A Simulink Hybrid Heart Model for Quantitative Verification of Cardiac Pacemakers

Taolue Chen
Department of Computer Science
University of Oxford, UK

Marco Diciolla
Department of Computer Science
University of Oxford, UK

Marta Kwiatkowska
Department of Computer Science
University of Oxford, UK

Alexandru Mereacre
Department of Computer Science
University of Oxford, UK

ABSTRACT

We develop a novel hybrid heart model in Simulink that is suitable for quantitative verification of implantable cardiac pacemakers. The heart model is formulated at the level of cardiac cells, can be adapted to patient data, and incorporates stochasticity. It is inspired by the timed and hybrid automata network models of Jiang *et al* and Ye *et al*, where probabilistic behaviour is not considered. In contrast to our earlier work, we work directly with action potential signals that the pacemaker sensor inputs from a specific cell, rather than ECG signals. We validate the model by demonstrating that its composition with a pacemaker model can be used to check safety properties by means of approximate probabilistic verification.

Categories and Subject Descriptors

I.6.4 [**Computing Methodologies**]: SIMULATION AND MODELING—*Model Validation and Analysis*

General Terms

Verification

Keywords

Hybrid systems, Quantitative verification, Pacemaker

1. INTRODUCTION

Today's implantable medical devices are increasingly often controlled by embedded software and rigorous software design methodologies are needed to ensure their safe operation and to avoid costly device recalls. The natural models for medical devices, such as cardiac pacemakers [6], GPCA infusion pumps [8] and continuous glucose monitors [16], are stochastic hybrid systems: they involve discrete mode switching and nonlinear continuous flows, e.g., electrical signal or glucose level, while at the same time allowing for

stochasticity that arises from randomness of the timing of events. Therefore, developing effective methodologies to provide safety assurance in this setting by means of *quantitative verification* is an important challenge.

Regarding cardiac pacemakers, a number of model-based frameworks have been proposed, to mention the Virtual Heart Model (VHM) of Jiang *et al* [4, 5]. Though mainly intended for simulation and testing, its timed automata pacemaker model [6] has been verified using UPPAAL [9] against a *random heart model*. The random heart model can capture the timing delays between events, but is unable to model the stochasticity in the timing that is characteristic in a heart rhythm and varies from patient to patient. Following a suggestion in [6] that physiologically-relevant heart models are needed to establish the correctness of more complex properties for pacemakers, we earlier proposed a realistic heart model that addresses this issue [1]. The model was adapted from a sophisticated model that generates multi-channel electrocardiogram (ECG) based on nonlinear ordinary differential equations (ODEs) due to Clifford *et al* [2]. To transfer to our setting, where we need to consider that the pacemaker is implanted in the heart tissue, we convert external ECG signals into action potential signals read by implanted sensors. A unique feature of the model of [1] is that the heart can probabilistically switch between normal and abnormal beat types, in a manner that can be learnt from patient data. We performed quantitative, probabilistic verification by analysing the composition of the pacemaker model of [6] and the heart model against typical correctness properties such as (i) whether the pacemaker corrects faulty heart beats, maintaining normal heart rhythm of 60-100 beats per minute (BPM), and (ii) that the pacemaker does not induce erroneous heart behaviours (that is, it does not overpace the heart unless necessary). These were implemented based on the probabilistic model checker PRISM [7] and MATLAB.

One of the shortcomings of the heart model in [1] is that it does not capture the electrical conduction system of the heart, and specifically the delays in the action potential signal as it is propagated from cell to cell. In this paper, we propose an accurate, fine-grained, heart model which is a network of cells, and which can therefore model the conduction delays. Moreover, we work directly with the action potentials that the pacemaker can read from a specific cell. Our model is inspired by the VHM system developed in [5], except that we model cells as *hybrid automata*, in the style of [17], and add *stochasticity* not considered in [5, 17].

The model of [5] can reproduce the timing of the action potential signals, but they do not address the voltage change when the signal is propagated through the cells. Our model is more precise, in that we work directly with cell action potential, explicitly representing the voltage of the cardiac cell signal as a hybrid automaton of [17]. As a result, the timing of the *effective refractory period* (ERP), i.e., the time of non-responsiveness for the cell to further stimulus, and the *relative refractory period* (RRP), i.e., the period of time when an altered secondary excitation stimulus is possible, can be naturally expressed.

Contribution. The contribution of the paper is as follows. We propose a physiologically-relevant heart model built as a network of communicating input-output hybrid automata which features stochasticity. The model enables the modelling of both diseased and normal rhythms, and can be adapted to exhibit random delays in the timing of events that are patient-specific. We implement the heart model in Simulink and validate it against the pacemaker models of [5], demonstrating basic safety properties of the pacemaker by means of probabilistic approximate model checking, with encouraging results.

Related work. [6] formulate a timed automata model for a cardiac pacemaker and verify it using UPPAAL against a simple random heart model. Tuan *et al* [12] develop a real-time formal model for a pacemaker and verify it with the PAT model checker. Networks of timed automata are employed to devise the Virtual Heart Model [4, 5] and hybrid automata are used in the model of [17], both analysed through simulation. Macedo *et al* [13] develop and analyse a concurrent and distributed real-time model for pacemakers through a pragmatic incremental approach using VDM and a scenario-based approach. Gomes *et al* [3] present a formal specification of the pacemaker using the Z notation and employ theorem proving, whereas Mert *et al* [15] use Event-B and the ProB tool, to validate their models in different situations. None of the above approaches considers stochastic behaviours and properties. Risk analysis of glucose infusion pumps is performed with physiological models using statistical model checking in [16], but there is no stochasticity in the models.

Organisation. The rest of the paper is organised as follows. Sect. 2 presents the necessary background on the function of human heart, its modelling and a pacemaker model. Sect. 3 introduces the electrical conduction system of the heart. There, it is discussed how single cells are implemented, how the SA node differs from other cells and how the conduction system works. Sect. 4 gives an overview of the pacemaker model and some of its characteristic features. Sect. 5 describes the composition of the heart and the pacemaker and how probabilistic approximate verification on such a model is performed. Sect. 6 presents experimental results for basic safety properties for pacemakers. Finally, Sect. 7 includes conclusion and possible future directions.

2. PRELIMINARIES

In this section we describe the working of the heart, including its electrical system. The main function of the human heart is to maintain blood circulation of the body. This rhythmic, pump-like function is driven by muscle contractions, in particular, the contraction of the atria and ventricles which are triggered by electrical signals.
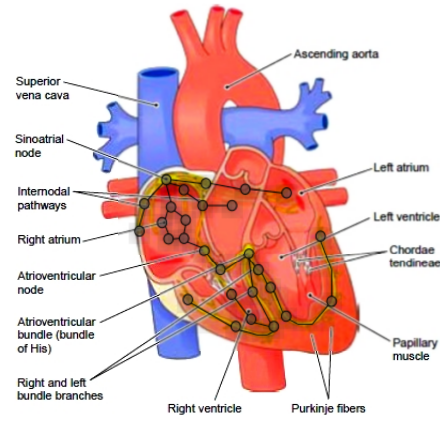
Figure 1: Electrical conduction system of the heart.

The *sinoatrial* (SA) node (a special tissue in the heart, see Fig. 1) spontaneously produces an electrical signal, which is the primary pacemaker of the heart. On each heart beat, it generates the control electrical signal which is conducted through prescribed *internodal pathways* into the atrium causing its contraction. The signal then passes through the slow conducting *atrioventricular (AV) node*, allowing the blood to empty out the atria and fill the ventricles. The fast conducting *Purkinje* system spreads the electricity through the ventricles, causing all tissues in both ventricles to contract simultaneously and to force blood out of the heart. This electrical system is called the *natural pacemaker* (in contrast to the artificial pacemaker) of the heart. At the cellular level, the electrical signal is a change in the potential across the cell membrane, which is caused by the flow of ions between the inside and outside of the cell.

Abnormalities in the electrical signal generation and propagation can cause different types of arrhythmias, such as *Tachycardia* (fast heart beat) and *Bradycardia* (slow heart beat), which require medical intervention in the form of medication, surgery or implantable pacemakers.

Action Potential. At the cellular level, the heart tissue is activated by an external voltage applied to the cell. After the activation, a transmembrane voltage change over time can be sensed due to ion channel activities, which is referred to as an *action potential* (AP). The AP is fired as an all-or-nothing response to a supra-threshold electrical signal, and each AP follows roughly the same sequence of events and has the same magnitude regardless of the applied stimulus. This is also the signal that an implantable pacemaker will receive or generate (see Sect. 2 of [1] for more detail).

3. ELECTRICAL CONDUCTION SYSTEM

In this section, we propose a model for the electrical conduction system of the heart which is tailored for the verification of pacemakers.
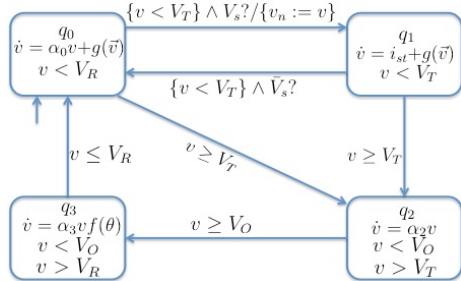
3.1 The Cardiac Cell

In this paper, we adapt the so called Luo-Rudy Guinea Pig Ventricular Cell model (LRd), which is presented in [17] as a hybrid automaton and is depicted in Fig. 2. In a nutshell, there are four (discrete) locations and each of them is associated with an AP phase: *resting and final repolarisation* (q_0),

stimulated (q_1), *upstroke* (q_2), and *plateau and early repolarisation* (q_3). The variables introduced in the model are: the membrane voltage v, which controls switches among locations; a memory variable v_n which is used to modify next ERP phases upon new rounds of excitation; and the excitation current i_{st}.

The memory variable v_n captures the proper response of AP to pacing frequency, which is an essential feature of cardiac excitation. Following [17] we define $\theta = \frac{v_n}{V_R}$ and incorporate the function $f(\theta) = 1 + 13\sqrt[6]{\theta}$ into location q_3. The function $g(\vec{v})$ denotes the voltage contribution from the neighbouring cells. Assuming the total number of cells connected to the current cell k is $N - 1$, we define $g_k(\vec{v})$, the function for the k'th cell to be $g_k(\vec{v}) = \sum_{i=1,i\neq k}^{N} v_i(t-\delta_{ki}) \cdot a_{ki} - v_k \cdot d_k$, where a_{ki} is the gain applied to the potential v_i from cell i, δ_{ki} is the time it takes for the potential to reach cell k, and d_k is the distance coefficient. Moreover, the mode invariants of each location are given as linear inequalities which constraint the membrane voltage. They depend on three model-specific constants: threshold voltage V_T, overshoot voltage V_O, and repolarisation voltage V_R.

The cell starts at location q_0 where two different scenarios are possible. If the cell is externally stimulated with the event $\mathsf{V_s}?$, it enters the stimulated mode updating its voltage according to the *stimulus current* (i_{st}). When the stimulus is terminated, via event $\overline{\mathsf{V}}_s?$, with a sub-threshold voltage, the cell returns to resting without firing an AP. If the stimulus is supra-threshold, i.e., $v \geq V_T$ holds, the excited cell will generate an AP by progressing to mode *upstroke*. Similarly, without any external stimulus, the cell can move from q_0 to q_2 if the voltage (due to the contribution of the neighbouring cells) is supra-threshold $v \geq V_T$. The recovery course of the cell follows transitions to mode *plateau and ER* and then to *resting and FR*. The jump conditions on the control switches monitor the transmembrane potential v, rather than imposing a rigid timing scheme. This approach allows for AP adaptation (response to various pacing frequencies).

Figure 2: Hybrid automaton for the cardiac cell model.

3.2 The Conduction System

Modelling every single cell of the electrical conduction system is computationally intensive. Thus, we abstract the conduction system as a network of cells in order to achieve a good trade-off between the complexity of the model and the running time of the experiments.

In our experiments each cell is connected to neighbouring cells forming a graph as shown in Fig. 1 (black circles connected with lines). There are 33 cells in our graph. The electrical conduction system of the heart consists of conduction pathways with different *conduction delays*. Cells are connected by pathways. The delays of the pathways depend on the physiology of the tissue considered. Moreover, it is possible to use the pathway delays to model various known tissue diseases.

More specifically, our model consists of the SA node, the AV node and 31 cells that share similar properties. The SA node generates sequences of AP which are propagated through the electrical conduction system of the heart (see Sect. 3.3). The AV node is a special cell with the role of slowing down the signal coming from the atria to the ventricles and 31 cells. The 31 cells are connected together forming the graph structure presented in Fig. 1.

Cells communicate with their neighbours through input and output actions. Output actions are propagated to neighbouring cells. For each output action there is a corresponding input action. The set of input actions is $\{\mathsf{AP}?, \overline{\mathsf{AP}}?, \mathsf{VP}?, \overline{\mathsf{VP}}?, \mathsf{V_s}? \text{ and } \overline{\mathsf{V}}_s?\}$, and the set of output actions is defined accordingly. $\mathsf{AP}?$ and $\overline{\mathsf{AP}}?$ are the start and end actions that the pacemaker generates when it paces the atrium (and similarly for $\mathsf{VP}?$ and $\overline{\mathsf{VP}}?$ in the ventricle). Cells that are not stimulated by the pacemaker are stimulated by the voltage of their neighbours.

3.3 The SA Node

In this section we present a model for the SA node, which is known to control the normal rhythm of the heart. The SA node is the impulse-generating tissue located in the right atrium of the heart.

The heart rate is composed of two main periodic components: *respiratory sinus arrhythmia* (RSA) and *Mayer wave* (MW). The RSA oscillation is located between $0.15 - 0.4$ Hz (HF band), while the MW oscillation is at approximately 0.1 Hz between $0.04 - 0.15$ Hz (LF band). The heart rate is measured by monitoring the electrocardiogram (ECG) of the heart. An ECG is a signal recorded from the surface of the human chest. Typically, an ECG signal describes a cardiac cycle, which has three main waves, P, QRS and T. The P wave denotes the atrial depolarisation. The QRS wave reflects the rapid depolarisation of the right and left ventricles. The T wave denotes the repolarisation of the ventricle. The RR-interval is the time between successive R-peaks of the QRS wave, and is the inverse of this time interval that determines the instantaneous heart rate. Analysis of variations in the instantaneous heart rate time series using the beat-to-beat RR-intervals is known as HRV analysis, which has been shown to provide an assessment of cardiovascular diseases.

The RR time series can be generated by first constructing the power spectrum $S(f)$ as a sum of two Gaussian distributions for the LF band and HF band

$$S(f) = \frac{\sigma_1^2}{\sqrt{2\pi c_1^2}} \exp\left(\frac{(f - f_1)^2}{2\pi c_1^2}\right) + \frac{\sigma_2^2}{\sqrt{2\pi c_2^2}} \exp\left(\frac{(f - f_2)^2}{2\pi c_2^2}\right),$$

with means f_1, f_2, standard deviations c_1, c_2 and power σ_1^2, σ_2^2. Then the spectrum is mapped into the time domain by inverse Fourier transform. More details on the construction of the RR time series are given in [14].

In the sequel, we use the RR time series to create the firing times for the events $\mathsf{V_s}?$ and $\overline{\mathsf{V}}_s?$ corresponding to the SA node cell (see Fig. 2). We define RR time series as a
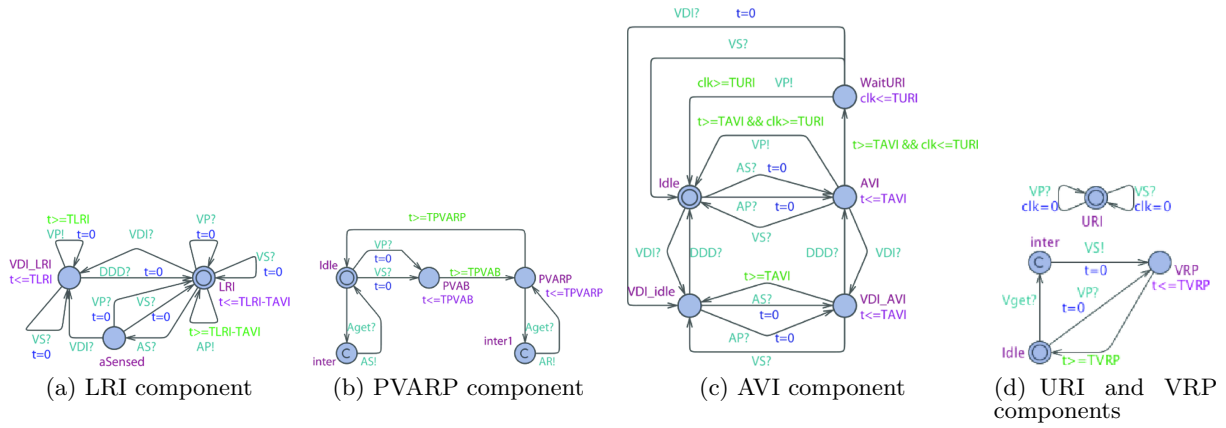
Figure 3: LRI, PVARP, AVI, URI and VRP components [6].

sequence $\{r_i\}_{i \in \mathbb{N}}$. Intuitively, each r_i denotes the period of two consecutive SA node stimulations, i.e., it marks the beginning and end of the stimulus. If the time interval r_i is small the SA node is not stimulated, i.e., the $\overline{V_s}$? happens before $v \geq V_T$. Otherwise, the SA node is stimulated and the stimulus is propagated through the heart.

4. PACEMAKER MODEL

The basic pacemaker model in [6] consists of five timed automata (TA) components: the lower rate interval (LRI) component, the atrio-ventricular interval (AVI) component, the upper rate interval (URI) component, the post ventricular atrial refractory period (PVARP) component and the ventricular refractory period (VRP) component. The LRI component (see Fig. 3(a)) has the function to keep the heart rate above a given minimum value. The AVI component (see Fig. 3(c)) has the purpose to maintain the synchronisation between the atrial and the ventricular events. An event is when the pacemaker senses or generates an action. The AVI component also defines the longest interval between an atrial event and a ventricular event. The PVARP component (see Fig. 3(b)) notifies all other components that an atrial event has occurred. Notice that there is no AR signal as we are not using the advanced algorithms given in [6]. The URI component (see Fig. 3(d) top) sets a lower bound on the times between consecutive ventricular events. The VRP component (see Fig. 3(d) bottom) filters noise and early events that may cause undesired behaviour.

There are four actions in the pacemaker model that are used to communicate with the heart model. The input actions Aget and Vget will notify the pacemaker when there is an action potential from the atrium or from the ventricle, respectively. The output actions AP and VP are responsible for pacing the atrium and the ventricle, respectively. Notice that in a real pacemaker device the input will be a signal. The pacemaker will have a voltage threshold that will be used to decide whether the signal yields an Aget or a Vget action. It is important to remark that all transitions from the pacemaker model that are not labelled with an input or output action are assumed to be labelled with the internal action τ. The locations that have transitions labelled with τ, as well as the locations labelled with $\mathbf{C}$, do not allow the time to elapse.

5. QUANTITATIVE VERIFICATION

In this section, we show how quantitative, probabilistic verification can be performed for a given heart model and the pacemaker model which exhibit real-time, hybrid and stochastic features.

5.1 Generation of Abnormal Heart Behaviour

The heart exhibits abnormal behaviours due to many different reasons and in this paper we consider the malfunction of the SA node. Modelling this aspect plays a crucial role in the verification of pacemakers, as the pacemaker's function is to correct the heart behaviour for such scenarios. We now describe our approach. For the disease of SA nodes, we consider a model based on Markov chains. In order to verify the pacemaker we generate different behaviours, including Bradycardia and Tachycardia, as well as a normal rhythm, and allow switching between them. To generate a behaviour where, for instance, the SA node goes from Bradycardia to Tachycardia, we construct a Markov chain with three states

Figure 4: Markov chain.

(modes) as shown in Fig. 4, labelled as Normal ($\mathbf{N}$), Tachycardia ($\mathbf{T}$) and Bradycardia ($\mathbf{B}$). An important observation is that the probability to switch between states can be learnt from patient data [2].
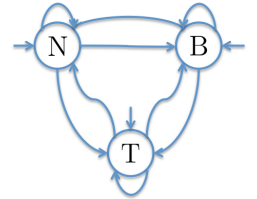
5.2 Property Specification

The composition of the heart and the pacemaker models gives rise to a timed sequence which records the voltage of certain cells of the heart, for which we need to specify a property that checks whether the sequence corresponds to a normal heart behaviour. Intuitively, we define as a "normal path" any path for which *there are between 60 and 100 heart beats (ventricular events) in any interval of window time of 60 seconds.*

5.3 Approximate Model Checking

Technically speaking, the composition of the heart model introduced in Sect. 3 and the pacemaker model yields a stochastic hybrid system. Automated verification of such

systems presents significant research challenges. We exploit techniques from *approximate model checking* (AMC) to deliver verification results with high confidence. AMC is an approach to verify quantitative properties of stochastic systems by simulating the system for finitely many runs, and then analysing the drawn samples (simulation trajectories) to obtain statistical evidence for the satisfaction or violation of the specification. Note that it is closely related to the *statistical model checking* technique [11], which relies on hypothesis testing.

Let T be the time bound, h be the simulation step, and $k = \frac{T}{h}$ be the path length. To apply AMC, one selects two parameters: $0 < \varepsilon < 1$ and $0 < \delta < 1$. Intuitively, ε is the error bound while $1 - \delta$ is the confidence level. We randomly draw $N = \log(\frac{2}{\delta})/2\varepsilon^2$ paths of length k. For each path, we check whether it is "normal". Suppose there are N' normal paths. By the Chernoff bound, one can conclude that the probability $Prob\left[\left|\frac{N'}{N} - p\right| \le \varepsilon\right] \ge 1 - \delta$, where p is what we want to estimate, i.e., the probability of all "normal paths" of length k. Intuitively this means that, with a very high probability (i.e., $1 - \delta$), the ratio $\frac{N'}{N}$ that we compute is ε-close to the real probability p. We refer the readers to [10] for further details.

6. EXPERIMENTAL RESULTS

We implement both the heart model and the pacemaker model in Simulink. We run the experiments on a 2.83GHz 4 Core(TM)2 Quad CPU with 3.7GB of memory. All model files can be accessed via `http://www.veriware.org`.

Fig. 5(a) shows the Simulink implementation of the SA node. The cell is implemented by means of three Simulink blocks: *Event generator*, *Hybrid set* and *Subsystem*. The *Event generator* block is responsible to generate the input events to the cell. The *Hybrid set* implements the cell hybrid automaton model described in Sec. 3.1. The *Subsystem* block performs the integration procedure to compute the voltage level of the cell. Fig. 5(b) shows a network of six cells. Each cell block is composed from the three sub-blocks shown in Fig. 5(a) and connected to other cells through delay and gain components.

Basic Safety Analysis. In the first set of experiments we induce Bradycardia from the SA node and verify that the pacemaker corrects the faulty behaviour by restoring a normal heart beat. In Fig. 6(a) we depict two signals. The first one (in blue) denotes the action potential generated by the SA node which is running in Bradycardia mode. More precisely, we have three beats in six seconds, which is approximately 30 beats per minute. The number of heart beats is thus too slow and needs the intervention of the pacemaker. The second signal (in red) denotes the action potential from one of the cells of the His bundle situated in the ventricle. This is the signal which is captured and paced by the pacemaker. Note that the pacemaker increases the number of beats per minute by first delivering a beat to the ventricle after approximately one second.

Probabilistic Analysis. The second experiment depicts the relation between the probability to generate Bradycardia and the number of pacemaker beats to the ventricle. We range probability from 0.05 to 0.95. The results are presented in Fig. 6(b). We run 40 experiments, each representing 8 minutes of the heart beat. As expected, by increasing
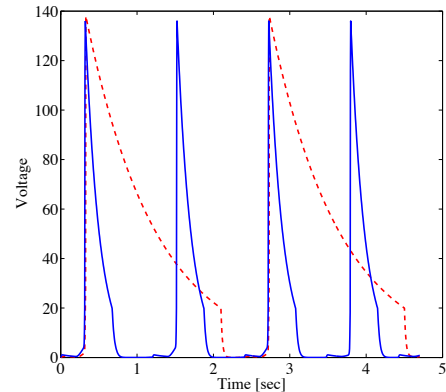
the probability, the pacemaker delivers more beats to the ventricle.
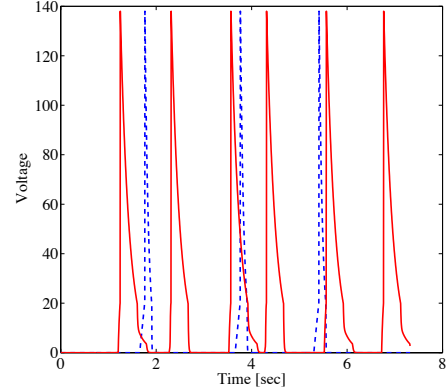
(a) Pacemaker beats
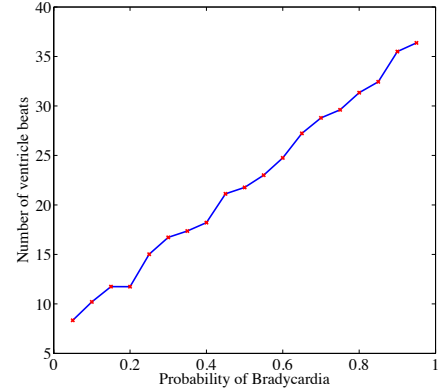
(b) Bradycardia experiment

(c) AV node block

Figure 6: Experiments.

AV node block. In Fig. 6(c) we depict the case when the ERP value of the AV node is long enough, so that it filters out the signal from the SA node. Since a cell cannot be stimulated during its ERP phase, increasing the ERP value of the AV node results in filtering some of the signal that comes from the atrium. In this case, the SA node signal (in blue) is being blocked by a high ERP value of the AV node
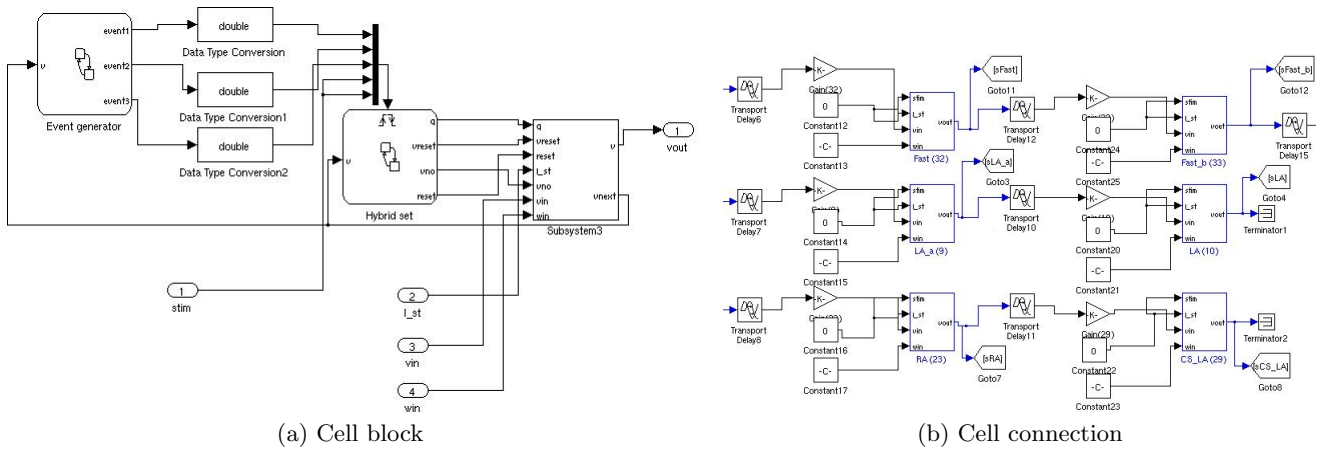
(a) Cell block   (b) Cell connection

Figure 5: Simulink models

(signal in red). The factor is 2 : 1 (two beats in the atrium result in one beat in the ventricle). A long ERP value for the AV node induces Bradycardia in the ventricle.

7. CONCLUSION

In this paper we have proposed a model for the electrical conduction system of the heart and performed quantitative verification of pacemaker models composed with the conduction system model. We worked directly with action potential signals that the pacemaker sensor inputs from specific cardiac cells. We have implemented our heart model in Simulink and evaluated it via approximate model checking using the timed automata pacemaker models of Jiang *et al*, with appropriate extensions.

There are several interesting directions for future work. For instance, we plan to explore the parameter synthesis problem of the pacemakers. Moreover, considering a failure model for the pacemaker seems to be a promising direction.

Acknowledgements

8. REFERENCES

[1] T. Chen, M. Diciolla, M. Kwiatkowska, and A. Mereacre. Quantitative Verification of Implantable Cardiac Pacemakers. *RTSS*, pp. 263-272. IEEE, 2012.

[2] G. Clifford, S. Nemati, and R. Sameni. An Artificial Vector Model for Generating Abnormal Electrocardiographic Rhythms. *Physiological Measurements*, 31(5):595–609, May 2010.

[3] A. Gomes and M. Oliveira. Formal Specification of a Cardiac Pacing System. *FM '09*, pp. 692–707, 2009.

[4] Z. Jiang, M. Pajic, A. Connolly, S. Dixit, and R. Mangharam. Real-Time Heart model for implantable cardiac device validation and verification. *ECRTS*, pp. 239–248. IEEE, 2010.

[5] Z. Jiang, M. Pajic, and R. Mangharam. Cyber-Physical Modeling of Implantable Cardiac Medical Devices. *Proceedings of the IEEE*, 100(1):122–137, 2012.

[6] Z. Jiang, M. Pajic, S. Moarref, R. Alur, and R. Mangharam. Modeling and Verification of a Dual Chamber Implantable Pacemaker. *TACAS*, pp. 188–203, 2012.

[7] M. Kwiatkowska, G. Norman and D. Parker. PRISM 4.0: Verification of Probabilistic Real-time Systems *CAV'11*, pp. 585–591, 2011.

[8] B. Kim, A. Ayoub, O. Sokolsky, I. Lee, P. L. Jones, Y. Zhang, and R. P. Jetley. Safety-assured development of the gpca infusion pump software. *EMSOFT'11*, pp. 155–164. ACM, 2011.

[9] K. G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a Nutshell, 1997.

[10] R. Lassaigne and S. Peyronnet. Probabilistic Verification and Approximation. *Ann. Pure Appl. Logic*, 152(1-3):122–131, 2008.

[11] A. Legay, B. Delahaye, and S. Bensalem. Statistical Model Checking: An Overview. *RV'10*, LNCS 6418, pp. 122–135. Springer, 2010.

[12] A. T. Luu, M. C. Zheng, and Q. T. Tho. Modeling and Verification of Safety Critical Systems: A Case Study on Pacemaker. *SSIRI*, pp. 23–32. IEEE, 2010.

[13] H. Macedo, P. Larsen, and J. Fitzgerald. Incremental development of a distributed real-time model of a cardiac pacing system using vdm. *FM'8*, LNCS 5014, pp. 181–197. 2008.

[14] P. McSharry, G. Clifford, L. Tarassenko, and L. Smith. A Dynamical Model for Generating Synthetic Electrocardiogram Signals. *Biomedical Engineering, IEEE Transactions on*, 50(3):289 –294, 2003.

[15] D. Méry and N. K. Singh. Pacemaker's Functional Behaviors in Event-B. Rapport de recherche, 2009.

[16] S. Sankaranarayanan and G. E. Fainekos. Simulating Insulin Infusion Pump Risks by In-Silico Modeling of the Insulin-Glucose Regulatory System. *CMSB*, pp. 322–341, 2012.

[17] P. Ye, E. Entcheva, R. Grosu, and S. A. Smolka. Efficient Modeling of Excitable Cells Using Hybrid Automata. *CMSB*, 2005.