stateflow Components of the Heart Model

In this section we provide a description of the PTIOA components of the heart model not already described in Example 2.6, expressed as Stateflow charts. For further details we refer to [Barbot et al. 2015b].

Atrial ectopic rhythm generation (Figure 23). Component AEctopic generates ectopic beats with rate AEcto.d. 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.

As for the SANode component described in Example 2.6, the potential is transmitted to the Atrium component through the output action `!Abeat`. The strength of the action potential is stored in variable `a_dV` and is determined by parameter `AEcto_dV`.

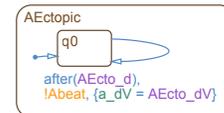


Fig. 23: AEctopic component.

Conductors (Figure 24). The AAVConductor (Atrium-AV node conductor) and AVVconductor (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. Delay parameter `AAV_anteD` is sampled from the discrete distribution D_f associated to half duration of the P waves; since it represents the time delay from `!Aget` to the start of the propagation of the action potential into the AV node.

Similarly, the AVVConductor is an abstraction of the nodes connecting the AV node and the ventricles, namely, the bundle of His and the Purkinje fibres. The AVVConductor is responsible for the delay between the time in which the action potential leaves the AV node and the time in which it is detected by the ventricular lead of the pacemaker. Hence, parameter `AVV_anteD` is implemented as a random variable distributed accordingly to the QR intervals.

Ventricle and ventricular rhythm generation (Figure 25). 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.

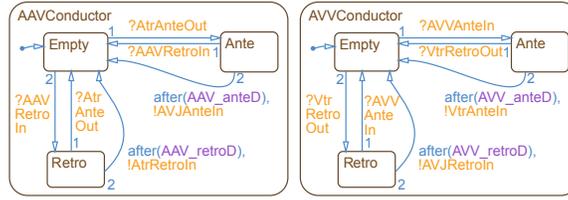


Fig. 24: AAVConductor (left) and AVVConductor (right) components.

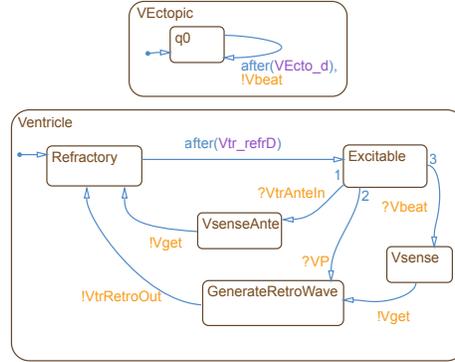


Fig. 25: Ventricular ectopic rhythm generation (top) and Ventricle component (bottom).

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. 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). As explained in Section 4.2.1, we implement the delay parameter Vtr_refrD as a random variable distributed accordingly to observed RT intervals.

AV node (Figure 26). The AV node plays a crucial role in the conduction of cardiac waves, and thus requires precise modelling of its action potential. It is 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. In Table IX, we report the main functions used in the model to implement the AV conduction delay.

The two delay parameters AV_anteDMin and AV_retroDMin, and the voltage threshold parameter AV_Vt are “hidden” parameters estimated as detailed in Section 4.2.1.

Inside the AV node the evolution of the action potential can be driven by: i) *intrinsic depolarisation*, and ii) *excitation from external stimulus*.

In the first case, the AV node does not receive any external stimulus. In the Recovery state the action potential increases linearly starting from the resting potential AV_Vr with a slow rate. This slow depolarisation phase is implemented in the model via the delay parameter avj_t4, computed through the function timeToDep. If avj_t4 seconds elapse before an external stimulus reaches the AVJ component, the transition from Recovery to Refractory is fired.

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

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

The second case is the most common one, where a signal reaches the AVJ during the recovery period. Depending on the intensity of the signal reaching the AVJ, the action potential increases promptly and, under healthy condition, is usually sufficient to trigger the depolarisation of the AV node. Depending on the type of conduction, in the AVJ component the stimulus is detected through input action AVJAnteIn (antegrade conduction) or AVJRetroIn (retrograde conduction), 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, hence always firing the transition from Retro to Refractory. Whereas, the impulse strength from the Atrium (stored in variable a_dV) is used to update the time spent in the Ante state through function initRefrPeriod.

The residence time in the Refractory state is determined by the time previously spent in the Recovery state (variable avj_tRec) through function initRefrPeriod(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. At the end of the refractory period the edge linking the Refractory and Recovery state is fired.

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

Stage	Minutes	METS	Subject 1 HR	Subject 2 HR	Subject 3 HR	Virtual population HR
1	3	4	93	85	82	89
2	3	7	106	99	94	103
3	3	10	119	113	106	116
4	3	11	123	118	110	121
5	3	13	132	128	118	130
6	3	15	140	137	126	139
7	3	19	157	156	142	158

Table X: First three columns: details on the time length and METS for each activity stage of the Bruce protocol. Remaining columns: estimated heart rates (BPM) for the virtual patients considered in Section 4.1.

REFERENCES

- Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. 2013. A public domain dataset for human activity recognition using smartphones. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN*.
- Paolo Ballarini, Benoit Barbot, Marie Dufлот, Serge Haddad, and Nihal Pekergin. 2015. HASL: A new approach for performance evaluation and model checking from concepts to experimentation. *Performance Evaluation* 90 (2015), 53–77.

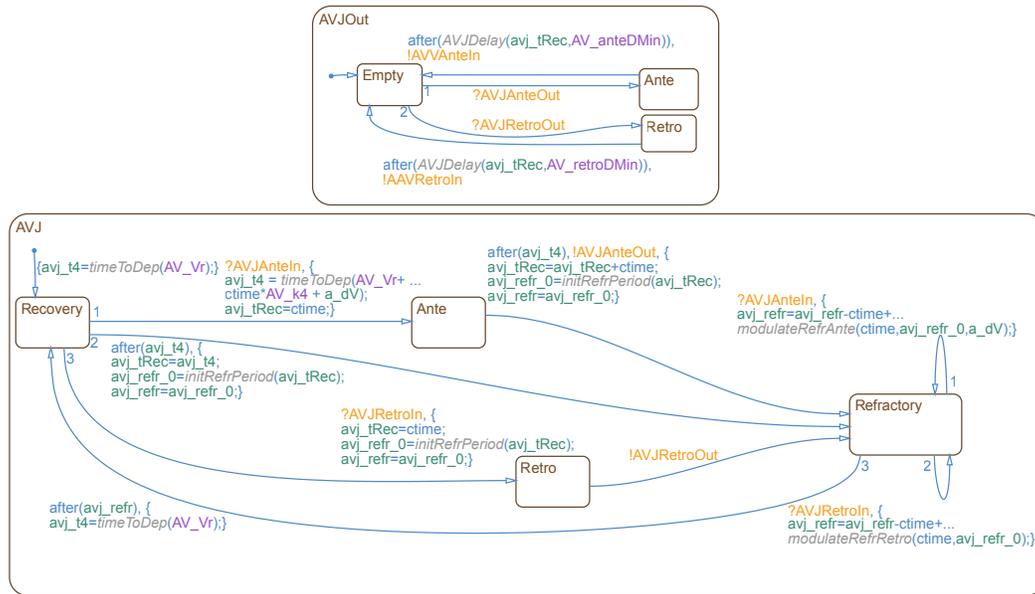


Fig. 26: Components modelling AV node.

Kazunori Ohkawara, Yoshitake Oshima, Yuki Hikiyara, Kazuko Ishikawa-Takata, Izumi Tabata, and Shigeo Tanaka. 2011. Real-time estimation of daily physical activity intensity by a triaxial accelerometer and a gravity-removal classification algorithm. *British Journal of Nutrition* 105 (6 2011), 1681–1691.

Esther Pueyo, Peter Smetana, Pablo Laguna, and Marek Malik. 2003. Estimation of the QT/RR hysteresis lag. *Journal of electrocardiology* 36 (2003), 187–190.

Bruce L Wilkoff, Joseph Corey, and Gordon Blackburn. 1989. A mathematical model of the cardiac chronotropic response to exercise. *Journal of Electrophysiology* 3, 3 (1989), 176–180.