On Quantitative Software Quality Assurance Methodologies for Cardiac Pacemakers

Marta Kwiatkowska

Department of Computer Science, University of Oxford
Joint work with Alexandru Mereacre and Nicola Paoletti

Medical CPS, ISOLA 2014, 9th Oct 2014

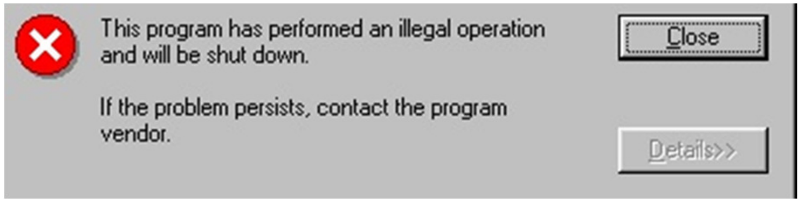
The setting: healthcare applications

- From implantable devices
  - cardiac pacemakers
  - glucose sensors
- through drug delivery
  - closed-loop infusion pumps
- to wearables
  - sports monitoring
  - and of course… Apple iWatch
- Medical CPSs
  - sensors and actuators are integral
  - embedded software controls physical processes
  - increasingly autonomous behaviour
  - combining discrete, continuous and stochastic dynamics

Copyright ©2008 Boston Scientific Corporation All rights reserved.

Are we safe?

- Embedded software at the heart of the device



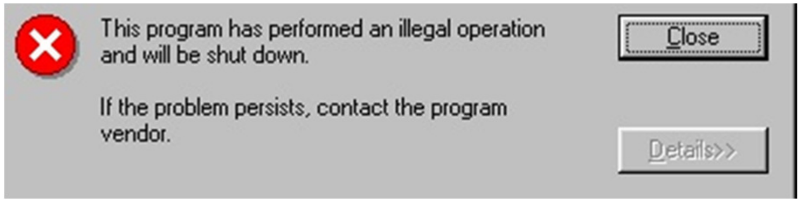
This program has performed an illegal operation and will be shut down.

If the problem persists, contact the program vendor.

- What if…
  - infusion pump software delivers wrong dosage
  - pacemaker software fails

Are we safe?

- Embedded software at the heart of the device



This program has performed an illegal operation and will be shut down.

If the problem persists, contact the program vendor.

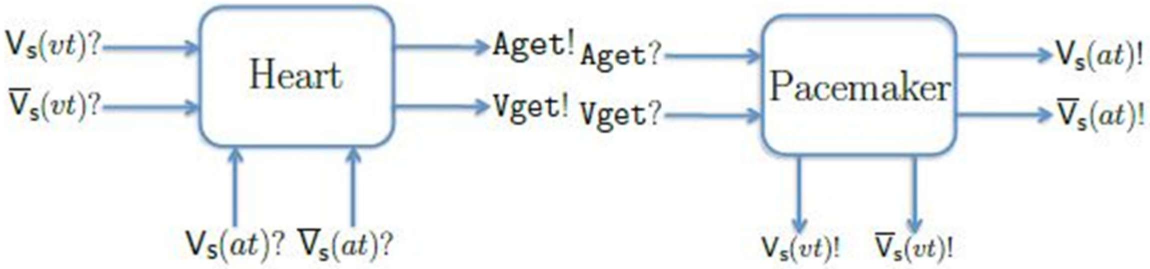
- What if…
  - infusion pump software delivers wrong dosage
  - pacemaker software fails
- Imagined or real?
  - May 2010 and each year since: FDA recalls programmable infusion pumps, over 710 patient deaths in five years, some because the device's software malfunctioned
  - Jan-June 2010 Killed by code: FDA recalls 23 defective pacemaker devices because they can cause adverse health consequences or death, six likely caused by software defects

Software quality assurance

- Software is an integral component
  - performs critical, lifesaving functions and basic daily tasks
  - software failure costly and life endangering
- Need quality assurance methodologies
  - model-based development
  - rigorous software engineering
- Use formal techniques to produce guarantees for:
  - safety, reliability, performance, resource usage, trust, …
  - (safety) “heart rate never drops below 30 BPM”
  - (energy) “energy usage is below 2000 mA per minute”
- Focus on automated, tool-supported methodologies
  - automated quantitative verification and synthesis
  - personalisation of analysis

Software quality assurance for pacemakers

- We formalise and implement a model-based framework
  - models are networks of communicating hybrid I/O automata, realised in Matlab Simulink
    - discrete mode switching and continuous flows: electrical conduction system
    - quantitative: energy usage and battery models
    - patient-specific parameterisation
  - framework supports plug-and-play composition of
    - **heart models** (timed/hybrid automata, some stochasticity)
    - **pacemaker models** (timed automata)



(a) Heart model (b) Pacemaker model

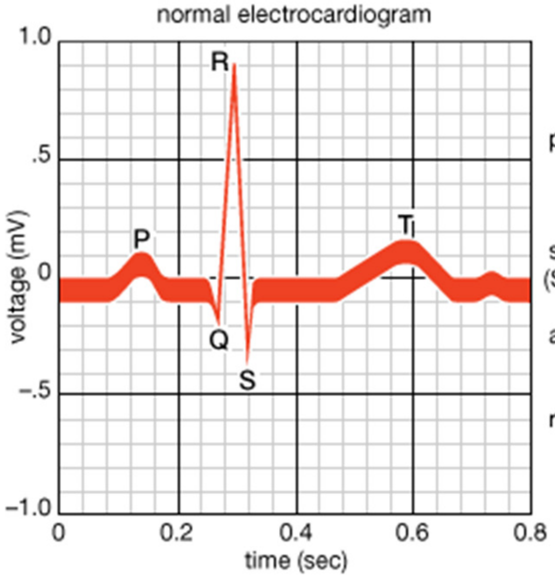
Software quality assurance for pacemakers

- Properties specified in Metric Temporal Logic (MTL)
  - and extensions, needed for some key properties
- Broad range of functionality
  - Monte-Carlo simulation of composed models
    - with (confidence level) guarantees for non-linear flows
  - (approximate) quantitative verification against variants of MTL
    - to ensure property is satisfied
  - parametric analysis
    - for in silico evaluation, to reduce need for testing on patients
  - (new) automated synthesis of optimal timing parameters
    - to determine delays between paces so that energy usage is optimised for a given patient
  - (work in progress) patient-specific parameterisation
  - (work in progress) hardware-in-the-loop simulation
- See http://www.veriware.org/pacemaker.php

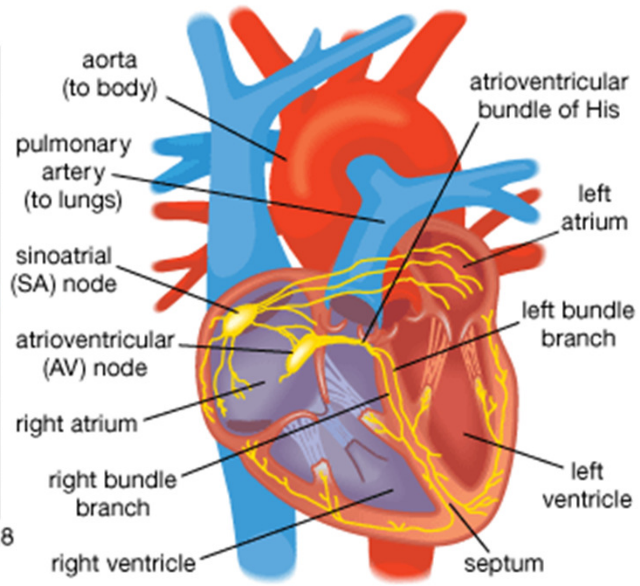
Modelling of the heart

- The heart maintains blood circulation by contracting the atria and ventricles
  - spontaneously generates electrical signal (action potential)
  - conducted through cellular pathways into atrium, causing contraction of atria then ventricles
  - repeats, maintaining 60-100 beats per minute

- Abnormalities in electrical conduction
  - missed/slow heart beat (Bradycardia)
  - treatable with pacemakers

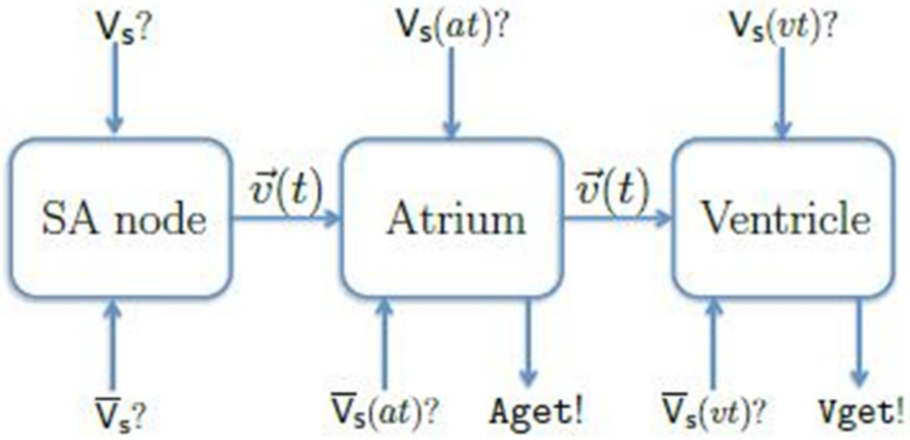


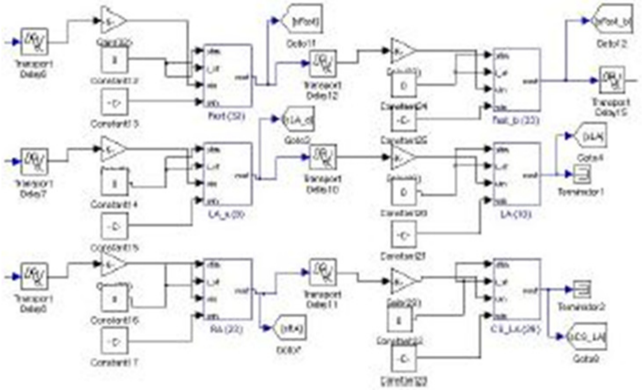
© 2008 Encyclopædia Britannica, Inc.



Cardiac cell heart model

- Use model of electrical conduction [Grosu et al]
  - abstracted as a network of cardiac cells that conduct voltage

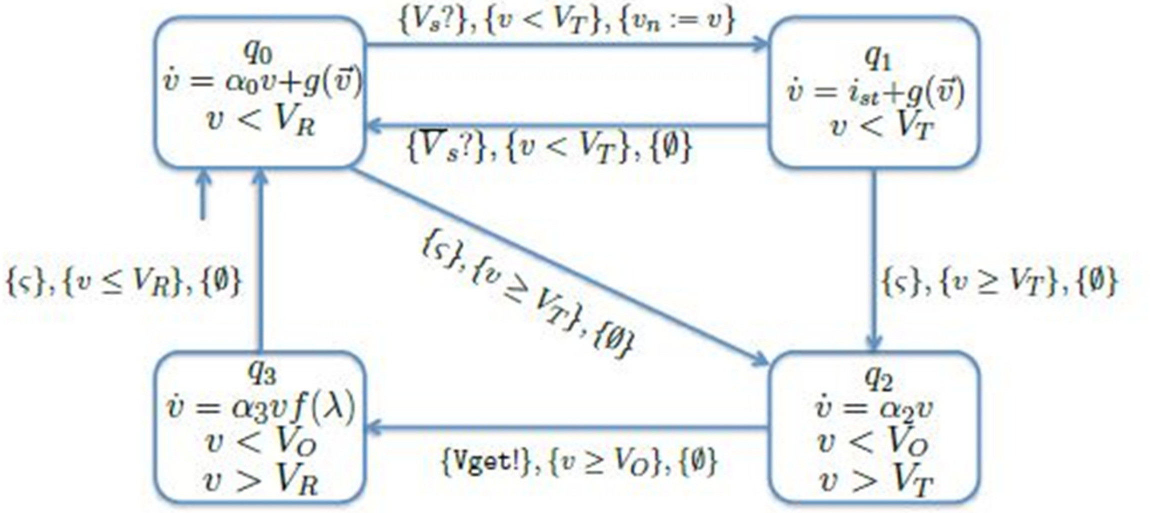




- cells connected by pathways, modelled using Simulink delay and gain components
- SA node is the natural pacemaker

Cardiac cell heart model: single cell

- Single ventricular cell [Grosu et al]
  - four modes: resting and final repolarisation (q_0), stimulated (q_1), upstroke (q_2) and plateau and early repolarisation (q_3)

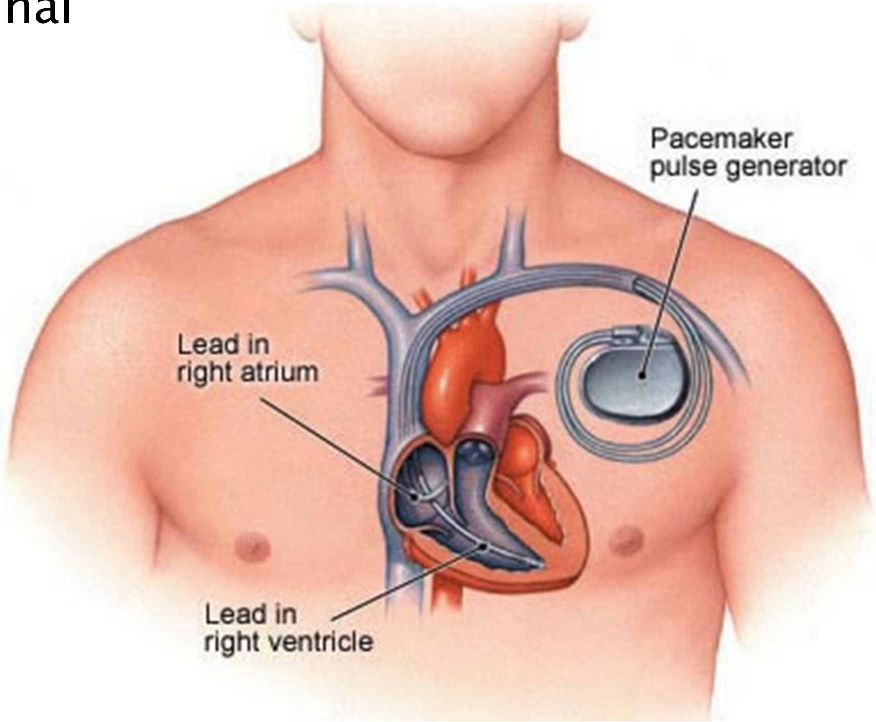


  - variables: v – membrane voltage, i_{st} – stimulus current
  - constants: V_R – repolarisation voltage, V_T – threshold, V_O – overshoot voltage

Implantable pacemaker

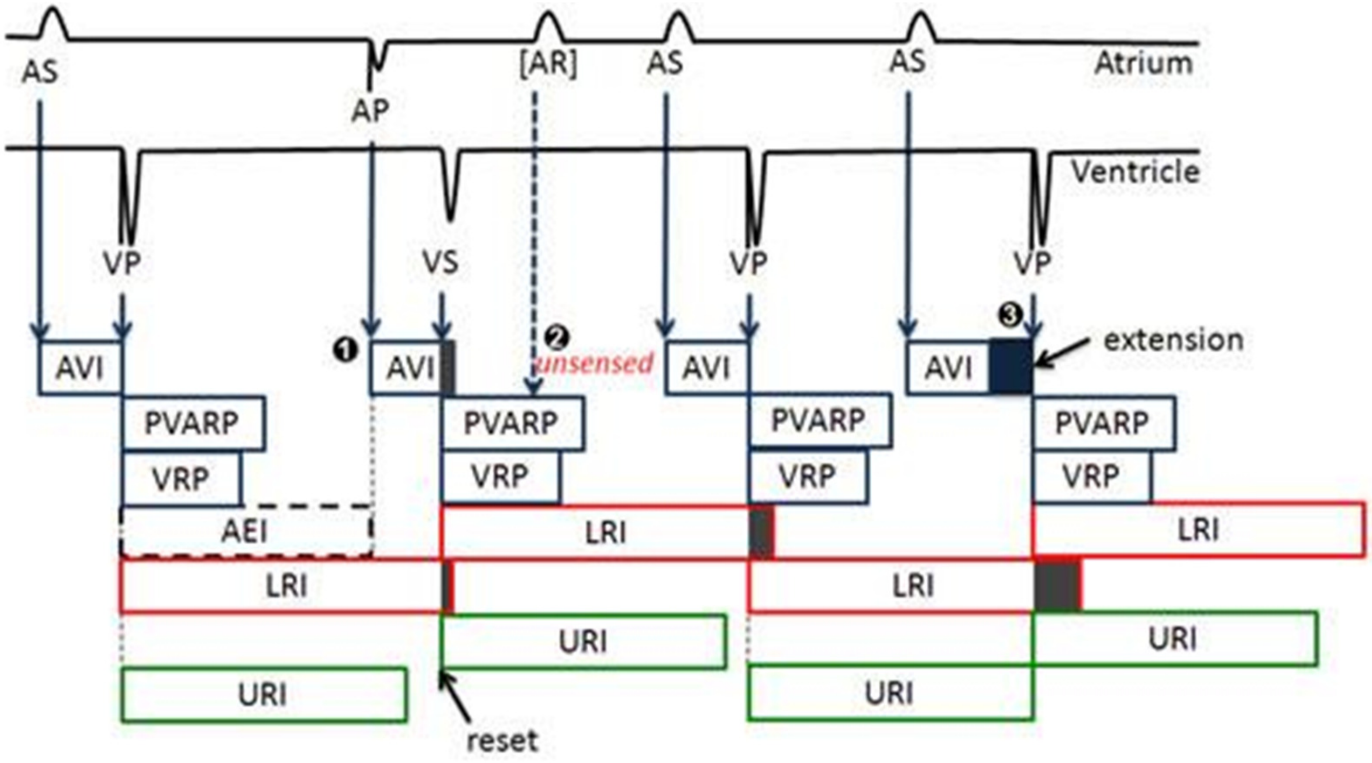
- How it works
  - reads electrical (action potential) signals through sensors placed in the right atrium and right ventricle
  - monitors the timing of heart beats and local electrical activity
  - generates artificial pacing signal as necessary
- Real-time system!

- Core specification by Boston Scientific
- Basic pacemaker can be modelled as a network of timed automata [Ziang et al]



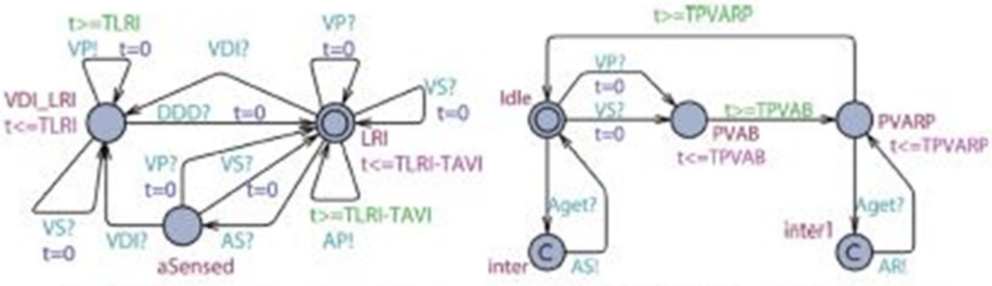
Pacemaker timing cycle

- Atrial and ventricular events



Basic pacemaker

- Consists of five communicating timed I/O automata components [Jiang et al]

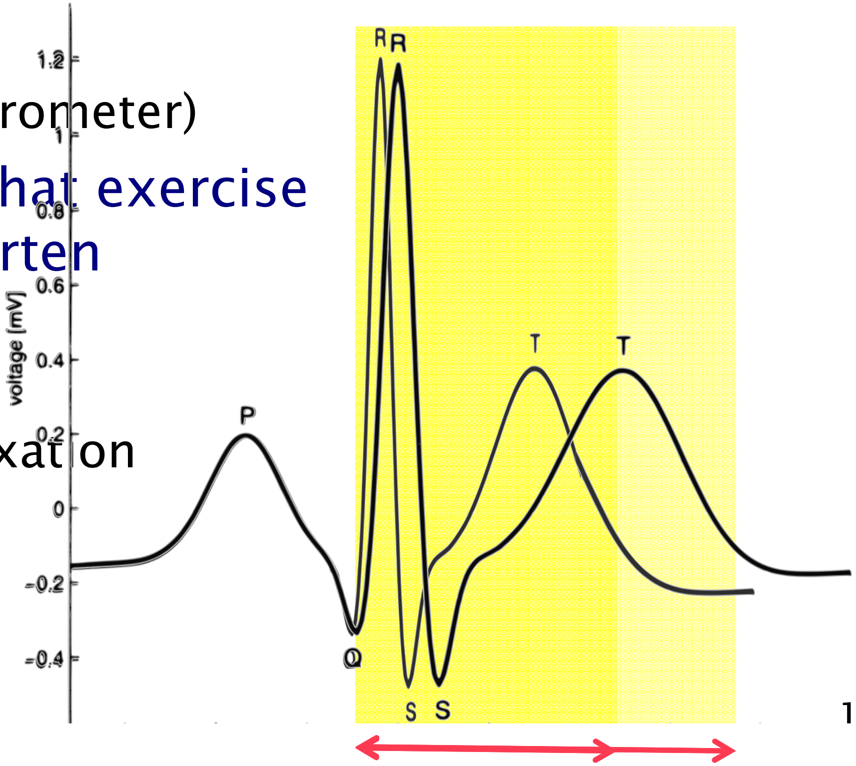


(a) LRI component (b) PVARP component

  - LRI keeps the heart rate above a given minimum value
  - PVARP notifies all other components that an atrial event has occurred
- Can be enhanced with noise and probabilistic switching between healthy and diseased heart (for personalisation)

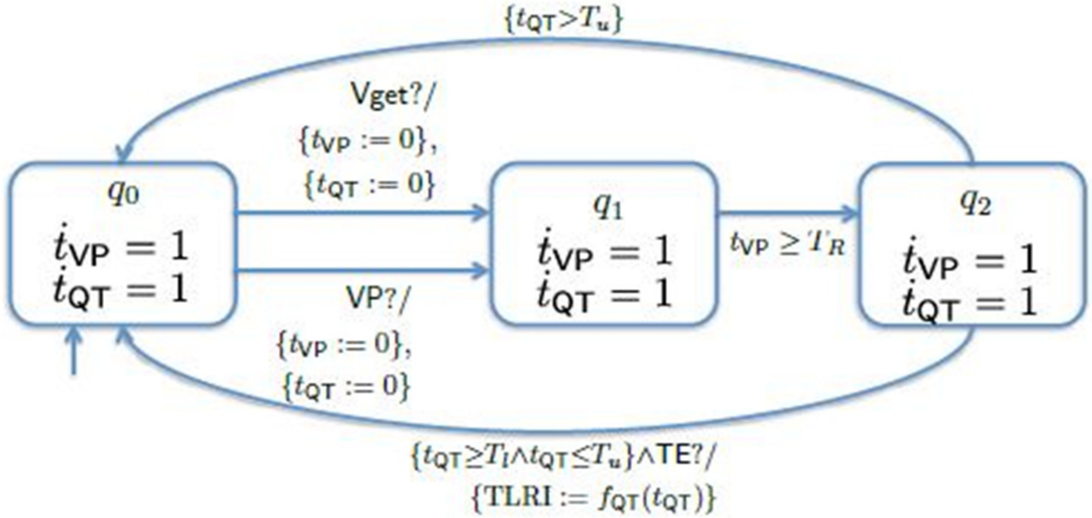
Rate-adaptive pacemakers

- Can regulate the pacing rate according to patient’s needs (exercise, stress, …)
  - needed when the heart cannot adapt its rate to increasing demand (chronotopic incompentence)
- Use implantable sensors to detect activity level and metabolic need
  - e.g. body movement (accelerometer)
- QT sensors exploit the fact that exercise and increased heart rate shorten QT interval (QTI)
  - QT in the ECG is the interval from the contraction to relaxation of ventricles
  - can be measured from ECG (method implemented)



Rate-adaptive component

- We focus on VVIR pacemakers: ventricle sensed and paced, plus rate adaptation with QT sensor

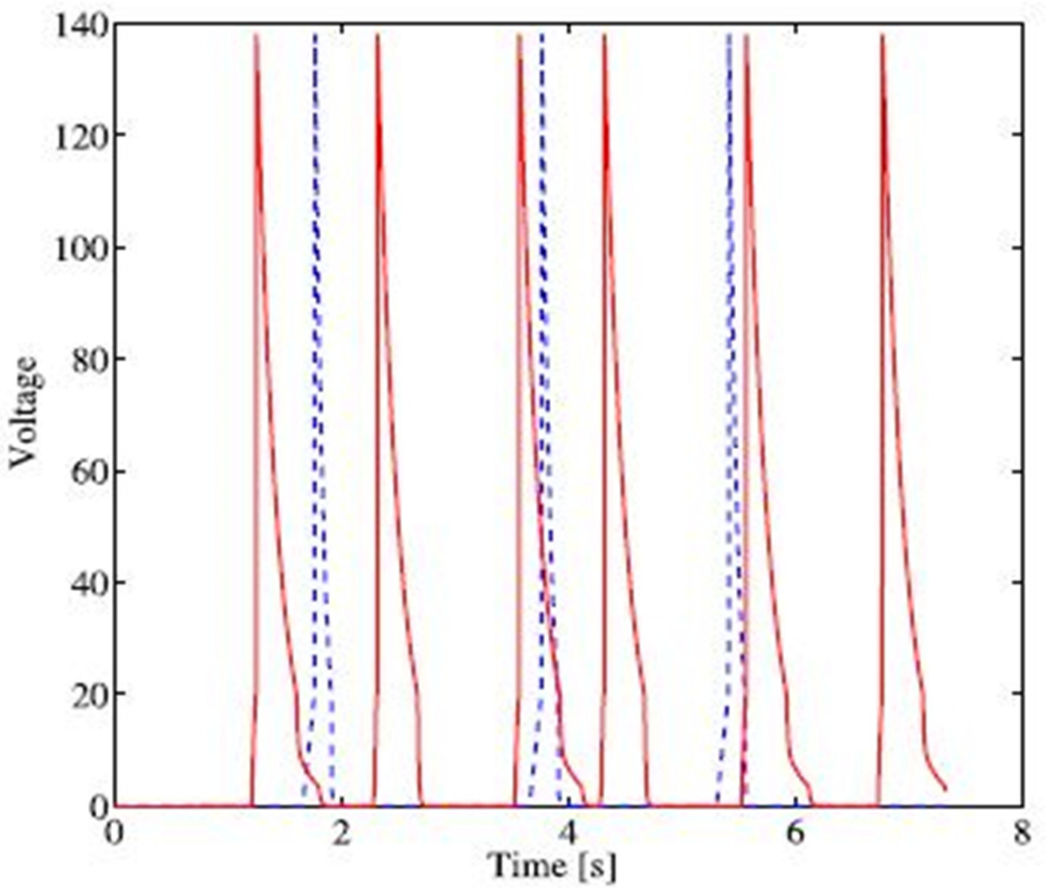


- Add rate-adaptive component to basic specification
  - measures interval between a ventricular event and the offset of the T wave (from the QT detector)
  - updates TLRI by averaging QTIs and applying regression law

Quantitative verification for pacemakers

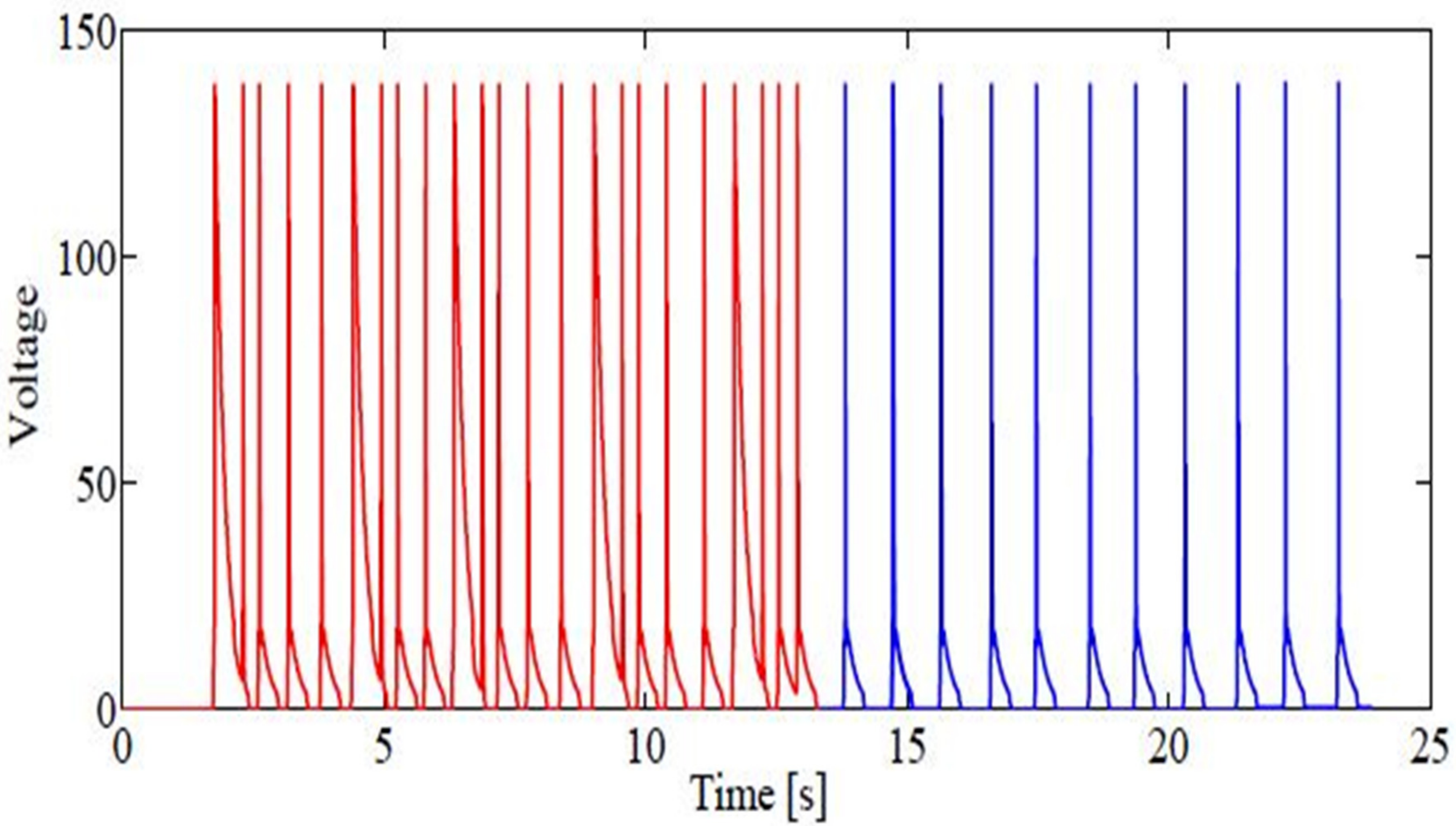
- Given a model of the pacemaker and a heart model
  - e.g. basic pacemaker and cardiac cell model
- compose and verify against MTL properties (syntax omitted):
  - basic safety: “for any 1 minute window, the number of heart beats lies in the interval [60,100]”
  - energy: “for a given time point T, the energy consumed is less than the given energy level V”
- But models are in Simulink, multi-component, hybrid, non-linear, and can contain stochasticity!
- Methodologies
  - rely on simulation and parameterise by simulation step
  - employ approximate verification based on finitely many simulation runs: estimate probability of satisfying property from Chernoff bound, for some confidence interval
  - overapproximate reach sets using annotations and ‘bloating’

Correction of Bradycardia



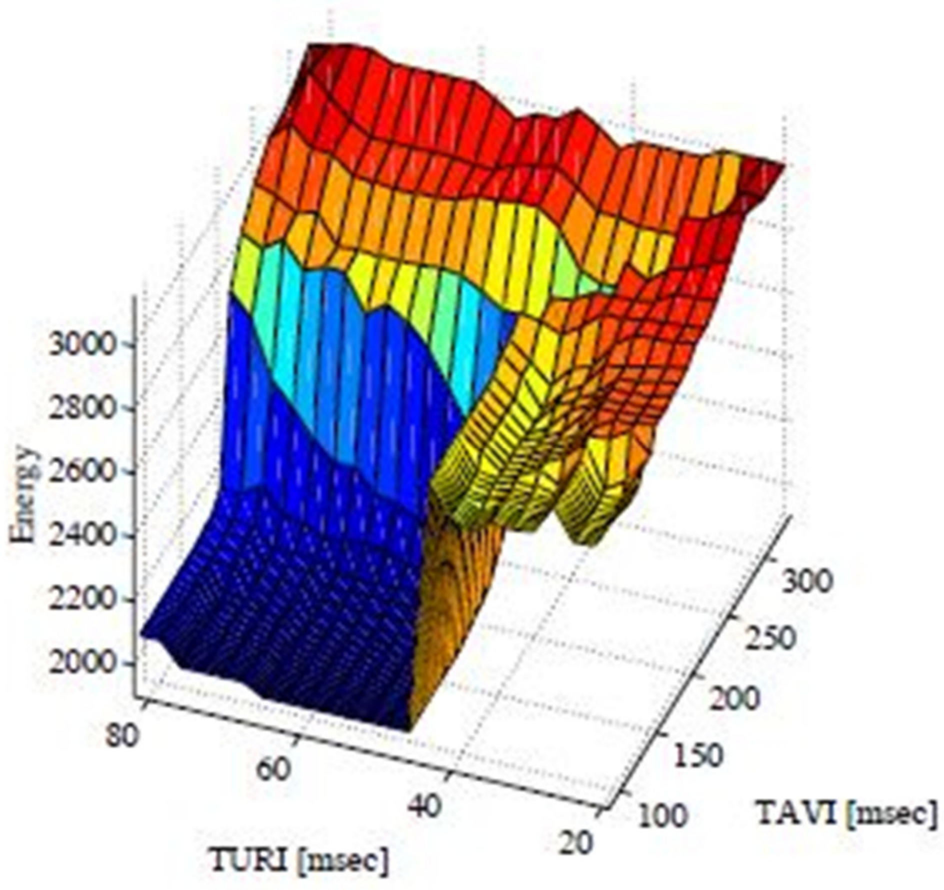
Blue lines original (slow) heart beat, red are induced (correcting)

Correction of PMT



Red lines original (PMT) heart beat, blue are induced (correcting)

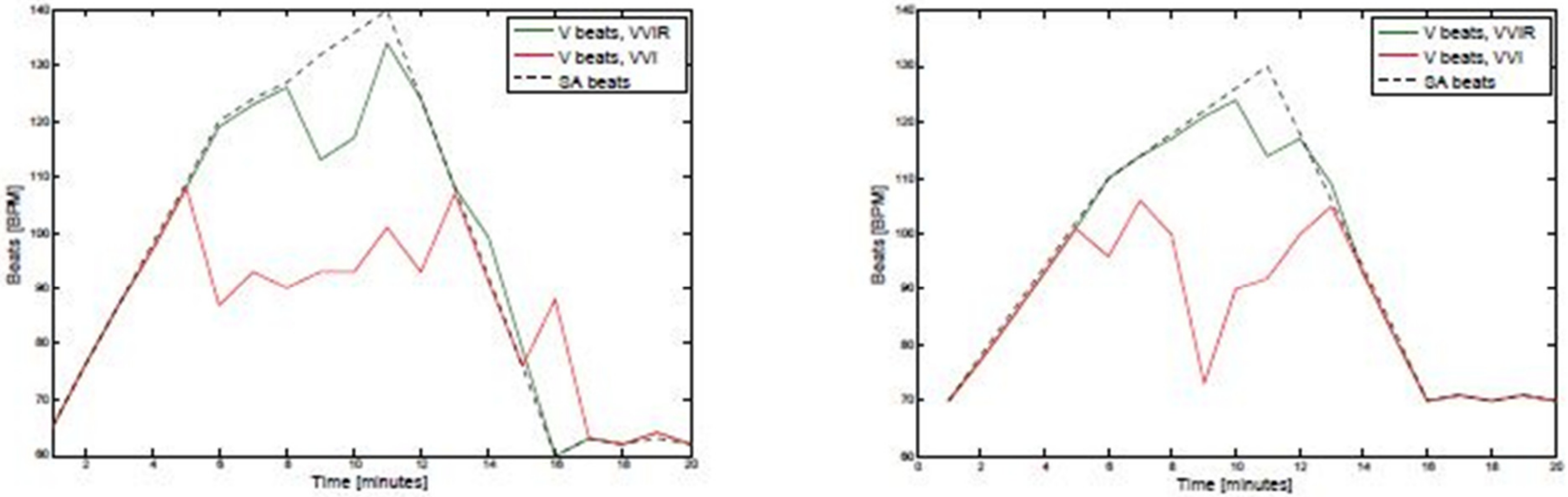
Energy consumption



Battery charge in 1 min under Bradycardia, varying timing parameters.

Quantitative Verification of Implantable Cardiac Pacemakers over Hybrid Heart Models.
Chen *et al*, *Information and Computation*, 2014

Modulation during physical activity

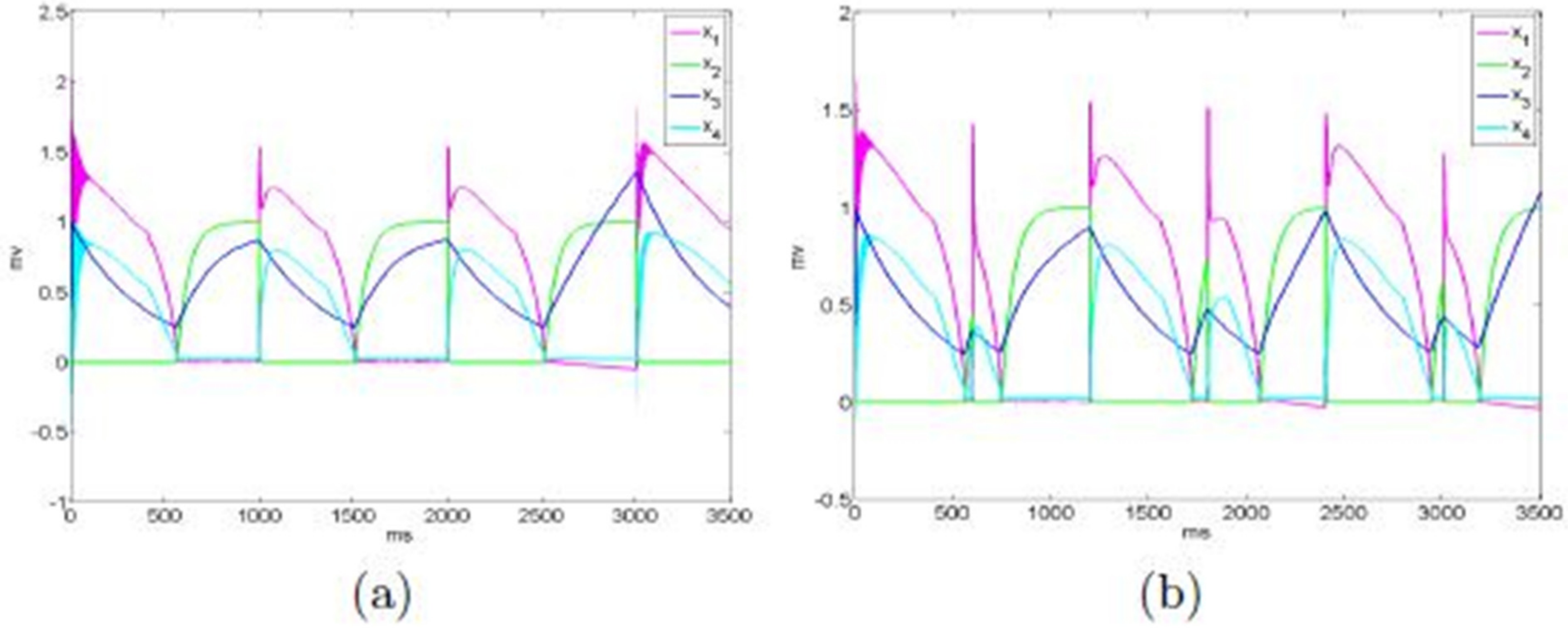


(a) Young patient (b) Old patient

Rate modulation during exercise. Black dashed line indicates metabolic demand, and the green and red curves show rate-adaptive VVIR and fixed-rate VVI pacemakers.

Formal Modelling and Validation of Rate-Adaptive Pacemakers, Kwiatkowska *et al*. In *IEEE International Conference on Healthcare Informatics*, ACM. 2014

Alternans in the heart



(a) (b)

We plot the reach set from a set of initial states with pacing rate of 1000 msec and observe that the AP durations do not change (a), whereas at a pacing rate of 600 msec (b) the AP durations alternate.

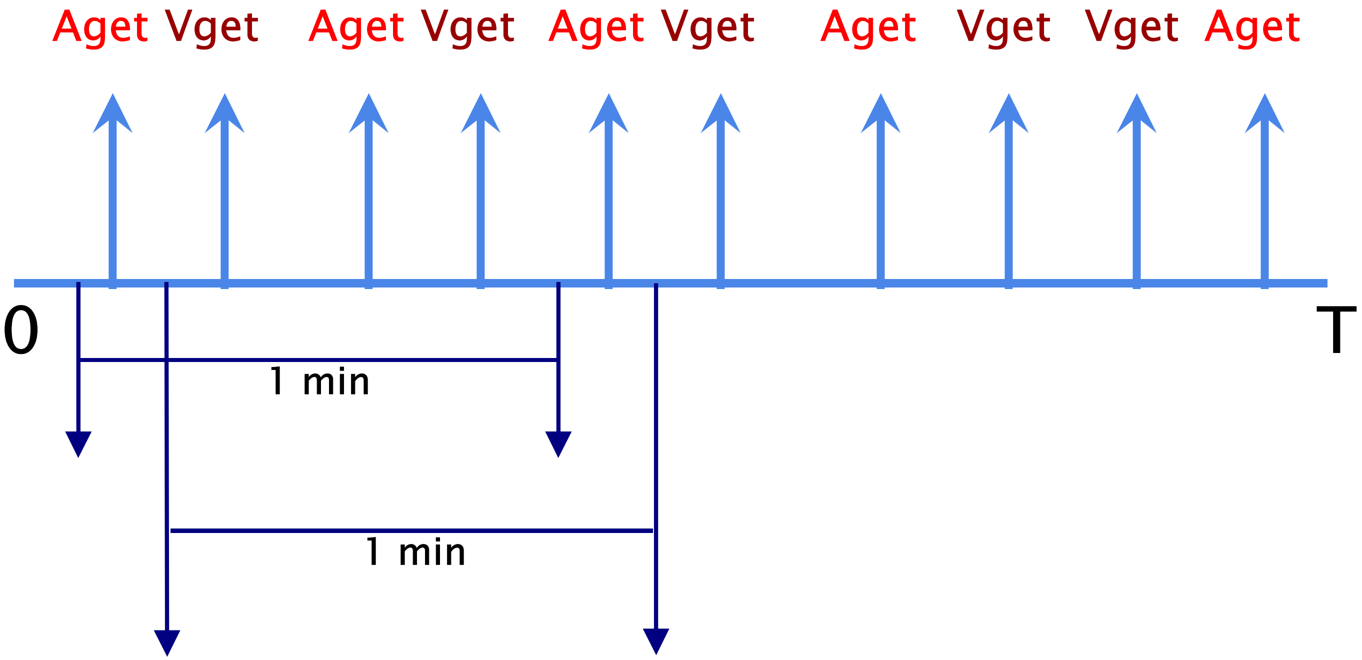
Invariant Verification of Nonlinear Hybrid Automata Networks of Cardiac Cells. Huang *et al*
In *CAV*, volume 8559 of LNCS, pages 373-390, Springer, 2014.

From verification to synthesis…

- Automated verification aims to establish if a property holds for a given model
- Can we find a model so that a property is satisfied?
  - difficult…
- The parameter synthesis problem is
  - given a parametric network of timed I/O automata, set of controllable and uncontrollable parameters, CMTL property ϕ and length of path n
  - find the optimal controllable parameter values, for any uncontrollable parameter values, with respect to an objective function O, such that the property ϕ is satisfied on paths of length n, if such values exist
- Objective function
  - maximise volume, or ensure robustness

Synthesising Optimal Timing Delays for Timed I/O Automata. Diciolla et al. In *14th International Conference on Embedded Software (EMSOFT'14)*, ACM. To appear. 2014

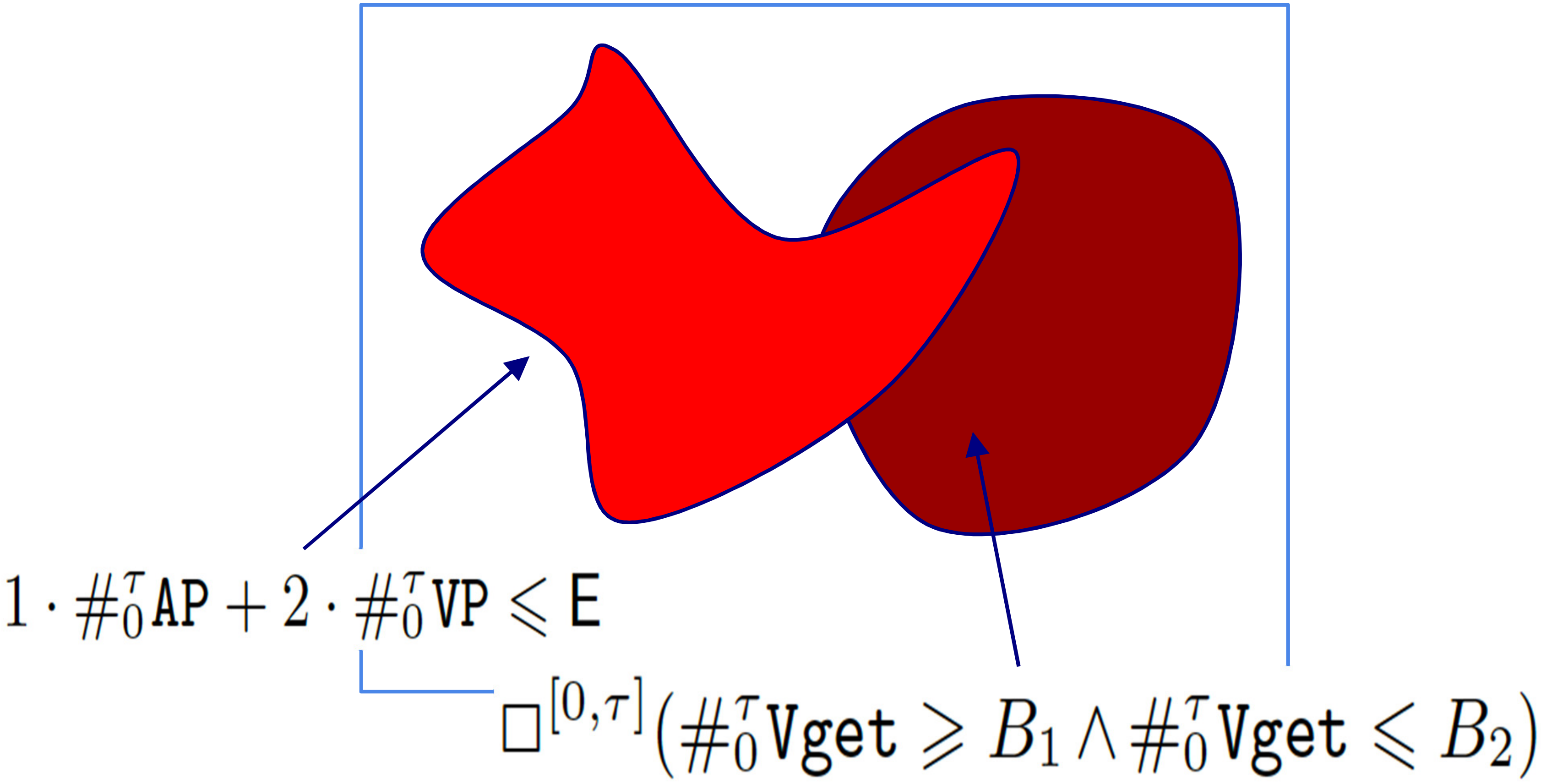
Property patterns: Counting MTL



$$\Box^{[0,\tau]}(\#_0^{\tau}\mathtt{Vget} \geqslant B_1 \wedge \#_0^{\tau}\mathtt{Vget} \leqslant B_2)$$

Event counting, weighted summation (not expressible in MTL).

Safety and energy efficiency

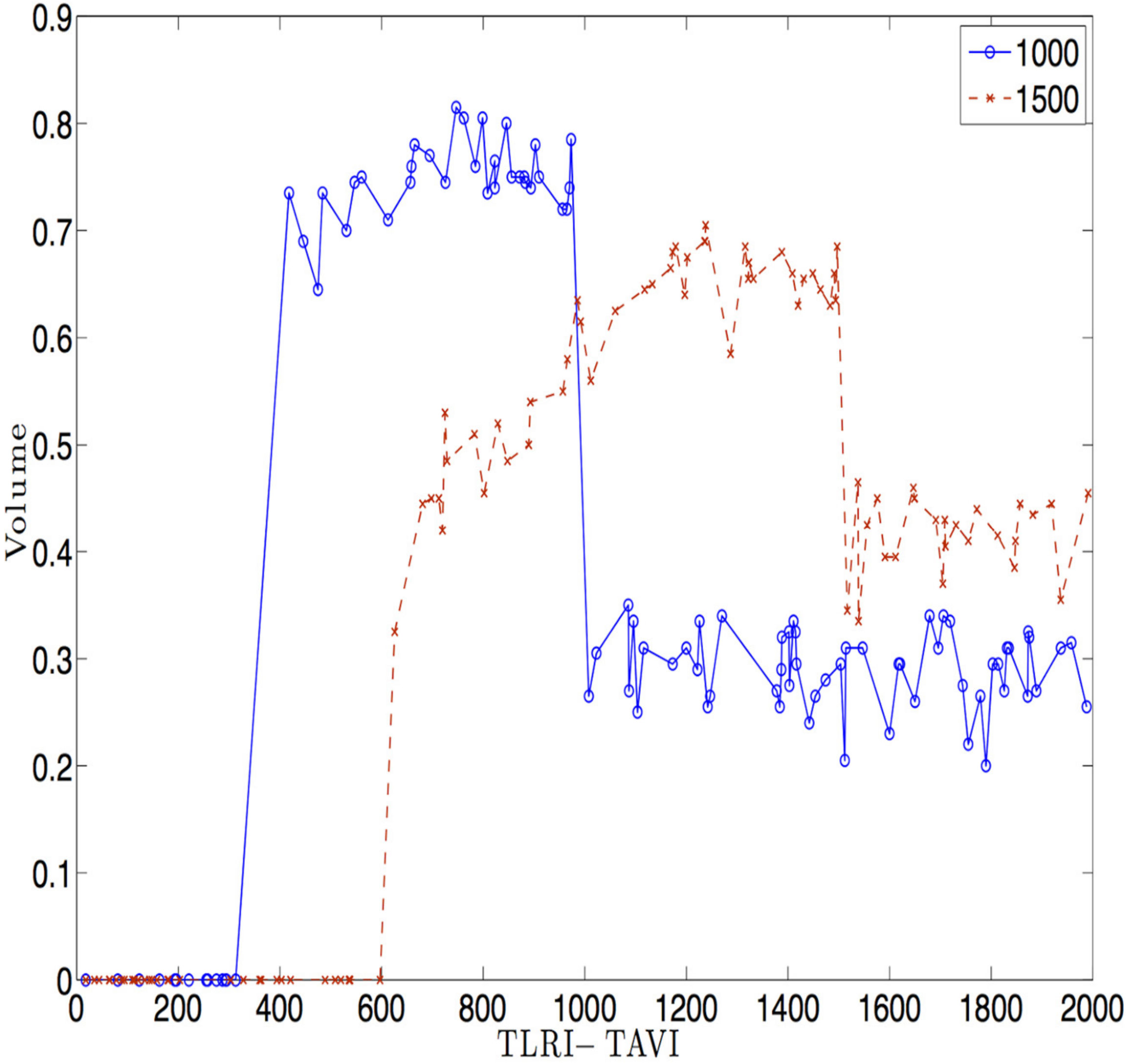


$$1 \cdot \#_0^{\tau}\mathtt{AP} + 2 \cdot \#_0^{\tau}\mathtt{VP} \leqslant \mathsf{E}$$

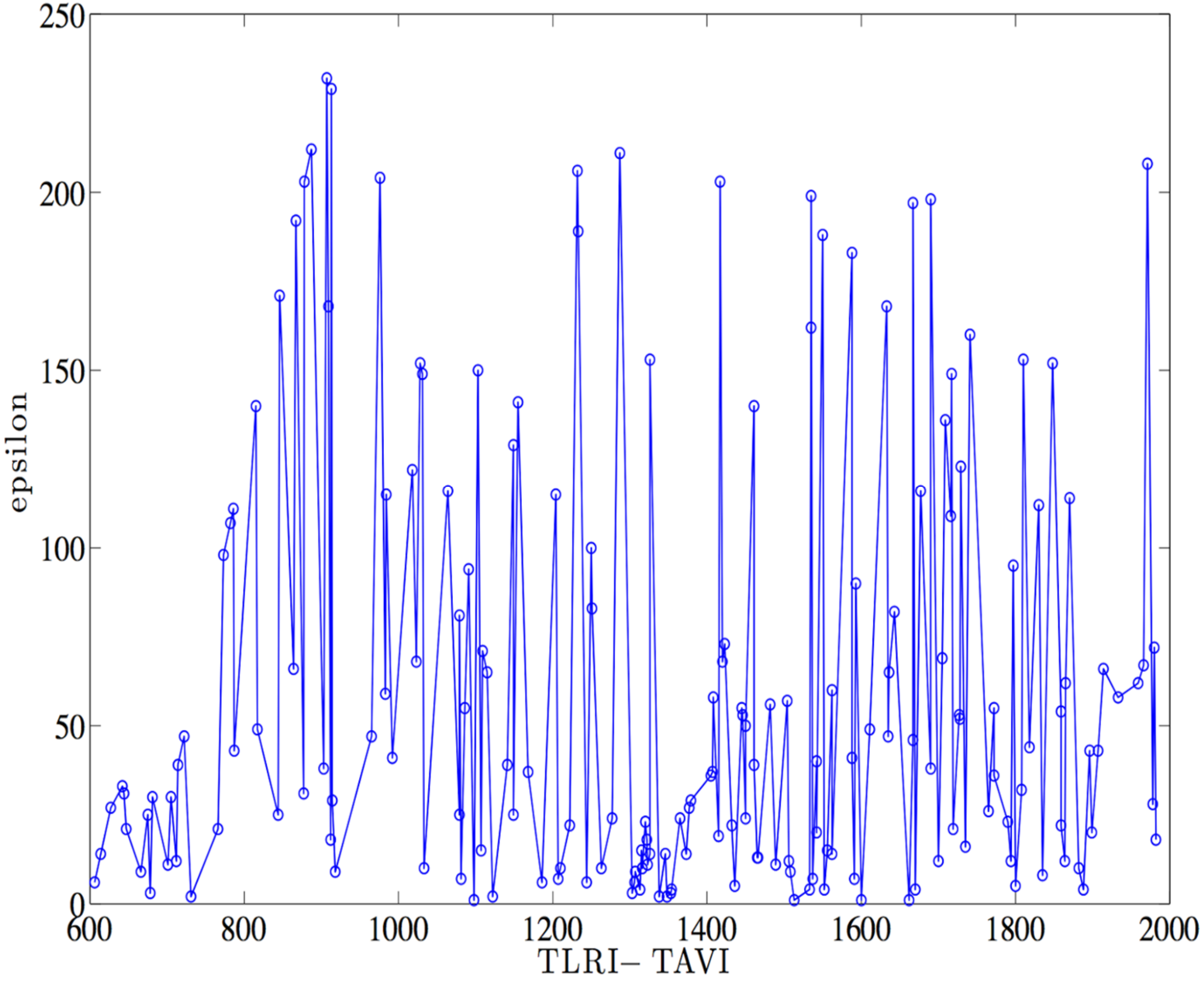
$$\Box^{[0,\tau]}(\#_0^{\tau}\mathtt{Vget} \geqslant B_1 \wedge \#_0^{\tau}\mathtt{Vget} \leqslant B_2)$$

Combine constraint solving with sampling of model parameters: sample, generate untimed path, then generate constraints that satisfy the property.

Results: maximal volume objective



Results: robustness objective



Combined with energy, synthesises robust parameter around 850.

Summing up…

- Medical CPSs, their take up is fast increasing
  - Implantable, closed-loop and wearable devices
  - Software an integrated and critical component
  - 24/7 health performance expectation
  - Safety-critical context, software failures on the increase

- Many scientific and technological challenges remain
  - Huge models!
  - Integration of discrete, continuous and stochastic dynamics
  - Scalability of quantitative verification
  - Accuracy of approximate verification
  - Efficiency of parameter synthesis
  - Model synthesis from quantitative requirements