Foundations and Trends® in Electronic Design Automation
Vol. XX, No. XX (2015) 1–34

DOI: 10.1561/XXXXXXXXXX

now
the essence of knowledge

Physiological Modeling in Medical Device Software Development and Certification

Zhihao Jiang
University of Pennsylvania
zhihaoj@seas.upenn.edu

Rahul Mangharam
University of Pennsylvania
rahulm@seas.upenn.edu

Contents

Abstract

Z. Jiang and R. Mangharam. *Physiological Modeling in Medical Device Software Development and Certification*. Foundations and Trends® in Electronic Design Automation, vol. XX, no. XX, pp. 1–34, 2015.
DOI: 10.1561/XXXXXXXXXX.

1

Introduction

- Why does the safety of the medical devices, the software component in particular, so important?
- What is the difference between software specification and software requirement?
- What is the current practice to ensure software safety?
- What's the problem with the current practice?
- Why do we need closed-loop verification?
- What is the benefit of using model-based design?
- Can we use model-based design for closed-loop verification?
- How much confidence can model-based closed-loop verification provide?
- How to validate a physiological model? Validated in what sense?
- How do we model the physiological environment that the device operates in? What are the challenges?

- What are the current practice for physiological modeling?
- What are the differences if we would like to use the models for closed-loop verification of medical devices?
- What are the different forms of model-based closed-loop verification?
- How to maintain safety guarantees through out the development process?
- What are possible problems that may arise during each steps?
- How can model-based design help during certification?

1.1 Medical Device Software Safety: Current Practice*

- What are the problems of the certification process?

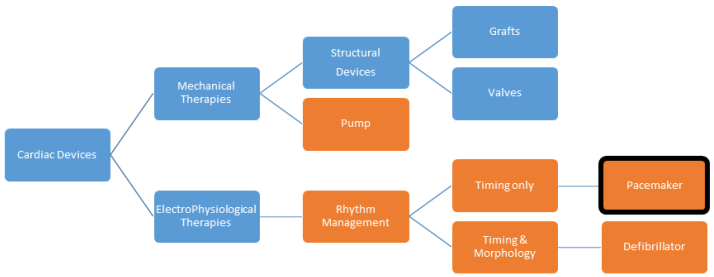


Figure 1.1: Scope of the survey

	Conditions	Treatment/Devices	VHM Coverage
Arrhythmia	Sinus Bradycardia	Pacemaker	Yes
	Heart Blocks (AV, Bundle Branches)	Pacemaker	Yes
	Reentry Tachycardia (Structural & Regular)	Ablation/ICD (anti-tachy pacing)	Yes
	Reentry Tachycardia (Irregular)	Ablation/ICD (shock)	Yes*
Mechanicalfailure	Heart Failure	Artificial Heart (pump)	No
	Valve malfunction	Valve Repair/Replacement	No
	Coronary heart disease	Coronary artery bypass grafting	No

Figure 1.2: List of Cardiac devices

- What are the safety proofs that the FDA receive?
- What is Open-loop Testing? What is the problem with it?
- What is the limitation of the clinical trials?

1.2 Model-based Closed-loop Verification of Medical Device Software

- Can we replace real patient with models? What are the challenges? How much guarantee can we provide?
- In what forms can model-based closed-loop verification performed?
- What are the different requirements for the environment model in these forms?

1.3 Traditional Physiological Modeling**

- Where are the applications of those physiological models?
- What are the key perspectives that the modeling should focus on?
- Can we use these models for model-based closed-loop verification?
- What are the difference between these applications?

1.4 Modeling Perspectives**

- Complexity
- Generality
- Validation

2

Understand The Physiological Environment

- What is the environment that the pacemaker works in?
- How does the environment work?
- How can the pacemaker interact with the environment?
- What are the diseases that the pacemaker can treat?

Medical devices interact with the human body and the physiological requirements are specified according to the pre- and post- conditions of the patient. The knowledge of the dynamics of the human physiology and how it interacts with the devices is essential for 1) constructing physiological models; 2) understanding and encoding physiological requirements; 3) evaluating the closed-loop interactions between the human body and the devices. It is important to understand the physiological environment of the device at the level that 1) unnecessary details unrelated to the interaction between the device and the human are abstracted away, and 2) essential details required to differentiate different patient conditions are maintained.

2.1 Heart, Arrhythmia and Pacemaker

The heart generates periodic electrical impulses to control heart rates according to physiological needs. These impulses conduct through the heart, triggering coordinated muscle contractions and pump blood to the rest of the body. The underlying pattern and timing of these impulses determine the heart's rhythm and are the key to proper heart functions. Derangements in this rhythm are referred to as *arrhythmia*, which impair the heart's ability to pump blood and compromise the patients' health. Arrhythmia are categorized into so-called Tachycardia and Bradycardia. Tachycardia features undesirable fast heart rate which results in inefficient blood pumping. Bradycardia features slow heart rate which results in insufficient blood supply. Bradycardia are due to failure of impulse generation with anomalies in the SA node, or failure of impulse propagation where the conduction from atria to the ventricles is delayed or blocked.

The implantable cardiac pacemaker is a rhythm management device designed to treat bradycardia. A typical dual chamber pacemaker has two leads inserted into the heart through the veins which can measure the local electrical activities of the right atrium and right ventricle, respectively. According to the timing between sensed impulses the pacemaker can deliver electrical pacing to the corresponding chamber to maintain proper heart rhythm.

2.2 Cellular Level ElectroPhysiology

The contraction of heart muscles are triggered by 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 (Fig. 2.1(a)). The upstroke of the action potential is called depolarization, during which the muscle will contract. The voltage change caused by the depolarization will depolarize the cells nearby, which causes an activation wave across the heart. After the depolarization there is a refractory period during which the cell recovers to the pre-excitation state and the voltage drops down to the resting potential. The refractory period can be divided into *Effective Refractory Period (ERP)* and *Relative Refractory Period (RRP)* (Fig. 2.1(b)). During ERP, the cell cannot be depolarized due to the lack of the ion. As the result, the activation wave will be "blocked" at the tissue during

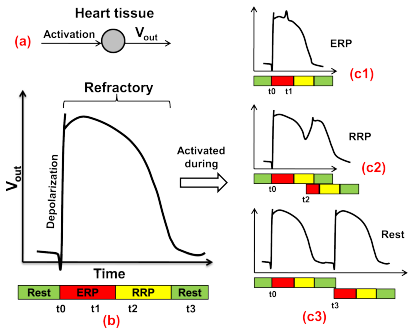
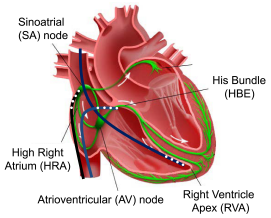


Figure 2.1: (a) The generation of Action potential; (b) Action potential; (c1) The second activation arrived during ERP; (c2) Arrived during RRP; (c3) Arrived after refractory.

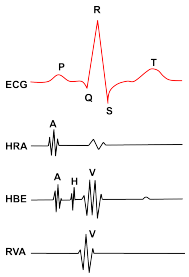
ERP (Fig. 2.1(c1)). During RRP, the cell is partially recovered and the cell can be depolarized. However, the new action potential generated by the depolarization will have different morphology, thus affecting the refractory periods of the tissue and conduction delay of the activation wave (Fig. 2.1(c2)). Fig. 2.1(c1)-(c3) show the action potential shape change and corresponding timing change in refractory periods when the cell is activated at time stamp $t1$, $t2$, $t3$ after the initial activation $t0$.

2.3 Electrical conduction system of the heart

Heart tissue with different timing parameters assemble the electrical conduction system to ensure coordinated contraction of the heart. First, specialized tissue at the Sinoatrial (SA) node periodically and spontaneously self-depolarizes. This is controlled by the nervous system and the SA node is the primary and natural pacemaker of the heart. The activation signal then travels through both atria, causing contraction and pushes blood into the ventricles. Then the activation is delayed at the Atrioventricular (AV) node which allows the ventricles to fill fully. The fast-conducting His-Purkinje system then spreads the activation signal within both the ventricles. The simultaneous contraction of the ventricle muscles will push the blood out of the heart.



(a)



(b)

Figure 2.2: (a) Node automaton. Dotted transition is only valid for pacemaker tissue like SA node; (b) Path automaton; (c) Model of the electrical conduction system of the heart using a network of node & path automata Jiang et al. [2010].

2.4 Electrophysiological Testing

The electrical activities of the heart can be monitored and used to diagnose arrhythmia. The most well-known method is Electrocardiogram (ECG or EKG), which measure the integration of electrical activities of the heart measured along different axis on the body surface. The electrical activities can also be measured by inserting electrodes through the vein into the heart. The electrodes are placed against the inside heart wall and localized electrical activities can be measured. (Fig. 2.2(a)) Physicians can also deliver pacing sequence through the electrodes to explore the heart conditions. This procedure is referred to as Electrophysiological (EP) Testing and the signals are referred to as electrograms (EGMs) (Fig. 3.2).

Implantable pacemakers and ICDs follow the principle of EP testing. For a dual chamber pacemaker, two leads are inserted into the right atrium and right ventricle, respectively. The pacemaker senses the intrinsic generation

and conduction of the electrical signals in the two chambers and deliver electrical pacing when the heart rate and/or atria-to-ventricles conduction interval are abnormal.

♣[1]

[1] HAO: A figure for heart and pacemaker

3

Model The Physiological Environment

- How to encode domain knowledge of the physiological environment into models?
- What are the applications that the models will be used?
- What are the differences in terms of environment models between model checking and simulation?
- How to balance complexity and expressiveness of the model?

Models (especially environment models) should be designed in accordance with their respective applications. During model-based development of medical devices, the environment model can be used for 1) closed-loop testing and 2) closed-loop model checking. Each application has different focus thus has very distinct requirements for the models. These requirements will affect the basic properties of the models, including 1) model complexity, 2) model identifiability etc.

Model Complexity is generally measured in terms of the size of state space and/or computation complexity of state transitions, which affects the computation cost (memory and time) for closed-loop verification. The complexity requirements of an environment model is usually determined by 1)

The complexity of the interactions between the environment model and the system model and 2) The complexity of the environment condition specified in the physiological requirements.

Model Identifiability is a metric of the feasibility/difficulty of identifying model parameters from data. It affects the validity of the model which is a key element for closed-loop verification. Model identifiability is generally related to 1) Model complexity and 2) Data availability/quality. In general, the more complex the model structure is, the harder the model can be fully identified. For physiological modeling, the data availability is almost always an issue.

In the following sections, we demonstrate how to construct heart models for closed-loop verification of implantable pacemaker. Note that for two different applications the models are constructed differently as we address their respective requirements for environment models.

3.1 Heart Models For Closed-loop Testing

- Why the models at this level have to be deterministic? Where can they be used?
- How electrophysiology reflects the functions of the heart?
- How to encode these knowledge into models?
- Why VHM has the right level of details for pacemaker verification?
- How VHM interacts with pacemaker?

During closed-loop testing, the devices interact with the environment (or its models) under different environmental conditions. The closed-loop executions are monitored and violations of safety/efficacy requirements are reported. In model-based closed-loop testing, the environment models are expected to mimic the behaviors of actual environment and its interaction with the device. Thus the environment models are in general deterministic so that the execution traces are reproducible. Complex dynamics during state transitions also need to be captured to validate violations within longer executions traces.

3.1.1 Modeling Philosophy

Interface to the device: The pacemaker can only sense and actuate from two locations within the heart, the spatial fidelity requirement for the heart model is thus low. The diagnosis of heart conditions only relies on the timing of the electrical events, thus the behaviors of the model can be reduced to timing only, if a temporal model of the heart is rich enough to represent majority of heart conditions.
Model expressiveness: Electrophysiology (EP) is an active field in cardiology based on the fact that the mechanical functions of the heart are largely correlated with the electrical activities. During an EP testing procedure, the physicians diagnose heart conditions by examining the patterns and intervals of local electrical activations (temporal) measured from electrodes placed into different locations of the heart (spatial).

Based on the analysis above, an EP-based spatial and temporal model of the heart is capable of representing the electrical behaviors of different heart conditions, and more importantly, their interaction with a pacemaker.

3.1.2 Heart Model Components

We introduce the model components that can be used to configure heart models corresponding to different heart conditions. As introduced in Chapter 2, the action potential of a heart tissue has 3 timing periods during which the tissue responds to external electrical stimuli differently. We use an extended timed-automata formulation (Alur and Dill [1994]) to model the timing behaviors of a heart tissue during each cycle. We refer the tissue model as *Node automaton* and Fig. 3.1(a) shows the structure of a node automaton i. 3 states correspond to 3 timing periods of the action potential. From Rest state, the node can either self activate or activated by external stimuli (Act_node) and go to ERP state. During ERP state the node does not respond to external stimuli (blocked). During RRP state, the node can still be activated and go to ERP state, however the ERP period and the conduction delay of the tissue are affected by the "earliness" of the activation arrived during the RRP period, which is tracked by a shared variable $C(i)$. The new ERP period is determined by a function over clock value $g(f(t))$ which mimic the beat-to-beat dynamics described in Josephson [2008]. The function g and f are given

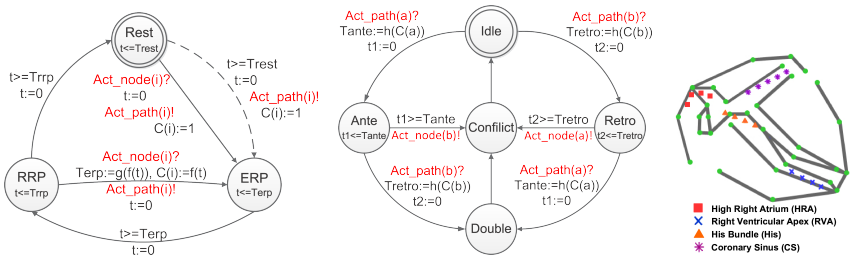


Figure 3.1: (a) Node automaton. Dotted transition is only valid for pacemaker tissue like SA node; (b) Path automaton; (c) Model of the electrical conduction system of the heart using a network of node & path automata Jiang et al. [2010].

by:

$$f(t) = 1 - t/Trrp \tag{3.1}$$

and

$$g(x) = \begin{cases} T_{min} + (1 - (1-x)^3) \cdot (T_{max} - T_{min}), i = AV \\ T_{min} + (1 - x^3) \cdot (T_{max} - T_{min}), i \neq AV \end{cases} \tag{3.2}$$

where T_{min} and T_{max} are the minimum and maximum value for *Terp* of the tissue.

Due to the limited number of observable points within the heart, modeling every tissue of the heart and its full anatomy is unnecessary and unfeasible. In our heart models, only self-activating tissue and key hubs of the electrical conduction system are modeled as node automata. The electrical conduction through the tissue between nodes are abstracted using *path automata*. The path automata can be used to represent structural or topological (functional) electrical connections between nodes. Fig. 3.1(b) shows a path automaton connecting node a and b.

The initial state of a path automaton is *Idle*, which corresponds to no conduction. The states corresponding to the two conduction directions are named after the physiological terms: Antegrade (Ante) and Retrograde (Retro). These states can be intuitively described as forward and backward conductions. If *Act_path* event is received from one of the nodes connected to it, the a transition to *Ante* or *Retro* state correspondingly will occur in the path

automaton. At the same time the clock invariant of the state is modified according to the shared variable *C(a/b)*. This corresponds to the change of the conduction delay that is caused by the early activation. Similar to node automaton, the changing trend is extracted from clinical data and the function h is defined as:

$$h(c) = \begin{cases} path\_len/v \cdot (1 + 3c), i = AV \\ path\_len/v \cdot (1 + 3c^2), i \neq AV \end{cases} \tag{3.3}$$

where $path\_len$ denotes the length of the path and v is the conduction velocity.

After *Tante* or *Tretro* time expires, the path automaton sends out *Act_node(b)* or *Act_node(a)* respectively. A transition to *Conflict* state occurs followed by the transition to *Idle* state. The intermediate state *Conflict* is designed to prevent back-flow, where the path is activated by the node *b* it has just activated. If during *Ante* or *Retro* state another *Act_path* event is received from the other node connected to the path automaton, a transition to *Double* state will occur, corresponding to the two-way conduction. In this case, the activation signals eventually cancel each other and the transition to *Idle* state is taken.

3.1.3 Modeling the Electrical Conduction System

The node and path automata are the basic components for heart modeling. Different hearts under different conditions have different timing parameters and/or different conduction topology. We connect node and path automata with different timing parameters into a network to represent different heart conditions. (Fig. 3.1(c))

3.1.4 Interaction With The Heart Model

In EP testing and pacemaker application, the local electrical activities, measured as electrogram (EGM) signals, are used to diagnose heart conditions. During heart model construction, we can assign a node automaton at electrode locations and the transitions to the ERP state can be used to represent the local activation events. In a more general setup where electrodes are assigned anywhere within the heart model, a probe model is designed to generate synthetic EGM signals using temporal information and spatial information from

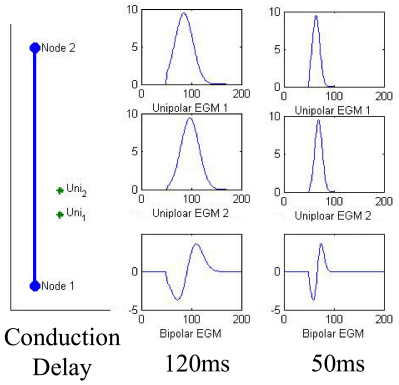


Figure 3.2: The influence of conduction velocity and probe configuration on the EGM morphology. The left columns show the placement of probes in relation to the path; the right columns show the functional EGM.

the network of node and path automata. Fig. 3.2 shows the morphology of EGM signal changes with different conduction velocity and probe configurations. Due to space limitation, detailed description of the probe model can be found in Jiang and Mangharam [2011].

3.2 Non-deterministic Models for Closed-loop Model Checking

- What does nondeterminism do? Where can these model be used?
- How to replace complex dynamics of the deterministic models with nondeterminism?
- What are the abstraction rules that can be applied to the heart models and what are their physiological basis?
- How to encode the information loss during each abstraction steps?

3.2.1 Modeling Philosophy

Model Formalism: Choosing an appropriate formalism for the physiological models is important as the formalism determines the closed-loop state space

and the feasibility to do model checking. Pacemaker utilize the timing of local electrical events to diagnose heart conditions. It is therefore intuitive to use timed-automata models of the heart. Timed-automata is also compatible with model checking tools like UPPAAL (Behrmann et al. [2004]) so that the whole closed-loop state space can be explored.

Model Coverage: Pacemakers are designed to treat bradycardia, however not only should the pacemakers maintain appropriate heart rate when the intrinsic rate is low, but also shall not degenerate other heart conditions. Even for the same patient the condition also changes over time that has to be taken into account. In order to achieve safety across all possible heart conditions, the heart models used during closed-loop verification should be able to cover all possible heart behaviors, more precisely, their mapping to pacemaker inputs. Over-approximation (Clarke et al. [2003]) with non-determinism can be used to simplify model structure while covering larger variety of environmental behaviors. Techniques like model-checking can then be used to examine the whole closed-loop state space for property violations.

Ambiguity Due To Low Sensing Capability: The sensing resolution of pacemakers is low, in terms of the number of sensing location (2 leads), as well as the information obtained from each sensing location (binary events). If abstraction rules utilize the fact that different heart conditions may generate exactly the same input sequence to the pacemaker, there will be ambiguities on concretizing abstract closed-loop executions. For certain conditional requirements, it is important to differentiate all possible concrete executions corresponding to an abstract execution. As the result, the heart model(s) should have the capability to differentiate these heart conditions when verifying certain properties.

Information Lose During Abstraction: While over-approximation achieves simplicity and coverage, it also inevitably introduces invalid behaviors into the model which can cause false-negatives and false-positives during model checking. To solve this problem, more refined models of the heart should be available which can differentiate and eliminate invalid executions when necessary to avoid false-positives.

The most challenging aspect during closed-loop model checking is environment model abstraction and refinement. In Jiang et al. [2014] we developed a series of heart model abstractions at various abstraction levels.

The models are abstracted using abstraction rules derived from physiological knowledge, thus ensuring that each abstraction step covers more physiological conditions. The models in adjacent abstraction levels also satisfy timed-simulation relationship (Yamane [2006]) to ensure complete coverage in the more abstract model. In the rest of the section, we briefly discuss the modeling process and the domain knowledge used. The detailed abstraction and proof for simulation relationship can be found in Jiang et al. [2014].

3.2.2 Initial Abstraction

For the initial heart model, spatially we assume that every heart tissue are modeled. Temporally we model each heart tissue as an automaton shown in Fig. 3.3.b. The beat-to-beat dynamics of heart tissue discussed in the last section is abstracted using non-determinism, as the ERP period and conduction delay of the tissue are non-deterministically chosen from ranges instead of deterministic functions. The model covers all possible timing behaviors of a heart tissue.

3.2.3 Abstract Conduction With Paths

At the first abstraction step, we only model the following kinds of heart tissue with node automata and abstract other heart tissue with path automata:

- Self-depolarizing tissue
- Tissue with long ERP period
- Tissue forming conduction loops

abstract heart tissue

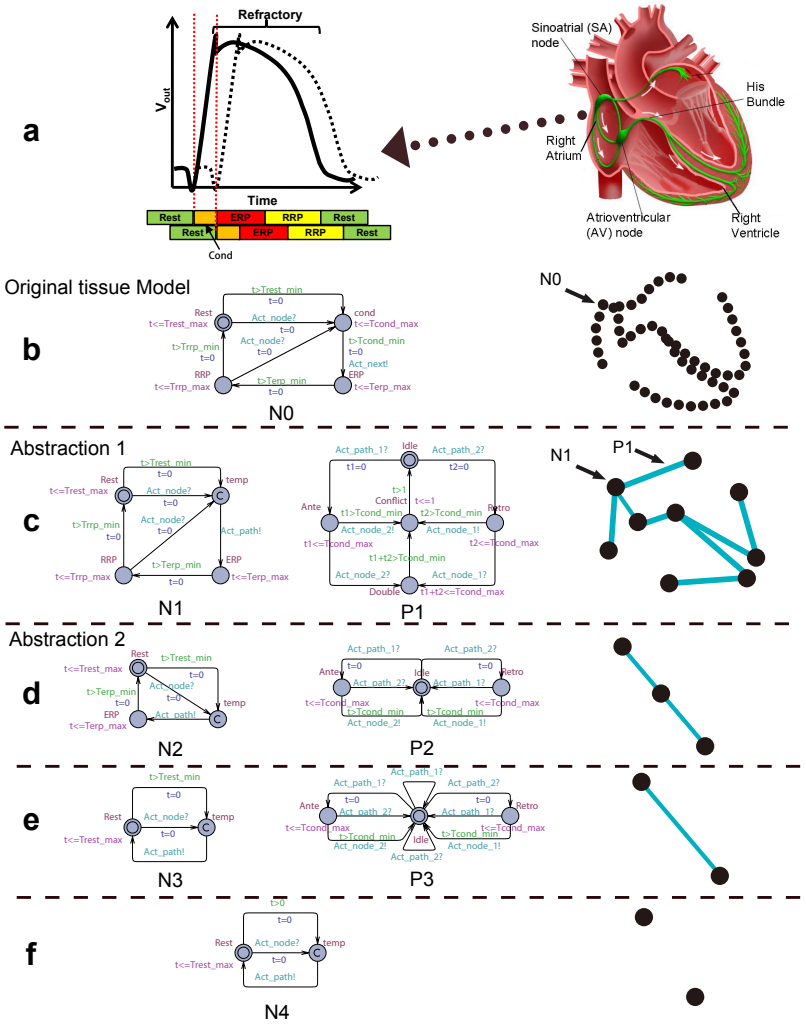


Figure 3.3: (a) The generation of Action potential; (b) Action potential; (c1) The second activation arrived during ERP; (c2) Arrived during RRP; (c3) Arrived after refractory. (Jiang et al. [2014])

4

Environment Model Identification

- Why model identification is important?
- How to identify?
- What's the data availability and quality?
- Are there differences for identifying different kind of models?
- What are the challenges?

Physiological models are developed to represent certain physiological condition in general, or the physiological condition of a specific patient. Therefore the structure of the model and corresponding parameters have to be identified. These information can be obtained from physiological data, which are from: 1) data collected during procedures, and 2) physiological literature in which physiological data are analyzed and summarized. Due to limitation on the interactions with the patient, the availability and quality of physiological data may be bad and there are not enough information to identify all the parameters in the model. It is essential to choose the right level of abstraction so that the model is identifiable (to a large extend). Having physiological correspondence for the model structure and parameters can simplify the iden-

tification process. The rigorousness of the model identification step is also an important factor for the validity of the model.

In the following sections, we briefly discuss our model identification effort for heart models used in two closed-loop verification applications, and their corresponding challenges.

4.1 Heart Model Identification in Closed-loop Simulation

In closed-loop simulation, a heart model should be identified to represent a specific patient under certain heart condition. The constraints for model parameters can be obtained from patient data during *ElectroPhysiological (EP) Testing*. During EP testing, the physicians deliver electrical pacing sequence from electrodes placed inside the patient's heart while extracting timing parameters from observed pattern and timing of electrical events. Since the goal for any EP testing procedure is not to determine all the timing parameters for a patient, the amount of parameters that can be identified from the patient data is limited.

An example: determine ERP and conduction delay for a two node one path heart

4.2 Heart Model Identification in Closed-loop Model Checking

In model checking, the heart models in general have simple structure and less parameters due to non-deterministic abstraction. The heart models are developed in consistent with Electrophysiology terminologies, thus each node and path automata and their timing parameters have physiological correspondence which can be found in literature (Josephson [2008]). The range for non-deterministic parameters directly correspond to the range for possible values of corresponding physiological parameters. Therefore model identification for model checking is much simpler.

An example: ♣[1]

[1] HAO: Need a photo of table of typical parameter values

5

Validate The Environment Model

- What are the different methods to validate physiological models?
- How much confidence can these validation provide?

The validity of the environment models determines the validity of the closed-loop verification results. Since models are always approximations of the actual environment, there is always discrepancies between the model and the actual patient (group). The challenge then is to determine how much safety guarantee that model-based closed-loop verification can provide.

5.1 Validate deterministic models

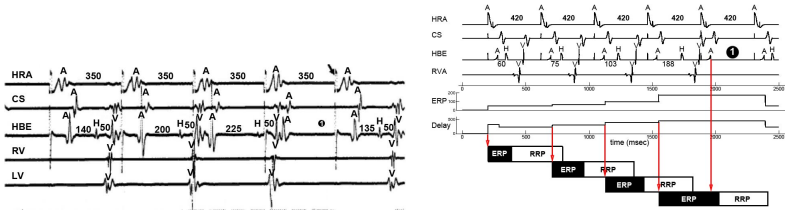
For deterministic models there are two steps

5.1.1 Validating the underlying mechanism

5.1.2 Validating the accuracy of model identification

5.2 Validate by Translating Domain Knowledge**

For non-deterministic models



(a) Testing result for a real patient (Josephson [2008])

(b) Testing result for VHM simulation

Figure 5.1: Key interval values when the coupling interval shortens for (a) a real patient and (b) in VHM simulation Jiang et al. [2010].

- The physiological basis for the initial model come from literature
- All the abstraction rules follow literature
- Models before and after the abstraction have simulation relationships
- Ranges of parameters come from literature.

6

Physiological Requirements

- How requirements are different to specifications?
- What are the forms of the requirements?
- How can the requirements be represented?
- Are those requirements binary?
- Are those requirements equally important?

//

6.1 Physiological Requirements For the heart

- Conditional requirements

6.2 Requirement Representations

6.2.1 TCTL

6.2.2 Simulink Block

6.3 Requirement Hierarchy*

7

A Dual Chamber Pacemaker Specification

- What are the basic functions?

- What happens if new functionalities are applied to the basic model?

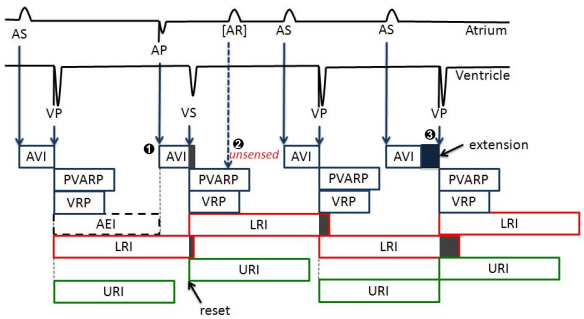
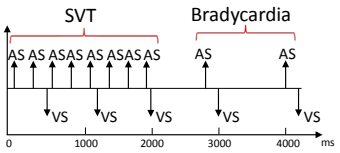
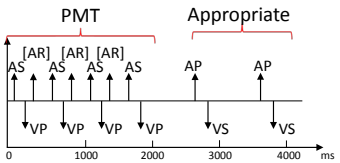


Figure 7.1: Basic 5 timing cycles for a dual chamber pacemaker



(a)



(b)

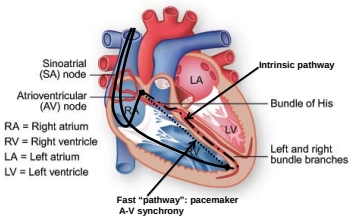
Figure 7.2: (a) Node automaton. Dotted transition is only valid for pacemaker tissue like SA node; (b) Path automaton; (c) Model of the electrical conduction system of the heart using a network of node & path automata Jiang et al. [2010].

7.1 Basic 5 timing cycles

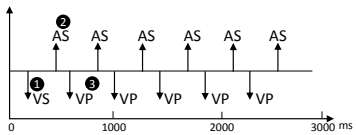
7.2 Atrial Tachycardia Response (Mode Switch)

7.3 Pacemaker Mediated Tachycardia Termination

show need for closed-loop. open-loop inputs of this scenario



(a)



(b)

Figure 7.3: (a) Node automaton. Dotted transition is only valid for pacemaker tissue like SA node; (b) Path automaton; (c) Model of the electrical conduction system of the heart using a network of node & path automata Jiang et al. [2010].

8

Closed-loop Model Checking

- How to use models to cover environmental conditions specified in the physiological requirements, that the device may encounter?

- What are the physiological requirements? What form should they be?

- Can model checking find violations that testing cannot find?

- What are the effects of adding new features? Can they disrupt the safety properties that the previous device hold?

- What is the model complexity requirements for each physiological requirement? When and how to refine the environment model?

- Exploring the whole state space sounds great. What are the limitations of model checking?

8.1 Basic Requirements

8.2 Advanced Functions

8.2.1 Atrial Tachycardia Response

8.2.2 Endless Loop Tachycardia

8.3 Heart Model Refinements

8.4 Quantitative Model Checking

9

Closed-loop Model Simulation/Testing

- What are the limitations of model checking?
- How can simulation complement that?

9.1 Crosstalk

9.2 Lead Displacement

10

Verified Model to Verified Implementation

- How to maintain safety confidence throughout the development process?
- What is the current practice?
- Can it be automated?
- How much confidence can it provide?

11

Certification

- Can model-based closed-loop verification provide more safety guarantee than current practice? How much?

11.1 Current Practice

11.2 Model-based Proof of Confidence

?

Bibliography

R. Alur and D. L. Dill. A Theory of Timed Automata. *Theoretical Computer Science*, 126:183–235, 1994.

Gerd Behrmann, Alexandre David, and Kim G. Larsen. A Tutorial on UPPAAL. *Formal Methods for the Design of Real-Time Systems, Lecture Notes in Computer Science*, pages 200–236, 2004.

Edmund Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith. Counter Example-Guided Abstraction Refinement for Symbolic Model Checking. *J. ACM*, 50(5):752–794, 2003.

Zhihao Jiang and Rahul Mangharam. Modeling Cardiac Pacemaker Malfunctions with the Virtual Heart Model. In *Engineering in Medicine and Biology Society,EMBC, 2011 Annual International Conference of the IEEE*, pages 263 –266, Sept 2011.

Zhihao Jiang, M. Pajic, A. Connolly, S. Dixit, and R. Mangharam. Real-time heart model for implantable cardiac device validation and verification. In *Real-Time Systems (ECRTS), 2010 22nd Euromicro Conference on*, pages 239 –248, July 2010.

Zhihao Jiang, Miroslav Pajic, Rajeev Alur, and Rahul Mangharam. Closed-loop verification of medical devices with model abstraction and refinement. *International Journal on Software Tools for Technology Transfer*, 16(2):191–213, 2014.

M.E. Josephson. *Clinical Cardiac Electrophysiology*. Lippincot Williams and Wilkins, 2008.

Satoshi Yamane. Timed Weak Simulation Verification and its Application to Stepwise Refinement of Real Time Software. *International Journal of Computer Science and Network Security*, 6, 2006.