Requirements-Guided Closed-Loop Testing of Cardiac Devices

Houssam Abbas and Zhihao Jiang and Rahul Mangharam[1]

***Abstract*— This paper proposes a closed-loop testing setup for pacemakers, in which the pacemaker is connected to a Virtual Heart Model (VHM) and both device and heart signals are used to assess correctness of the device's operation. The test inputs are automatically generated by a requirements-guided algorithm that uses the desired heart behavior as a guide for finding test cases that violate it. These can be replayed by the designer to determine whether the pacemaker operated incorrectly. The advantages of closed-loop testing over open-loop testing are illustrated with two experiments involving the VHM and a validated pacemaker model.**

I. Introduction

Medical devices like implantable pacemakers are designed to diagnose and improve undesired physiological conditions. The capability to affect the physiological conditions of the patient makes the safety of the devices an essential consideration. The software component of these devices is getting increasingly complex, inevitably leading to more safety violations. From 1996-2006, the percentage of software-related causes in medical device recalls has grown from 10% to 21% [1]. During the first half of 2010, the US Food and Drug Administration (FDA) issued 23 recalls of defective devices, all of which can cause serious adverse health consequences or death. At least six of the recalls were caused by software defects [8]. In this paper we focus on the testing of cardiac pacemakers and their software, although the proposed methods are more broadly applicable.

To ensure the safety and efficacy of pacemaker software, we first need to ensure that it exhibits the desired input-output relationship, which is captured in the *Software Specifications*. We also have to ensure that the software is capable of improving the physiological conditions as designed, which is captured in the *Physiological Requirements*. The conformance of the device to its specifications is verified using *open-loop testing* in which the device is given input signals and its output signals are compared with desired outputs according to the Specifications. The input signals are generated by an engineer, or from recorded traces of previous devices. However, open-loop testing is not sufficient nor effective in finding safety violations which involve complex and/or lengthy interactions between the device and the heart. That's because it is not possible to think ahead of all possible interaction sequences, nor is it feasible to generate all possible input sequences to the pacemaker. Thus the Physiological Requirements should be verified with closed-loop testing where the device is tested within its physiological context. Today, the Physiological Requirements are mostly verified by *clinical trials* in which the devices are implanted in a select group of patients and monitored over a certain period of time. The limitations of clinical trials are extremely high cost and limited sample size of patient groups.

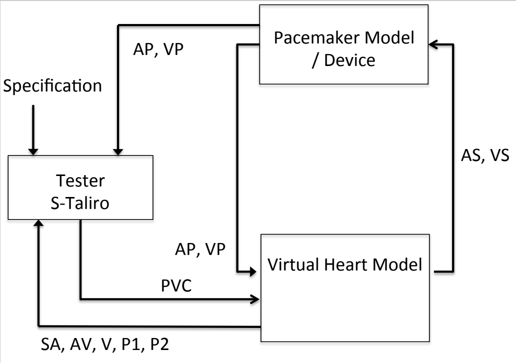

Fig. 1. Requirements-guided closed-loop testing methodology.

Model-based design enables closed-loop verification at an earlier design stage. The device (or corresponding software or even a mathematical model) interacts with a simulation model of the heart. The latter can simulate signals into the device and respond to device outputs.

Contributions. In this paper, we propose the use of a Virtual Heart Model (VHM) for closed-loop testing of pacemakers. The testing setup can be applied to testing actual devices, or software models of them. In addition, we propose the use of an automatic test generation algorithm which uses the Specifications and Requirements as a guide for generating the tests. The tester is guaranteed to find faulty behavior if it exists, under suitable conditions. We demonstrate that using this setup, we can find unsafe or undesirable heart conditions that can not be found in open-loop testing.

II. Method: Requirements-guided closed-loop testing

Figure 1 shows the requirements-guided closed-loop testing setup that we use in this paper to find unsafe or undesirable behavior of the heart connected to the pacemaker. In what follows, we describe each component in details.

A. Virtual heart and pacemaker models

The coordinated contractions of the heart muscles are governed by the generation and conduction of electrical signals throughout the heart. Anomalies in the timing and pattern of the electrical signals results in abnormal heart rhythms, or

*This work was partially supported by the grants NSF CAREER 1253842 and NSF MRI 566112.

[1] The Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, U.S.A. {habbas,zhihaoj,rahulm}@seas.upenn.edu

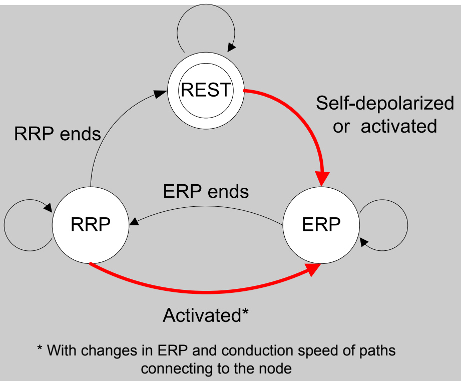
REST
Self-depolarized
or activated
RRP ends
ERP ends
RRP
ERP
Activated*
* With changes in ERP and conduction speed of paths connecting to the node

Fig. 2. FSM modeling the SA node showing Effective and Relative Refractory Periods (ERP and RRP) and the Rest state.

arrhythmia. Implantable pacemakers are designed to treat a subset of arrhythmia by monitoring local electrical activities in the atria and the ventricles and deliver electrical pacing to maintain appropriate heart rhythm.

The electrical activity of the heart is studied extensively in *Electrophysiology* (EP) [7]. The EP principles describe the majority of heart conditions, and are the foundation of all pacemaker operations. In [6] we developed the Virtual Heart Model (VHM) based on the EP principles. The VHM uses Finite State Machines (FSM) to model anatomical and/or topological structures of the electrical conduction system of the heart. The refractory properties of heart tissue are captured using node FSMs and the conduction delays between nodes are captured by the path FSMs. See Fig. 2 for an example of a node FSM. By assembling node and path FSMs with different parameters we are able to simulate different heart conditions, and more importantly, their interaction with any pacemakers. The VHM functional output has been validated by the director of cardiac electrophysiology in the Philadelphia VA Hospital and by electrophysiologists in the Hospital of the University of Pennsylvania. More details are available in [6].

To run the experiments in this paper, we used a DDD pacemaker model developed according to the specification derived from the Boston Scientific Challenge [2]. The model has been validated against the specifications using open-loop testing [5]. Local electrical impulses in the atrium and the ventricle can trigger AS and VS events, respectively. The pacemaker can deliver electrical pacing from the atrial lead (AP) and the ventricular lead (VP). The most basic function of a DDD pacemaker is described as follows: from any ventricular events (VS,VP), if no AS appear within $TLRI - TAVI$, the pacemaker will deliver AP. From any atrial events (AS,AP), if no VS appear within $TAVI$ and the interval from the last ventricular event (VS,VP) is longer than $TURI$, the pacemaker will deliver VP. Together these two functions guarantee that the ventricular rate of the heart maintained above $60000/TLRI$ beats per minute (bpm) and the paced ventricular rate is lower than $60000/TURI$ bpm.

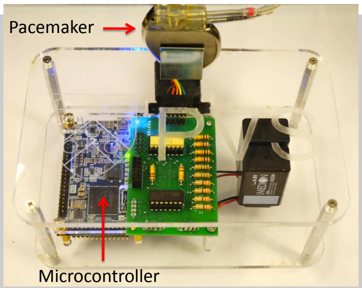

Fig. 3. Heart-on-a-Chip platform, showing the pacemaker and the microcontroller running the VHM code.

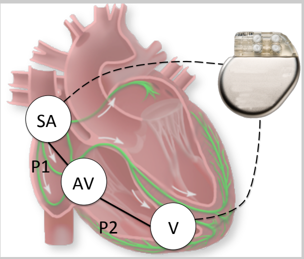

Fig. 4. Portion of the VHM: the conduction paths P1 and P2 connect the SA node to the ventricle V via the AV node (solid line). The pacemaker induces a second, 'virtual', pathway (dashed line).

B. The Heart-on-a-Chip platform

Our closed-loop testing scheme can be performed not only on pacemaker models and code, but also on off-the-shelf pacemaker devices. Fig. 3 shows the Heart-on-a-Chip (HoC) platform for closed-loop testing of pacemakers. The platform consists of a micro-controller running the code of the VHM, and an analog interface to the pacemaker. The analog interface converts VHM outputs to physiological heart signals, which will be input to the pacemaker, and converts pacemaker pacing signals to node activation events. A user interface on the host computer monitors the closed-loop interaction between the heart and the pacemaker and violations to the physiological requirements are reported.

C. Closed-loop testing of pacemaker

Closed-loop setup. The closed-loop setup is shown in Fig. 1. The VHM and pacemaker are connected in the same way as a real heart and pacemaker would be connected. Thus, the inputs to the pacemaker are automatically generated by the VHM as atrial and ventricular sensing events (AS and VS), and don't need to be pre-programmed by the validation engineer as they would be in open-loop testing. The VHM, in turn, is stimulated by the pacemaker's Atrial and Ventricular Pacing events (AP and VP). In addition, the tester is connected at the PVC 'inputs' to the heart. That is, the tester provides waveforms that mimic PVC events as will be explained below. The tester can read all required signals from the VHM. In our experiments, it reads five waveforms: the depolarization events at SA node, AV node, and ventricle (SA, NA and V), and the conduction state of paths P1 and P2 connecting them. (See Fig. 4.) It also reads the AP and VP events from the pacemaker.

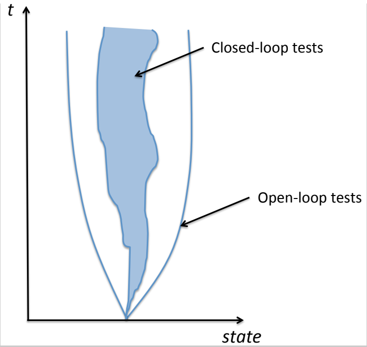

Fig. 5. The space of closed-loop inputs to the pacemaker is constrained by the VHM and is thus smaller than the open-loop space.

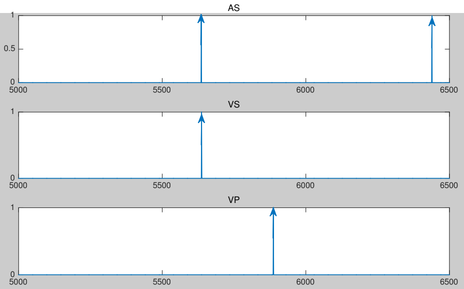

Fig. 6. A PVC pattern that causes a premature ventricular pacing.

Tester-controlled inputs. In our setup, the testing algorithm will generate Premature Ventricular Complex (PVC) waveforms to mimic the abnormal depolarizations of the ventricles that occur in the human heart, and can fool the pacemaker sensing. A *test* is then defined as a PVC waveform of a pre-determined duration T ms. These are used by the tester to try and cause the closed loop to manifest unsafe or undesirable heart conditions. The constraint on the waveforms generated by the tester is that there should be at least 400ms between consecutive PVC impulses. More generally, the tester can also generate Premature Atrial Contraction (PAC) events and vary the closed loop parameters within their physiological range.

The tester. The tester itself is a requirements-guided automatic test generation algorithm, whose theory is given in [3], and we review it here briefly. This testing algorithm has been implemented in the tool S-Taliro [4], and has been used in other medical applications like the analysis of insulin infusion pumps schedules [9]. The operation of the tester is as follows: we provide S-Taliro with a specification that the pacemaker+heart closed loop must satisfy, e.g., “there should be a minimum delay of 500ms between VP events”. S-Taliro iteratively tries to find a test (i.e., a PVC waveform of pre-determined duration) such that the resulting closed loop behavior *violates* the specification. S-Taliro does this by iterating the following process: it generates a test, and observes the output waveform from the closed loop and calculates the *the degree* to which it satisfies the specification or not. This degree is then used to choose the next test to apply. It can be shown that if the system can exhibit a behavior that violates the specification, then this iterative process converges to a test that will provoke this incorrect behavior [3]. The pacemaker’s designer can then replay this test to see where things went wrong, and whether the pacemaker needs to be adjusted accordingly.

Advantages over open-loop testing. The advantages of the proposed requirements-guided closed-loop testing approach over directed open-loop testing are summarized in Table I, and discussed here:

- Choice of inputs: because a heart model provides the input traces to the pacemaker, we know that only relevant test cases will be generated. That is, only traces that the pacemaker may actually see during live operation are fed to it during testing. [1] Compare this to open-loop testing where nothing constrains the inputs, as illustrated in Fig. 5. It is difficult if not impossible to reason a priori about how the heart will constrain the pacemaker’s inputs, especially for *deep* behaviors that take a long time to occur. Moreover, because the tester produces the tests systematically, we are guaranteed to find violating behavior if it exists.
- Criterion of correctness: because we have a VHM, we can express correctness *as a property of the heart’s behavior*, and not of the pacemaker alone. Thus we can evaluate what truly matters: is the heart (as modeled by the VHM) displaying unsafe or undesirable behavior?

Interpretation of testing results. It must be stressed that the interpretation of closed-loop testing results depends on the specific violating behaviors that are found. Some will be determined to be bugs in the pacemaker (or VHM). Others will be determined to be undesirable but unavoidable behaviors, and we show such a case in Section III.

III. Experiments

a) *Exploring behavior*: In this experiment, we tried to falsify the following specification: “The interval between an activation of the ventricle node to a ventricular pacing (VP) should be longer than 500ms”. This specification is designed to identify closed-loop execution traces in which the pacemaker is pacing the heart too fast.

The specification was violated by the execution shown in Fig. 6. Upon investigating the reasons for this violation, it was concluded that one of the noise filters, the post-atrial ventricular blocking (PAVB) designed to avoid crosstalk between channels, caused the problem. A PVC happened shortly after an atrial sense (AS) which fell into the PAVB period and was ignored. Due to its limited sensing capability the pacemaker cannot distinguish noise from a valid input which happened at a rare time. Thus, the designer and physician must decide whether this is an acceptable case, or the pacemaker needs to be adjusted (if at all possible) to prevent this from happening (while maintaining the VSP safety feature).

b) *Finding harmful heart behavior*: In this experiment, we tested the closed loop to see if the pacemaker could lead

[1] Of course, this ultimately depends on the quality of the VHM.

TABLE I
Comparison between open-loop and proposed closed-loop testing.

	Open-Loop	Requirements-Guided Closed-Loop
Choice of pacemaker input traces	Manual and recorded traces. Might miss interesting behavior, or include irrelevant behavior.	Automatic and provided by the VHM, so only relevant input traces are used.
Criterion of correctness	Only pacemaker behavior	The heart and pacemakers's joint behavior, so physiological effects of pacemaker actions can be used to determine correctness.
Choice of tests	Tests = input traces to pacemaker. Manually chosen.	Tests = traces of external disturbances, like PVC and PAC. Automatically selected by S-Taliro

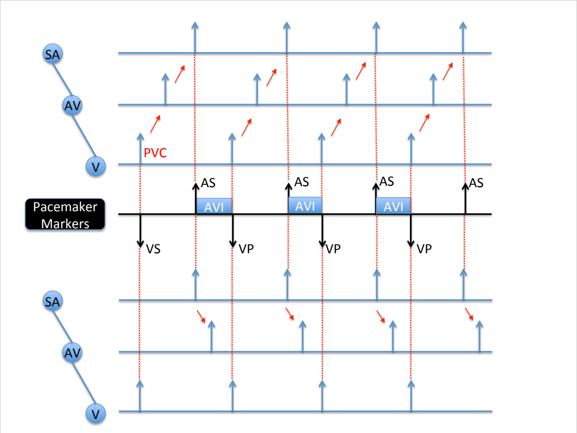

Fig. 7. An ELT sequence (above the Markers line) and an intrinsic conduction (below it) cause the same sensed events in the pacemaker.

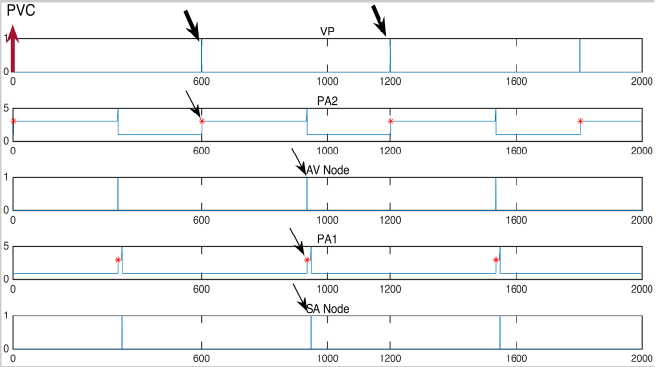

Fig. 8. ELT. The vertical red arrow indicates the initial PVC, thick arrows indicate the beginning of each ELT cycle, and thin arrows indicate the events involved in the ELT diagnosis.

the heart into a harmful condition known as Endless Loop Tachycardia (ELT). The heart has one intrinsic conduction pathway from atria to ventricles, namely from the SA node to the ventricles via the AV node and His bundle. The AVI period of the DDD pacemaker introduces another, virtual, pathway between the atrial lead and the ventricular lead. See Fig. 4. In an ELT, first, a PVC triggers V-A conduction along the intrinsic pathway, which in turn triggers an AS. The pacemaker will then pace the ventricle (issue a VP) after TAVI ms according to its A-V synchrony function. This VP then triggers another V-A conduction, and so on. The conduction loop is then formed and the VP-AS pattern will persist if no actions are taken, and the heart rate is kept as high as the upper rate limit of the pacemaker.As shown in Fig.7, the pacemaker's limited sensing capabilities can not distinguish between a PVC-induced VS and an intrinsic VS. ELT is a harmful condition since a fast fixed heart rate that will cause inefficient pumping of blood. Thus even though the pacemaker is correct according to its specification, it can still lead the heart into ELT if a PVC interferes with its operation as described. S-Taliro was given the ELT specification, and a PVC constraint of at most 2 PVCs in a 10,000ms interval. The total test duration was $T = 10,000$ms. S-Taliro found a PVC pattern, shown in Fig. 8, that caused ELT.

IV. Conclusions

In this paper, we demonstrated a closed-loop testing methodology that uses a specification-guided tester to find unsafe and undesirable heart conditions. The advantages over open-loop were outlined and demonstrated in experiments involving our Virtual Heart Model. In future work, we will apply this testing setup to the Heart-on-a-Chip platform, thus allowing us to test real pacemakers in an automatic manner. It will also allow us to explore the range of parameters for the pacemaker for which safe operation can be guaranteed, and the environmental conditions (e.g., frequency of PVCs and PACs) under which these guarantees hold.

References

[1] List of Device Recalls, U.S. Food and Drug Admin., (last visited Jul. 19, 2010).

[2] PACEMAKER System Specification. Boston Scientific. 2007.

[3] H. Abbas, G. E. Fainekos, S. Sankaranarayanan, F. Ivancic, and A. Gupta. Probabilistic temporal logic falsification of cyber-physical systems. *ACM Transactions on Embedded Computing Systems*, 12(s2), May 2013.

[4] Y. S. R. Annapureddy, C. Liu, G. E. Fainekos, and S. Sankaranarayanan. S-taliro: A tool for temporal logic falsification for hybrid systems. In *Tools and algorithms for the construction and analysis of systems*, volume 6605 of *LNCS*, pages 254–257. Springer, 2011.

[5] G. Chen, Z. Jiang, and R. Mangharam. Model-Based Conformance Testing for Implantable Pacemakers. *SUNFEST Final Report*, 2013.

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

[7] M. Josephson. *Clinical Cardiac Electrophysiology*. Lippincot Williams and Wilkins, 2008.

[8] K. Sandler, L. Ohrstrom, L. Moy, and R. McVay. Killed by Code: Software Transparency in Implantable Medical Devices. *Software Freedom Law Center*, 2010.

[9] S. Sankaranarayanan and G. Fainekos. Simulating insulin infusion pump risks by in-silico modeling of the insulin-glucose regulatory system. In *International Conference on Computational Methods in Systems Biology*, 2012.