Physiological Closed-loop Controllers for MDCPS

Rahul Mangharam & George Pappas

{rahulm, pappasg}@seas.upenn.edu

University of Pennsylvania

Interoperability for Patient Safety

- Modern medical care is heavily reliant on devices
  - Sensors: patient monitors, thermometers, glucose meters, EKG
  - Actuators: infusion pumps, radiation therapy, pacemakers
- Caregiver is always in the loop
  - Continuous monitoring is not possible
  - Relies on alarms to detect events
- Alarms are frequently irrelevant (false positive) or ignored (alarm fatigue)

Model-Driven Safety Analysis of Closed-Loop Medical Systems

Miroslav Pajic Rahul Mangharam
Insup Lee Oleg Sokolsky David Arney

Computer and Info. Science Electrical and Systems Engineering
University of Pennsylvania

Julian M. Goldman
Massachusetts General Hospital & CIMIT

Penn Engineering

PRECISE
PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

Overview

1. Patient-in-loop Medical Case Study

2. Modeling and analysis
   1. Continuous time detailed Matlab Model
   2. Timed Automata abstract UPPAAL Model

3. Discussion

PCA Case Study

- Patient Controlled Analgesia
  - Common technique for delivering pain medication (opiates)

1. Patient presses a button to request a dose

2. Overdoses result in respiratory distress, ultimately death

3. Pumps have safeguards, but overdoses can still happen

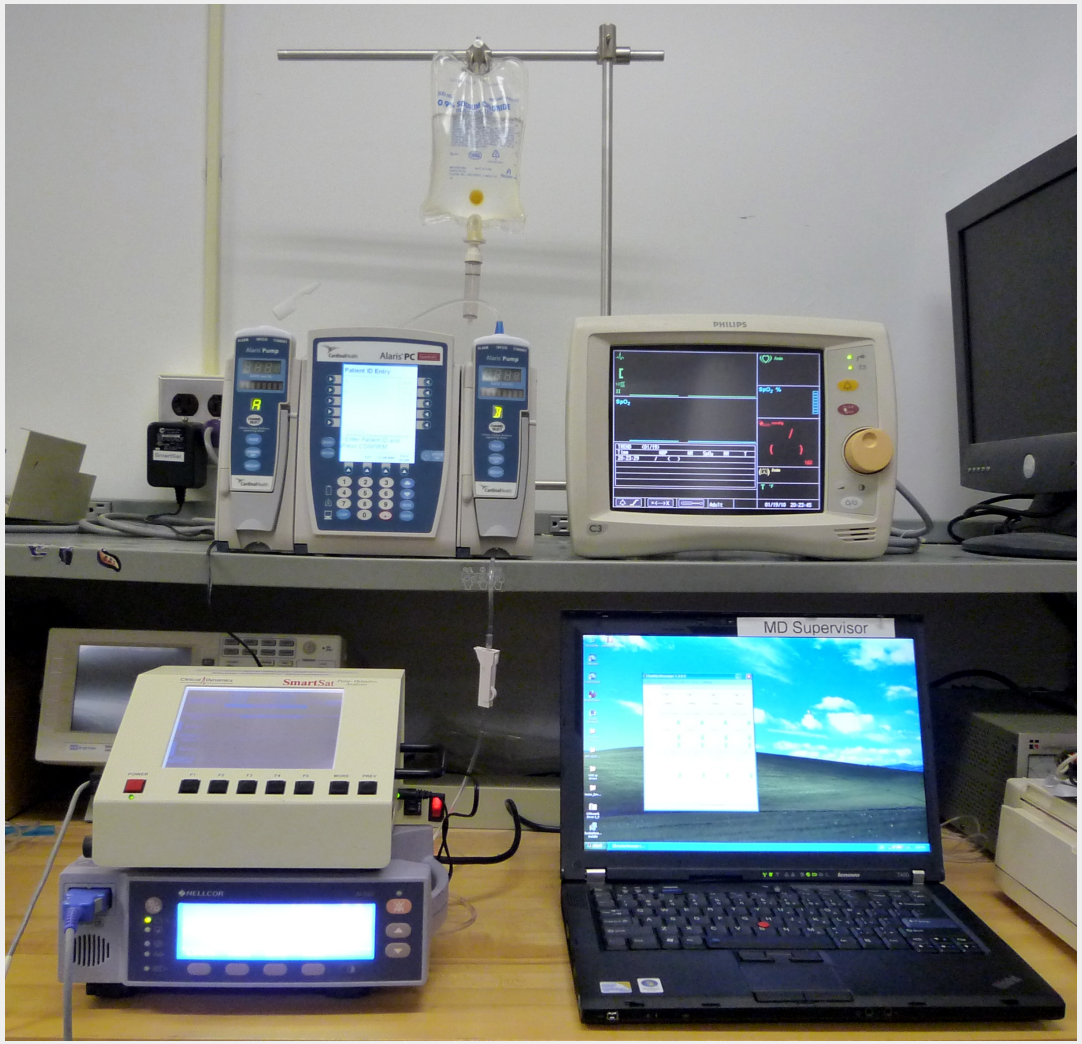
4. PCA is a significant source of adverse events

Challenges with Physiological Control Systems

- Patient Modeling
  - People are unpredictable ☺
    - Non-deterministic, uncertainty, variation
  - Models do not exist or are too complex
    - 80+ states, 100s-100Ks ODEs – online model adaptation is hard
  - Under-actuated system with limited observability
- Verifying Safety Properties
  - Individual devices and whole system
- Physical connectivity and communication infrastructure
- Regulatory Challenges
  - Who is the manufacturer of the composed system?

Case Study Components

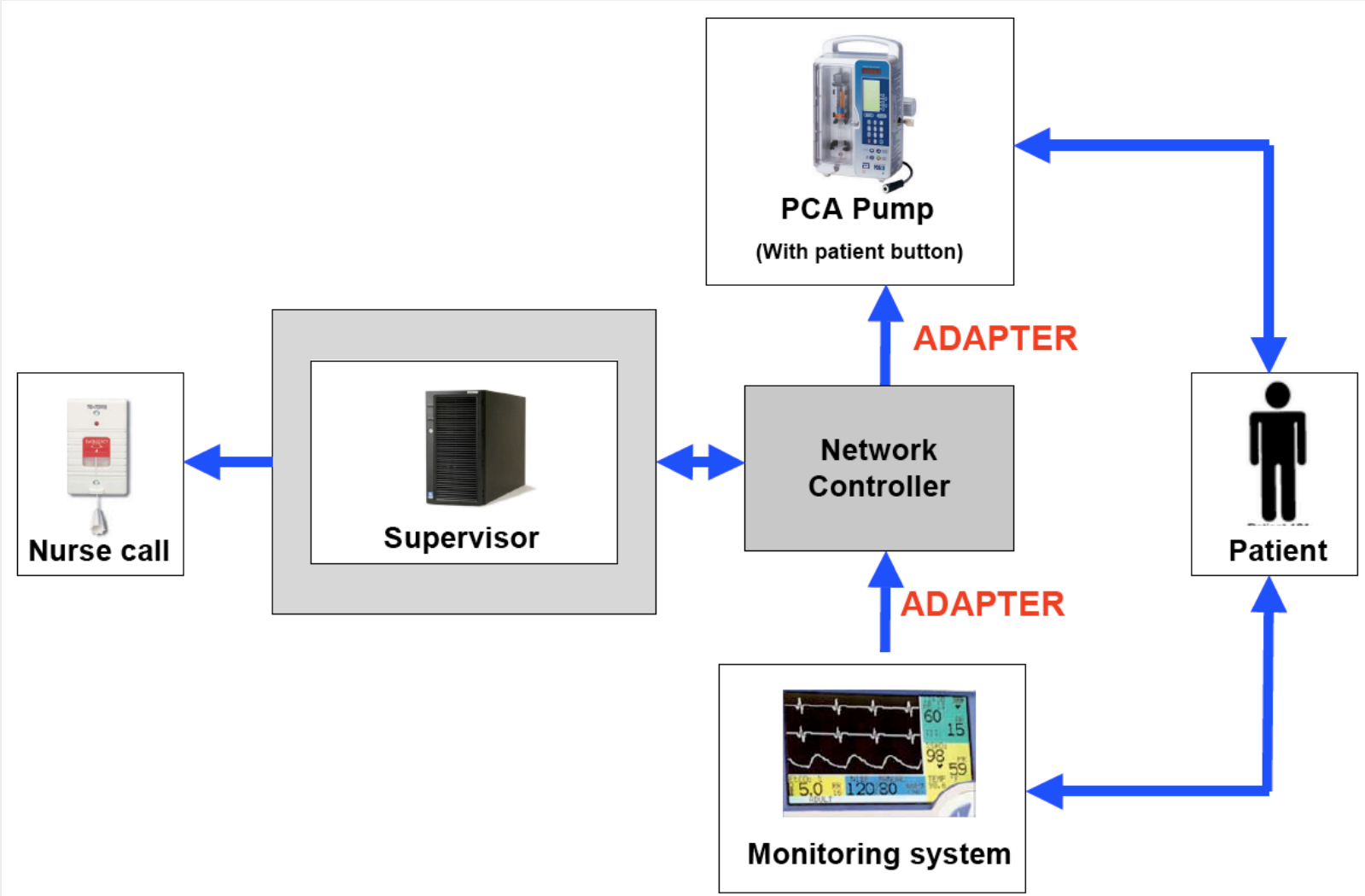
- PCA infusion pump
- Pulse-Oximeter
- Supervisor
- Patient Model



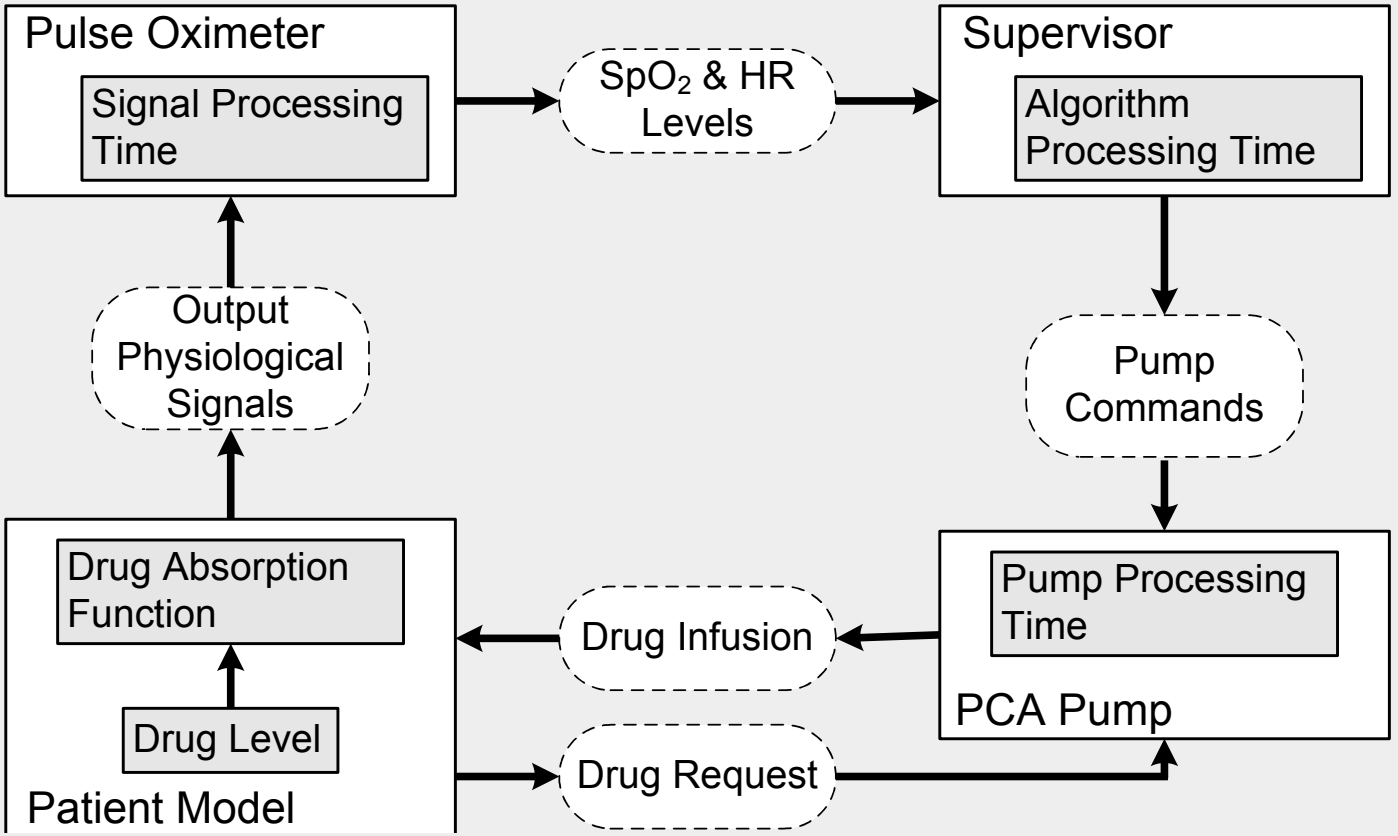
Penn Engineering

PRECISE
PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

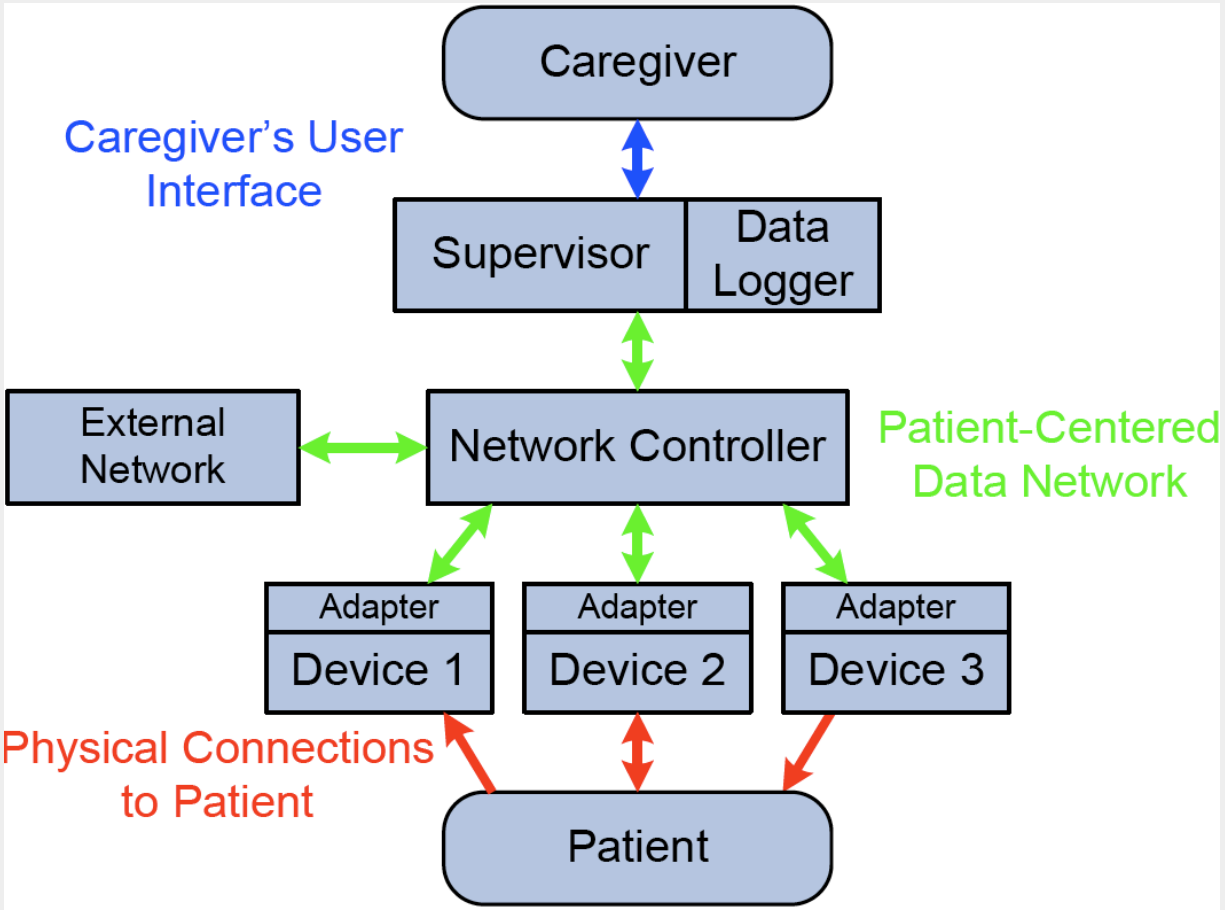
Case Study Components



Control Loop

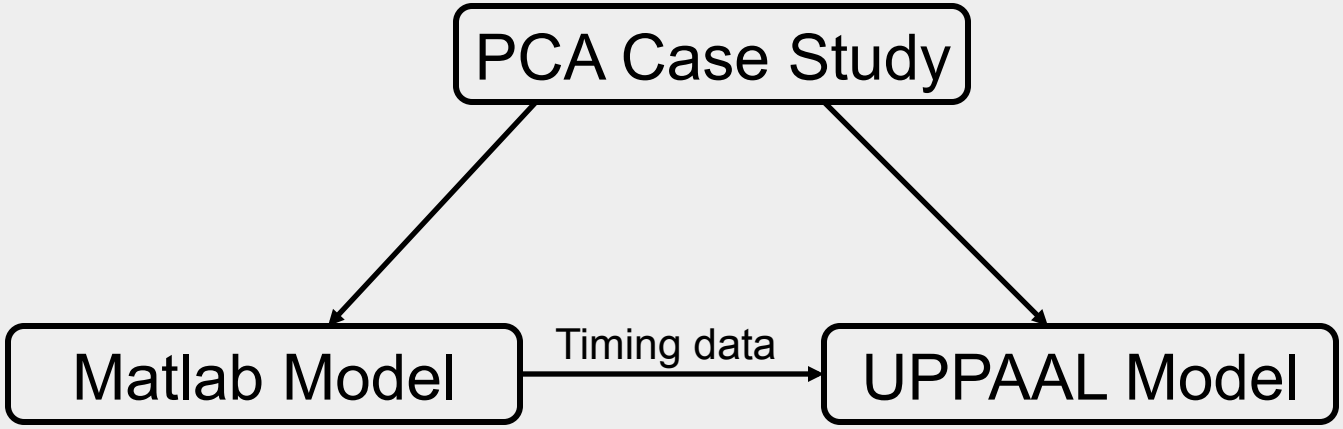


Supervisory Control



Modeling approach

- Matlab / Simulink model captures continuous dynamics
- Simulation provides timing data to tune the more abstract UPPAAL model
- Formal verification in UPPAAL



3-Compartment LTI Patient Model

Derived from pharmacokinetics model for intravenous delivery of anesthetic drugs

Blood Plasma Volume

$$\begin{bmatrix}\dot{C}_1\\ \dot{C}_2\\ \dot{C}_3\end{bmatrix} = \underbrace{\begin{bmatrix}-(k_{12}+k_{13}+k_{10}) & k_{21} & k_{31}\\ k_{12} & -k_{12} & 0\\ k_{13} & 0 & -k_{31}\end{bmatrix}}_{\mathbf{A}} \begin{bmatrix}C_1\\ C_2\\ C_3\end{bmatrix} + \underbrace{\begin{bmatrix}\frac{1}{V_1}\\ 0\\ 0\end{bmatrix}}_{\mathbf{B}} I$$

Drug conc.

Medication Level In Patient's body

$$dl = \underbrace{\begin{bmatrix}1 & 0 & 0\end{bmatrix}}_{\mathbf{C}} \begin{bmatrix}C_1\\ C_2\\ C_3\end{bmatrix}$$

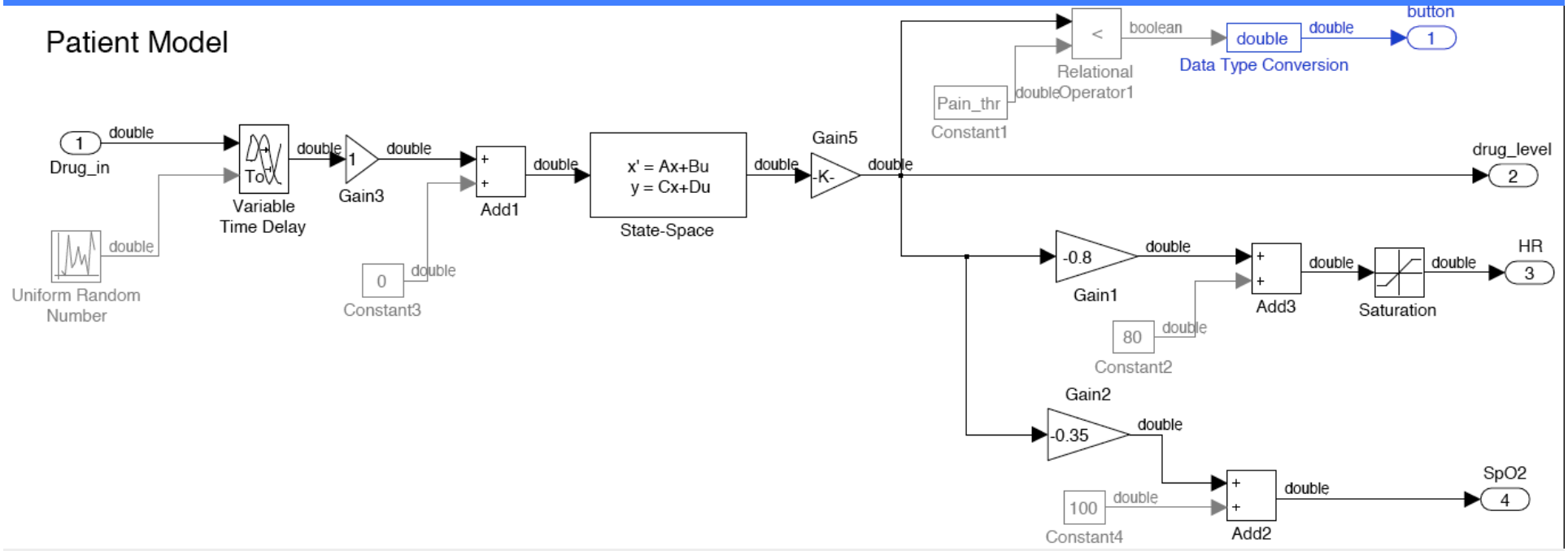
Modeling Patient specific behavior – model with uncertain parameters

$$k_{ij} \in \left[\hat{k}_{ij} - \Delta k_{ij}, \hat{k}_{ij} + \Delta k_{ij}\right]$$

$$V_1 \in \left[\hat{V}_1 - \Delta V, \hat{V}_1 + \Delta V\right]$$

3-Compartment Patient Model

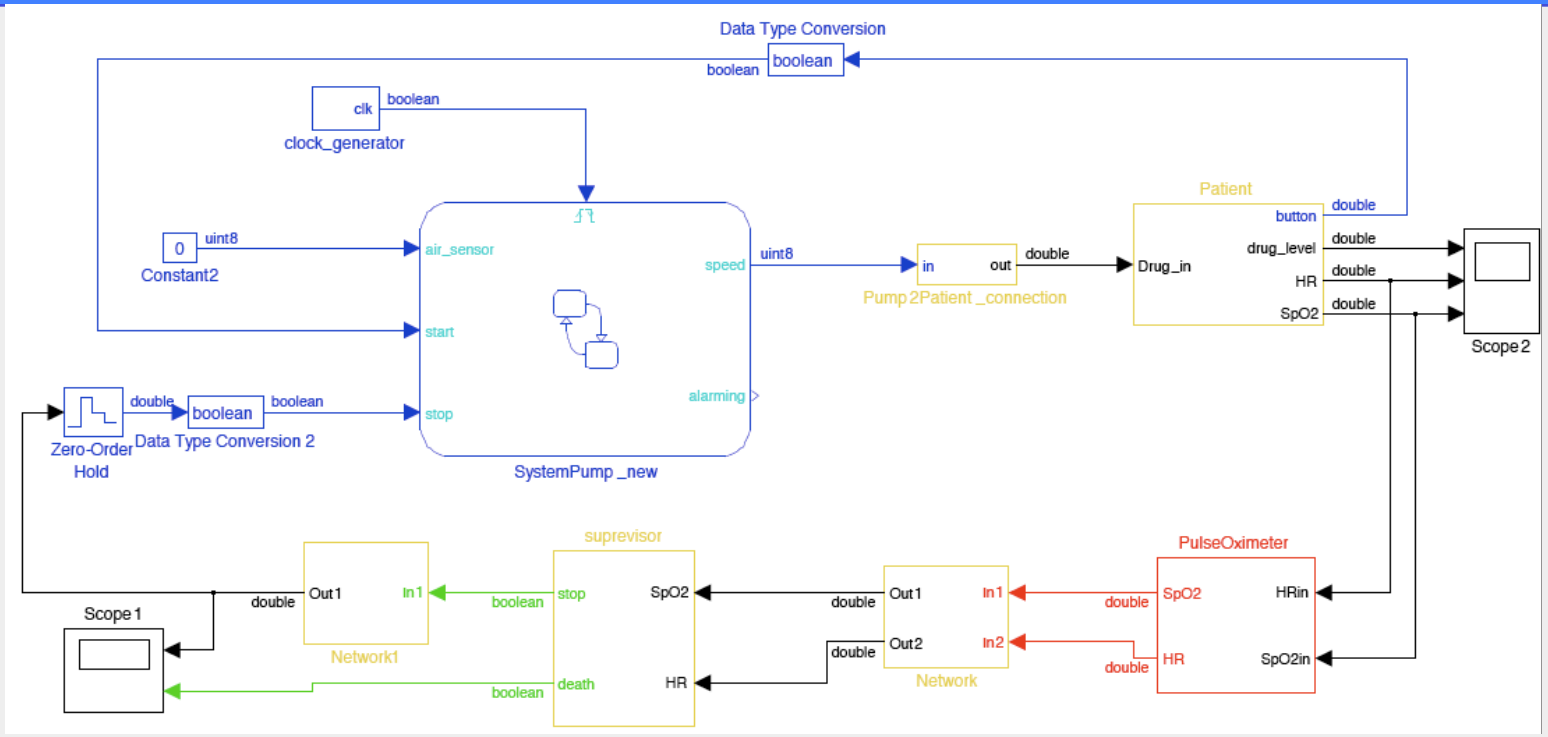
in Simulink



- Captures the dynamics of the PCA pump, pulse-oximeter, patient model, and supervisor
- Defines safe, critical, and alarming regions
- Simulations of the model allow us to estimate t_{crit}
- Allows us to study effects of faults

3-Compartment Patient Model

in Simulink



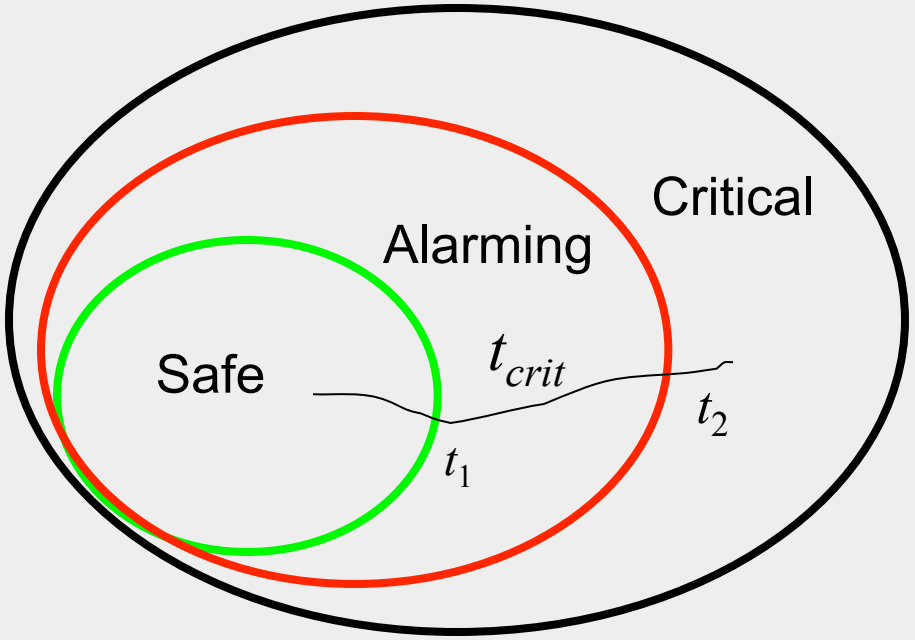
- Captures the dynamics of the PCA pump, pulse-oximeter, patient model, and supervisor
- Defines safe, critical, and alarming regions
- Simulations of the model allow us to estimate t_{crit}
- Allows us to study effects of faults

Patient Model

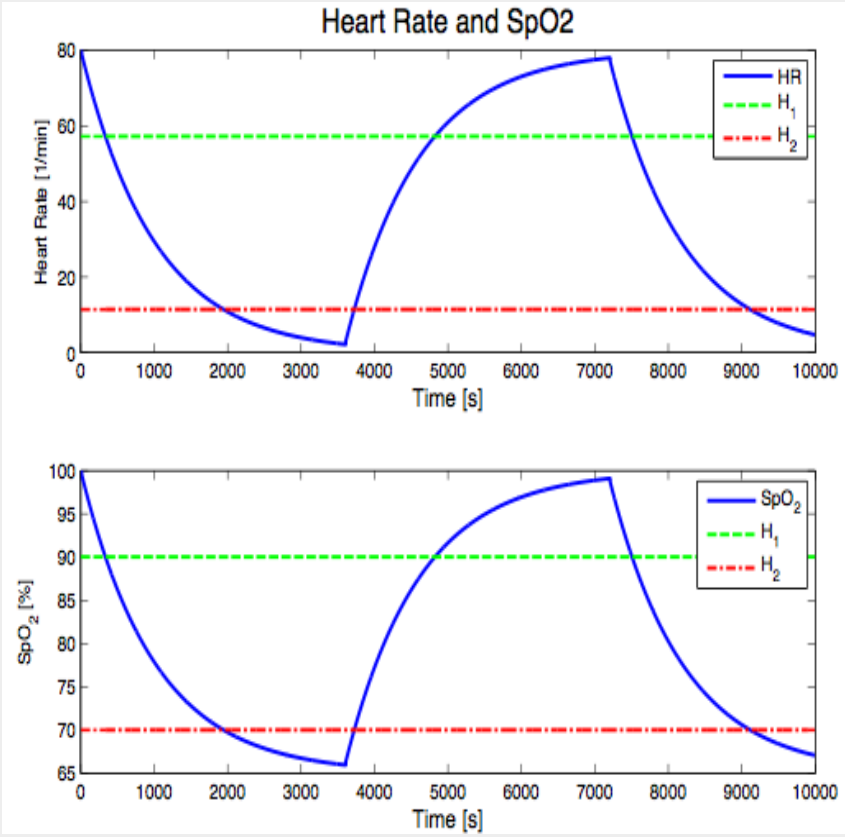
First-order continuous system

$$SpO_2 = C_{\min} + Ae^{-\alpha t}$$

Patient Critical Regions

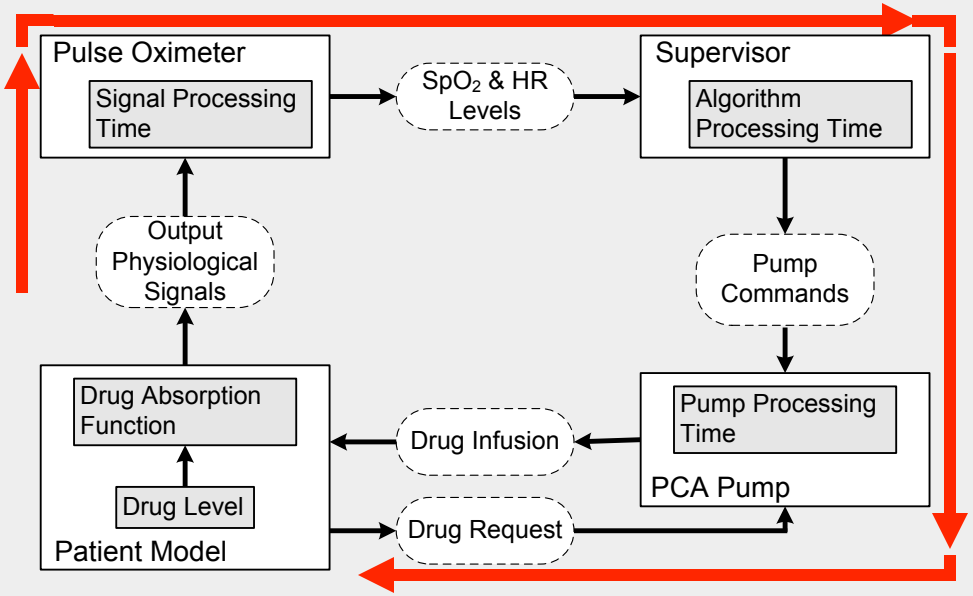


Patient Response to Drug



Key Safety Property

Pump stops in time if total delay <= t_{crit}



Total delay is the sum of:

- tPOdel: worst case delay from PO (1s)
- tnet: worst case delay from network (0.5s)
- tSup: worst case delay from Supervisor (0.2s)
- tPump: worst case delay from pump (0.1s)
- tP2PO: worst case latency for pump to stop (2s)
- tcrit: shortest time the patient can spend in the alarming region before going critical

Obtaining t_{crit}

- For our patient model, determine t_{crit} analytically given the drug level (C), Heart Rate (H_1), SpO_2 (H_2)

$$t_{crit} = \frac{1}{\alpha} \log \frac{H_1 - C_{\min}}{H_2 - C_{\min}}$$

- In a more complex case, obtain through Matlab simulation
- For a more precise result, a modal value can be derived
  - E.g., account for patient context such as weight.

t_{crit} – for cases with uncertainty

- For the patient model with fixed parameters t_{crit} determined **analytically**
- For model with uncertain parameters
  - Matrices **A**, **B**, **C** belong to regions
  - Providing an upper bound on t_{crit}

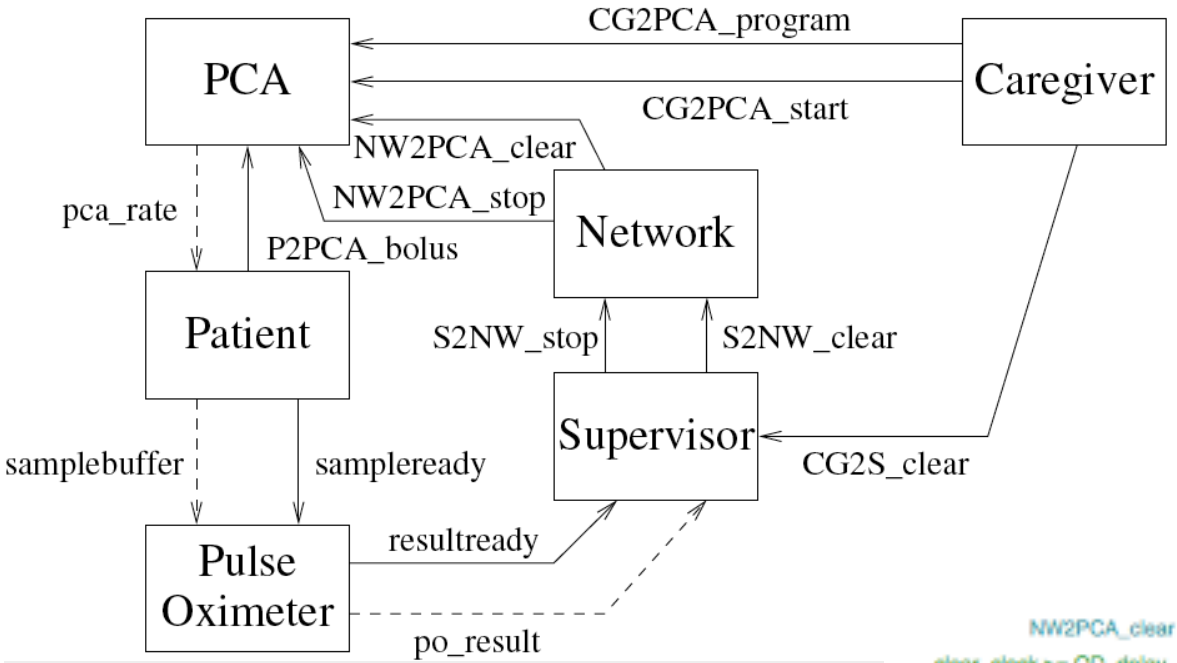
$$\tilde{t}_{crit} = \frac{1}{||\tilde{\mathbf{A}}||} \ln \left(\frac{\frac{|\Delta H|}{gain}}{||\tilde{\mathbf{C}}|| \cdot \left(||\tilde{x}_0|| + \frac{||\tilde{\mathbf{B}} u_i||}{||\mathbf{A}_{max}||} \right)} + 1 \right)$$

$$\tilde{\mathbf{A}} = \underset{\mathbf{A} \in \Re\{\mathbf{A}\}}{\operatorname{argmax}} ||\mathbf{A}||, \tilde{\mathbf{B}} = \underset{\mathbf{B} \in \Re\{\mathbf{B}\}}{\operatorname{argmax}} ||\mathbf{B}\mathbf{u_i}||, \tilde{\mathbf{C}} = \underset{\mathbf{C} \in \Re\{\mathbf{C}\}}{\operatorname{argmax}} ||\mathbf{C}||$$

$$\mathbf{A}_{min} = \underset{\mathbf{A} \in \Re\{\mathbf{A}\}}{\operatorname{argmin}} ||\mathbf{A}||$$

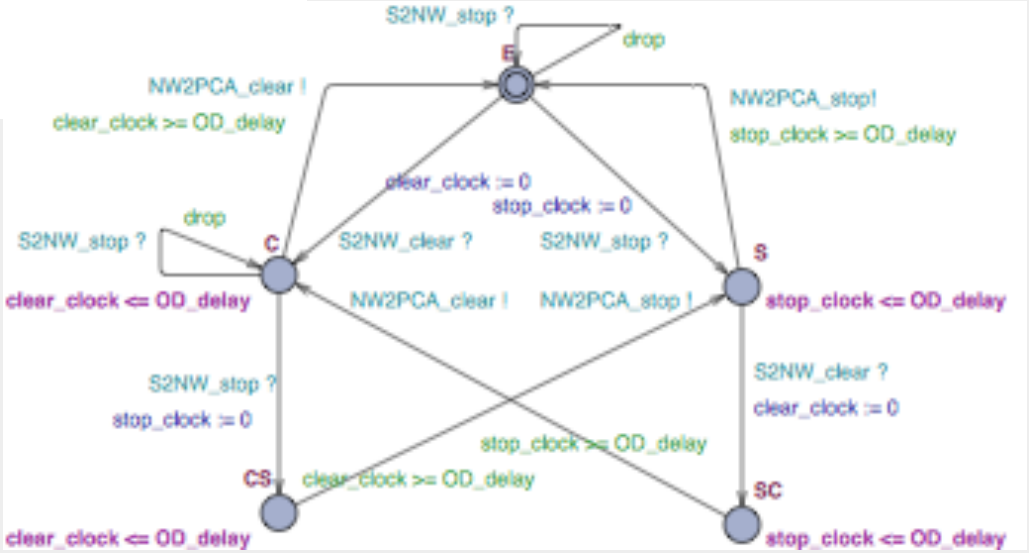
- In a more complex case, obtain using Matlab simulation

UPPAAL Model



Using Synchronous communication channels and Shared variables between components

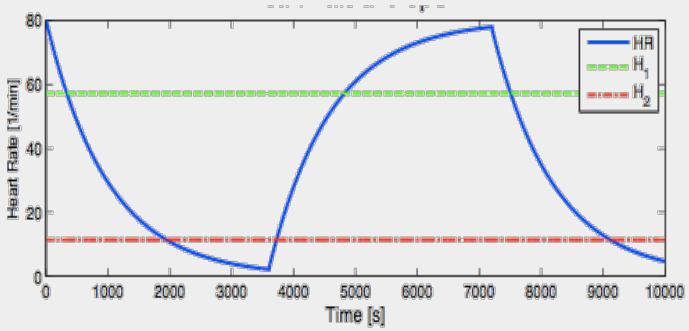
Network Component

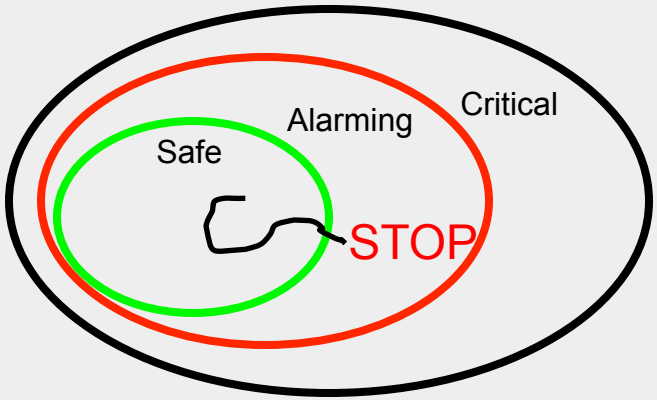


Properties verified with UPPAAL

- **Once SpO2 drops below pain threshold, it eventually goes back up**

  A[] (samplebuffer < pain_thresh -> A <> samplebuffer >= pain_thresh)



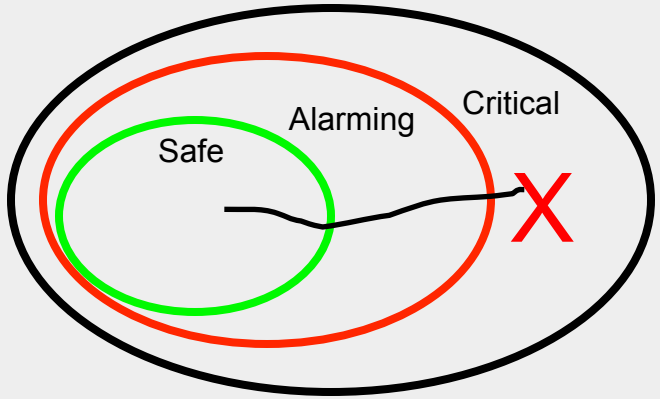


The pump is stopped if patient enters *alarming*

A[] (samplebuffer < alarm_thresh ->
A<> (PCA.Rstopped V PCA.Bstopped)

The patient can not go into the critical region

A[] (samplebuffer >= critical)



Effects of unreliable network

- Problem:
  - The pump may not receive stop commands.
- Solution:
  - Instead of sending simple start and stop commands, send a command giving the pump permission to run for a certain period of time.
- Open-loop stability
  - We need to determine how long the pump can run without endangering the patient

System Implementation

- FPGA boards for the device interfaces and real-time network
- Real devices where possible
- Homegrown pump prototype for control

Penn Engineering

PRECISE
PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

Conclusions

- Medical CPS offer plenty of challenging problems that urgently need solutions
- Not all of these problems are technical
  - Some are organizational, cultural, etc.
- We presented first step
  - Case study of a real clinical problem
  - Modeling approach combines simulation and formal verification
- But much research is still needed

Future work

- Better patient model
  - More realistic dynamics, parametric variability
  - More sophisticated control-theoretic analysis
- Sensor fusion
  - Better reliability
  - Faster detection
- Safety in dynamically created scenarios
  - Compositional reasoning?
  - Safety case construction
- Modeling of clinical scenarios
  - Workflows, requirements for devices, safety criteria

Current and Future Research

Closed-loop Glucose Control Systems

George Pappas *Sanjian Chen*
Insup Lee *Oleg Sokolsky*

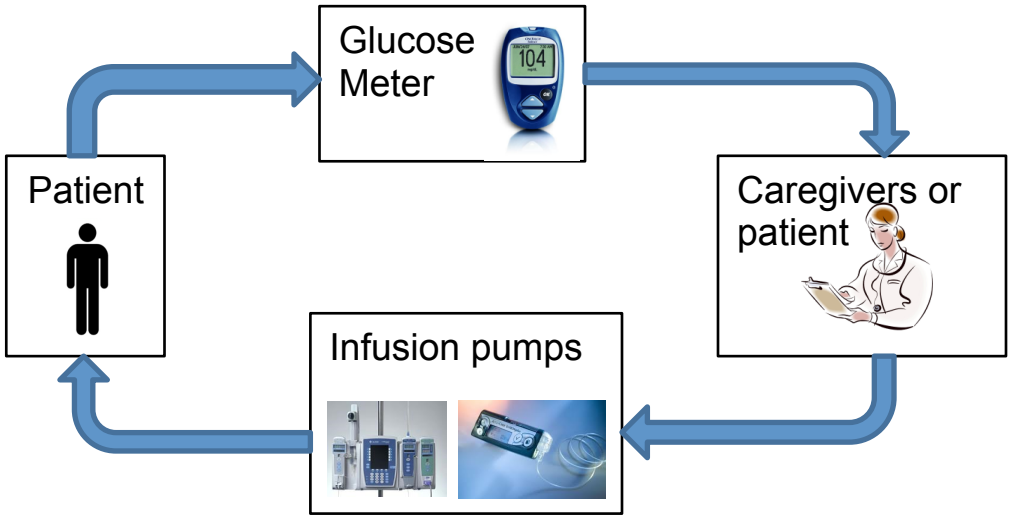
Hospital at University of Pennsylvania

Penn Engineering

PRECISE
PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

Diabetes & Glucose Control Sytems

- Diabetes: a growing problem
  - **26 million (8.3% of the population) in US** have diabetes
  - 7-th leading cause of death
  - Costs $174 billion annually
  - 5-10% are Type 1 (T1D), others are Type 2 (T2D)
- Improved blood glucose regulation benefits
  - maintain glucose level within certain ranges

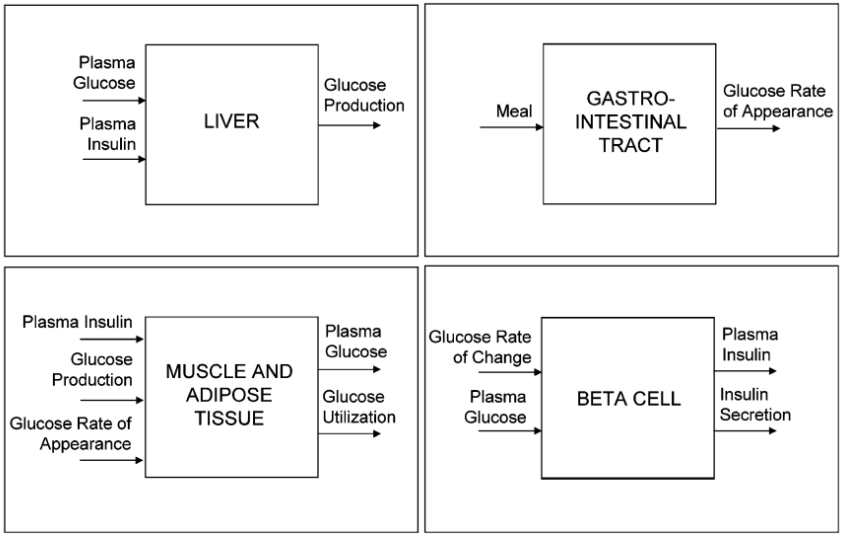


Research Objective

- Model-based development
  - Needs patient model and controller model
- Safety property: patient's physiological states never become critical
  - Hypoglycemia & Hyperglycemia
  - In present of hazardous situations and uncertainties in environment, e.g., component failures, delay food feedings
- Validation and verification

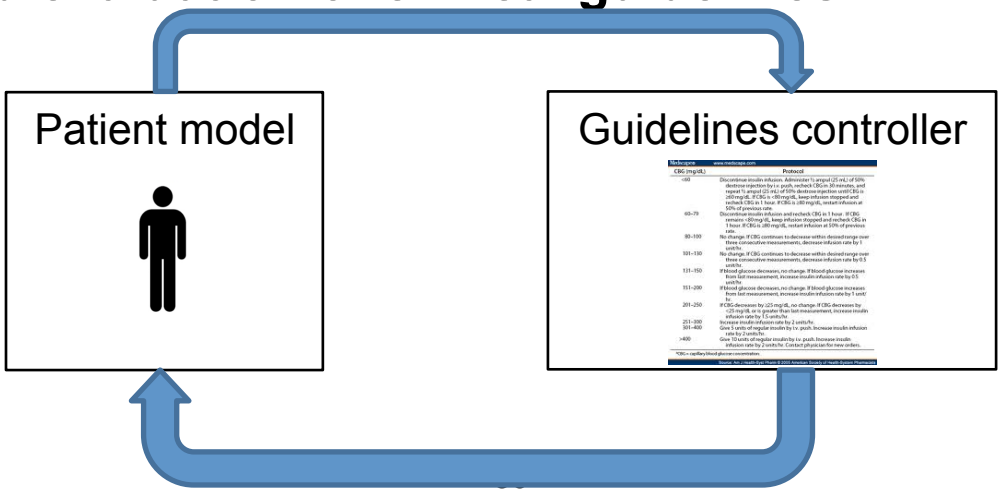
Background: Patient Model

- Modeling the human glucose-insulin dynamics
  - 60's: simplest linear model by Bolie
  - 70's – 80's: minimal (coarse-grain) modeling strategy
  - 90's – now: maximal (fine-grain) models
    - High-order nonlinear model with many **unknown parameters**
    - Not easily identifiable
    - Man et al., 2007, meal simulation model



Guideline-based Controller (1)

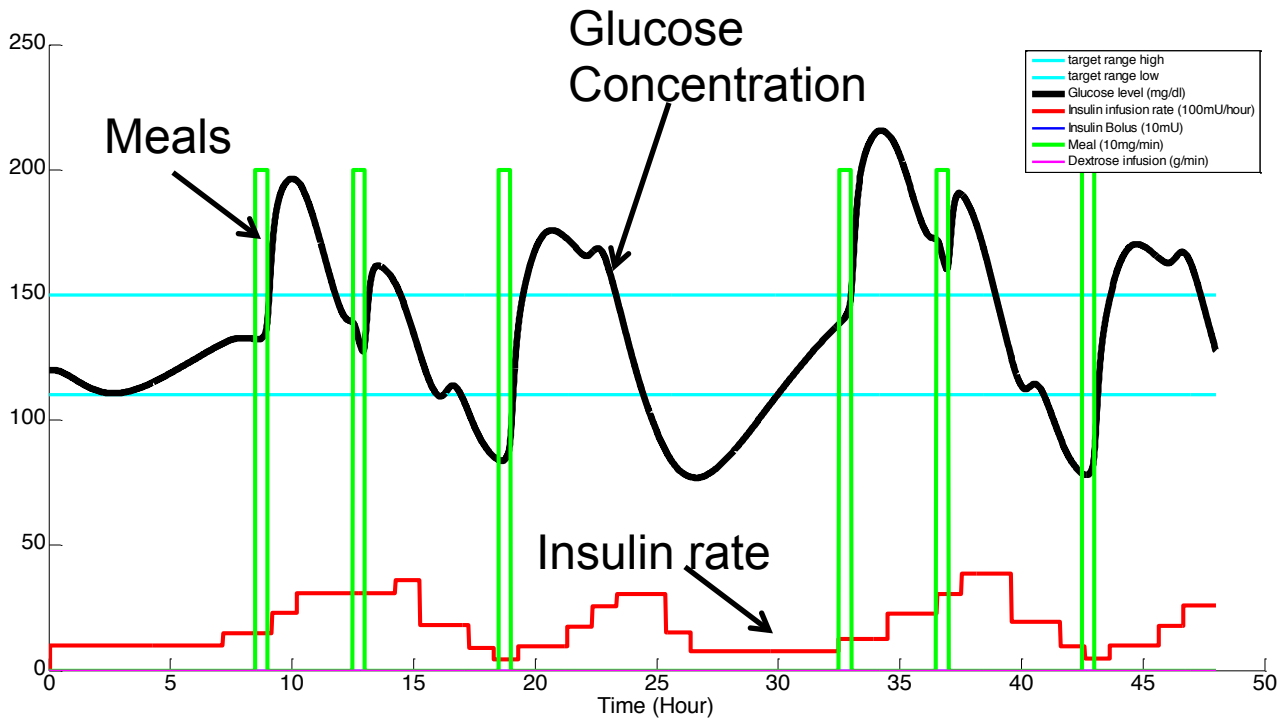
- **Controller: clinical guidelines**
  - 5 ICU insulin infusion guidelines from a hospital
  - programmed as rule-based controllers
- **Patient model:UVa/Padova T1DM Metabolic Simulator***
  - Based on a maximal model (Man et al., 2007)
  - 30 “virtual” subjects settings
  - Full version (with 300 virtual subjects) approved by FDA in 2008 to substitute animal trials in the pre-clinical testing of certain control strategies
- **Model-based evaluation of clinical guidelines**



*: © 2011, The Epsilon Group

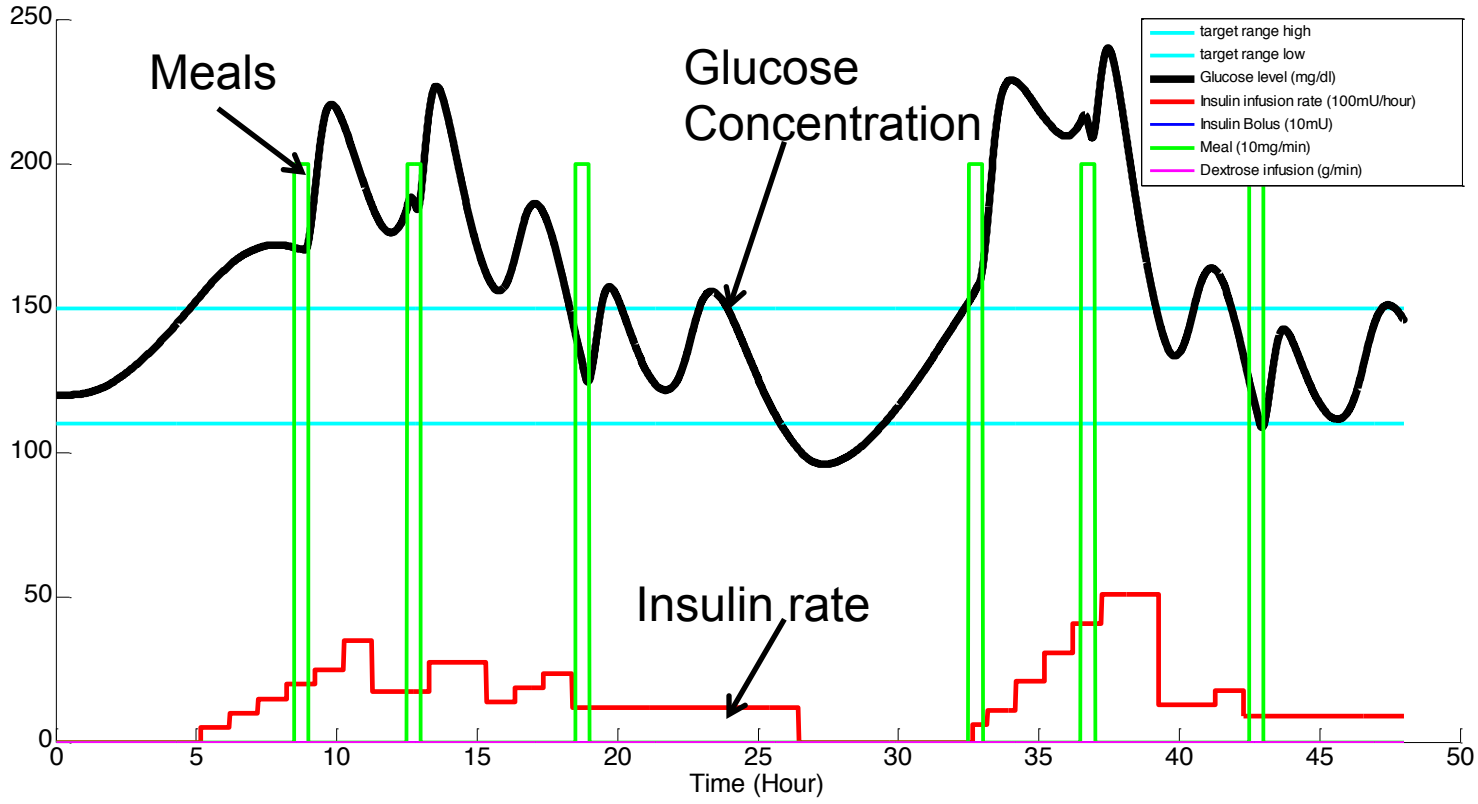
Guideline-based Controller (2)

- Guideline controls are not always effective
- Hypoglycemia (low glucose) and **serious oscillations** in glucose level observed on some virtual subjects
  - Example:



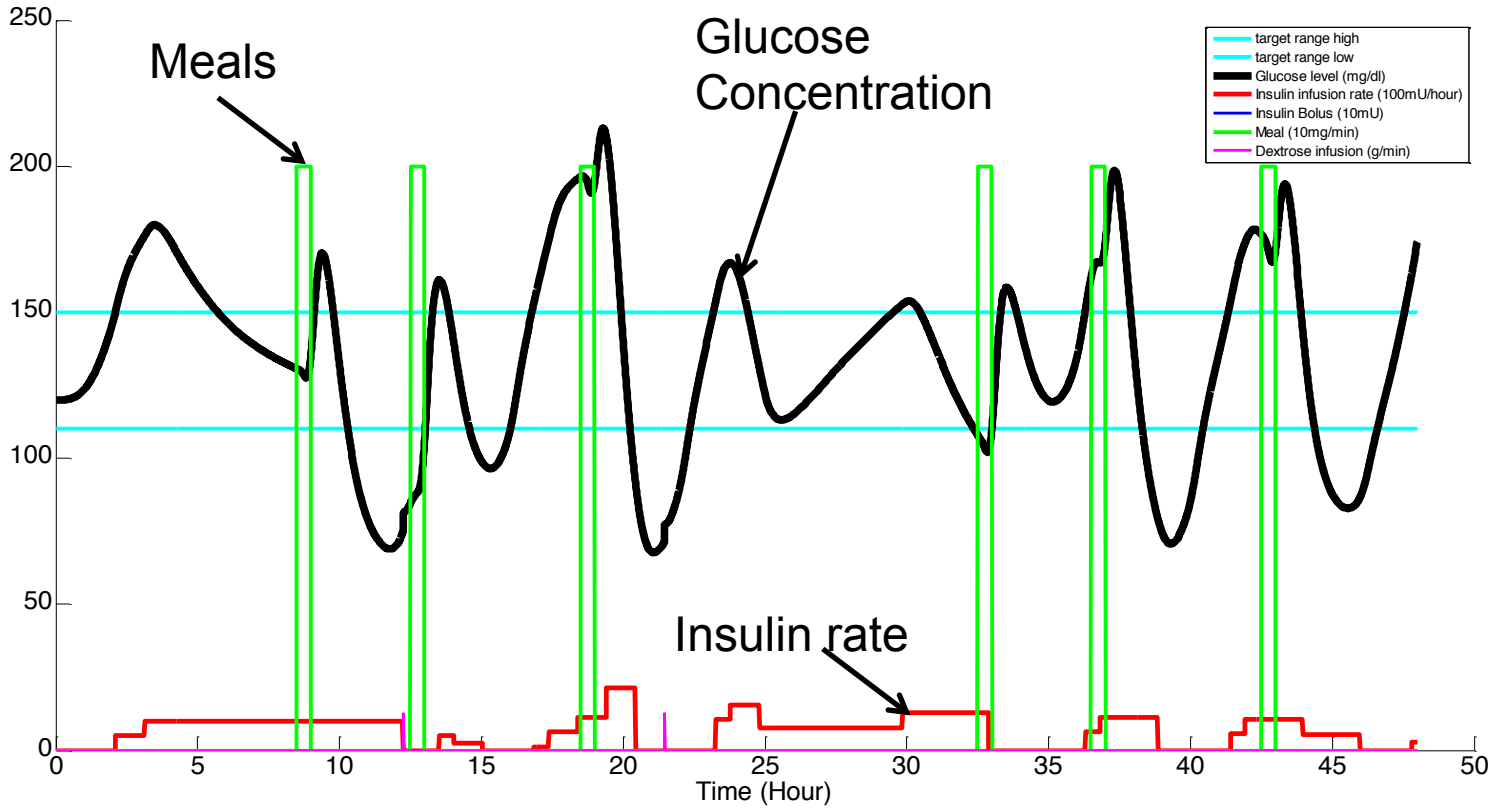
Guideline-based Controller (3)

Some subjects are **more resistant to insulin**



Guideline-based Controller (4)

Some subjects are **sensitive to insulin**



Guideline-based Controller (5)

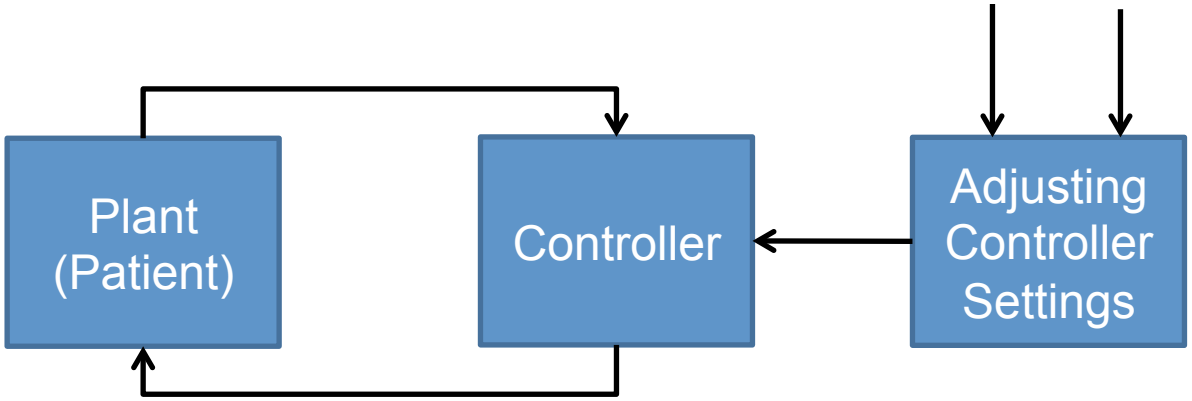
- Clinical guidelines use fixed rule tables
  - Not adaptive to inter-subject variability within the same patient population
- Need more effective controllers for the networked control system

Outline

- Introduction
- Our Vision
- Current state of affairs
- Our Approaches
  - Model-based safe adaptive/robust control
  - Simulation/testing based verification

Adaptive/Robust Control

- Deal with physiological parameter uncertainties
  - *Adaptive approach:*
    - Adjusting controller settings at run-time
    - Explicit adaptive control: learn model parameters at run-time
      - Difficult for a ~20-D non-linear model with ~30 parameters
    - Implicit adaptive control may apply
  - *Robust approach:*
    - stabilize the plant with bounded parameter uncertainties
- Challenges: **verification** of adaptive/robust controller

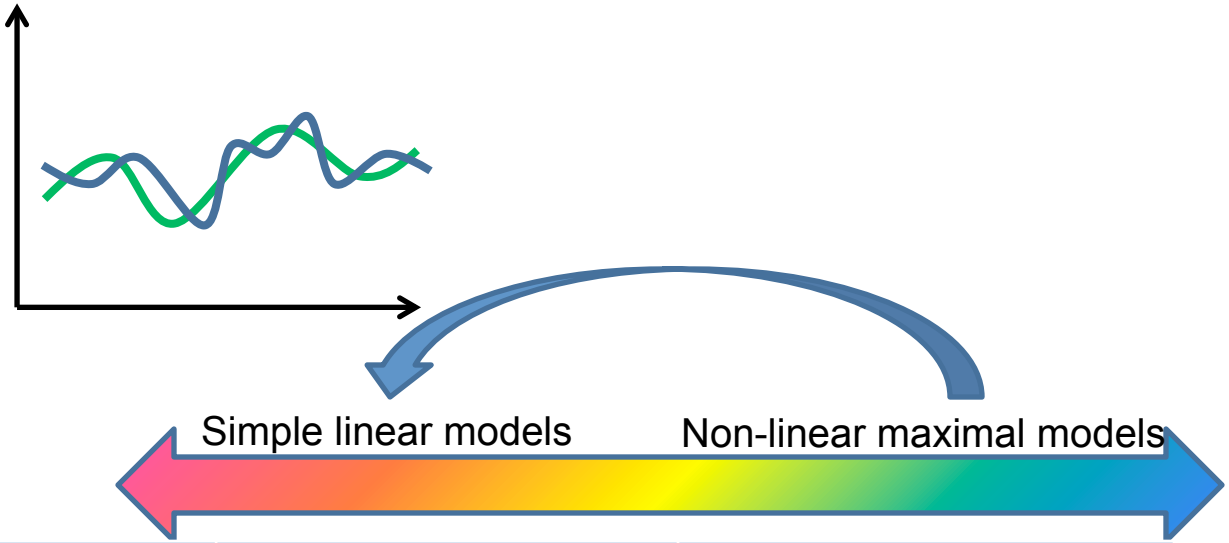


Safe Adaptive Exploration

- Adaptive control often involves learning the parameters by feeding in extreme inputs
  - Example: aggressively turning a car
- Not safe for patient-in-the-loop systems
- Open issue: adaptive exploration with safety constraints

Safe Non-linear Model Reduction

- Model complexity trade-off
- Reduction with bounded discrepancy

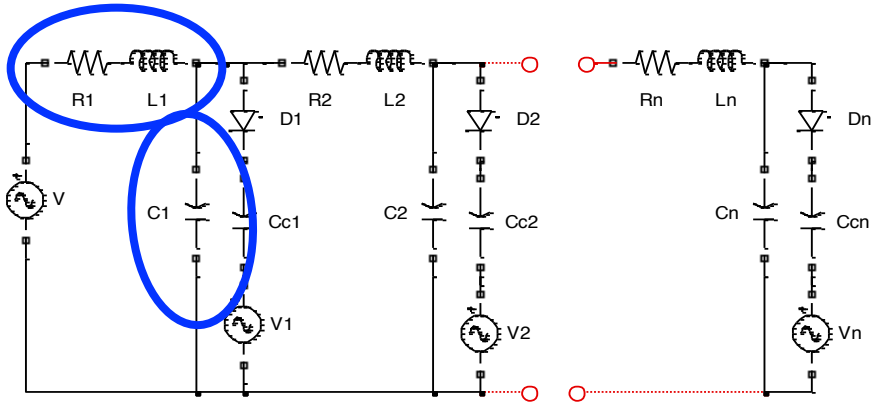


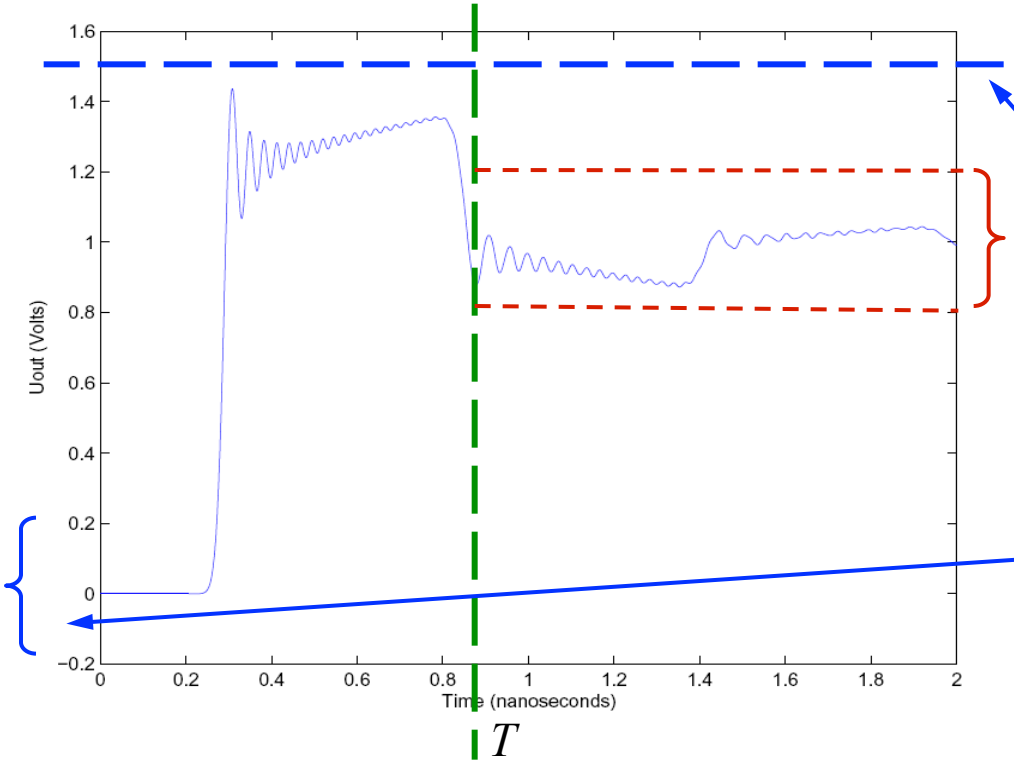
System identification	✓	✗
Controller design	✓	✗
Testing and verification	✓	✗
Model accuracy	✗	✓

Outline

- Introduction
- Our Vision
- Current state of affairs
- Our Approaches
  - Model-based safe adaptive/robust control
  - Simulation/testing based verification

Robust Verification for Linear Systems: Example





System:

$$\dot{x}(t) = A_i x(t) + b_i U_{in}(t)$$

$$U_{out}(t) = Cx(t)$$

Step input $(t > 0)$:

$$U_{in}(t) = 1$$

Steady state at $t = 0$:

$$x(0) = -A^{-1} b U_{in}(0)$$

Property:

$$\Phi = \mathrm{G}\ \pi_1 \wedge \mathrm{F}_{[0,0.85]} \mathrm{G}\ \pi_2$$

$$O(\pi_1) = [-1.5, 1.5]$$

$$O(\pi_2) = [0.8, 1.2]$$

Initial conditions:

$$U_{in}(0) \in [-0.2, 0.2]$$

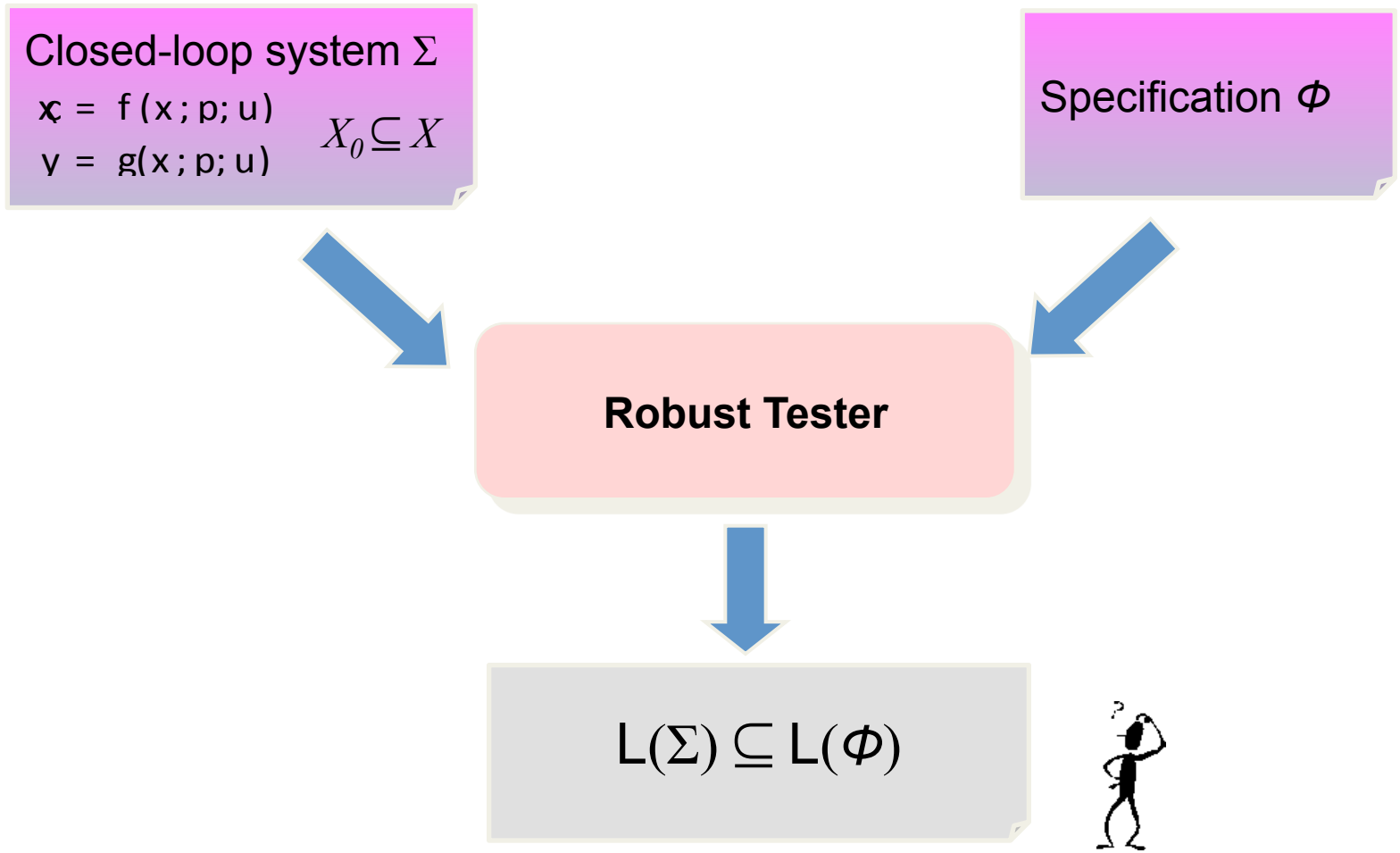
Uncertain parameters

$$e.g.\ C \in [a_1, a_2]$$

This is a transmission line system. A ~80 dimension linear system. The property we want to verify is that the output y(t) globally stays within Pi_1 (-1.5,1.5), and y(t) enters Pi_2 (0.8,1.2) within [0,0.85] time interval. Such kind of properties are closely related to common control performance metrics like rise time, settling time, constraints on input/state, etc. This shows the properties we are interested in and how to interpret the properties

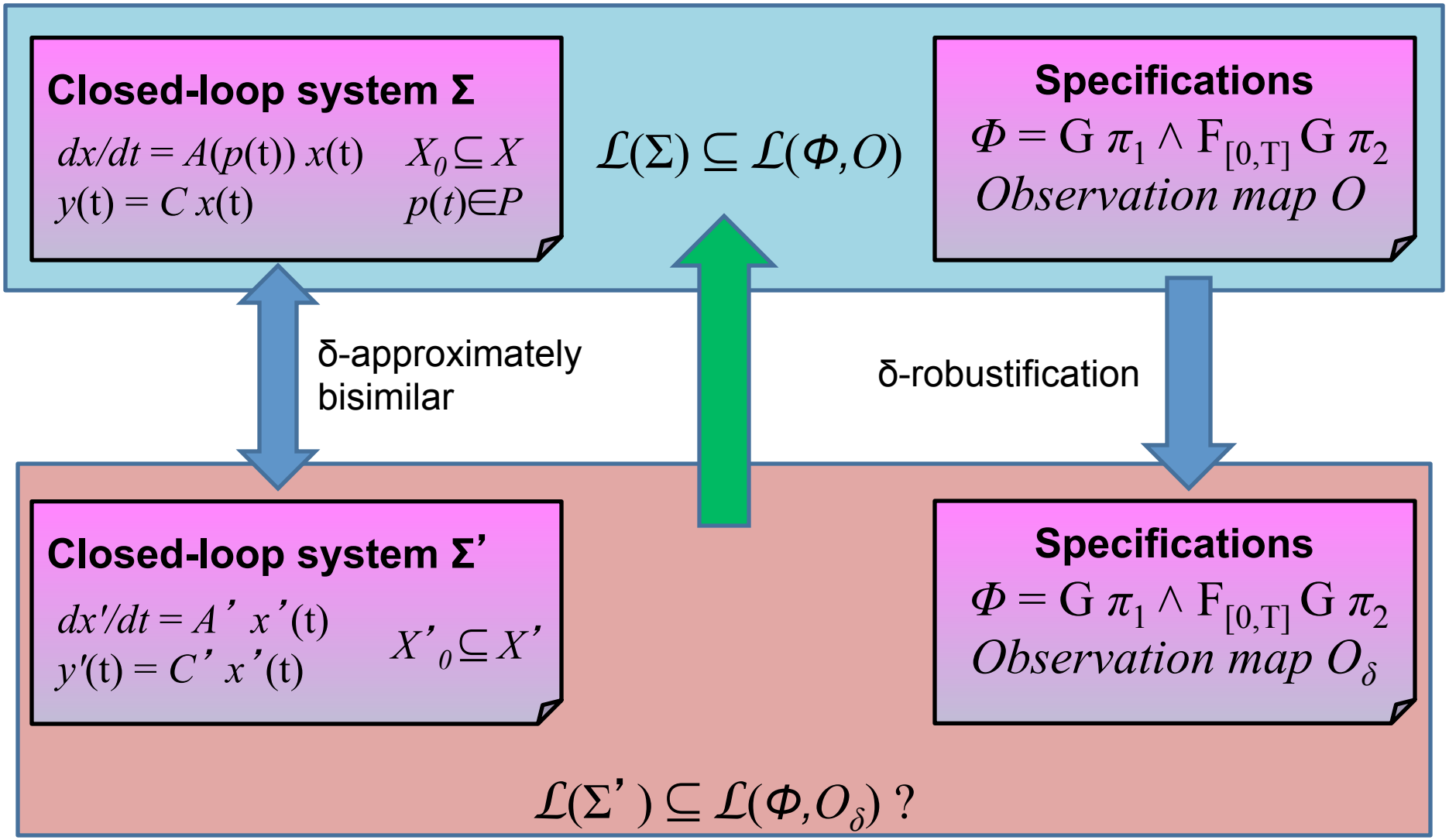
Problem Formulation

Given a closed-loop system model, and a set of specifications, we want a tester to tell whether the system satisfies the specifications, in the sense that the set of all possible system traces is a subset of all traces on which the specifications are true



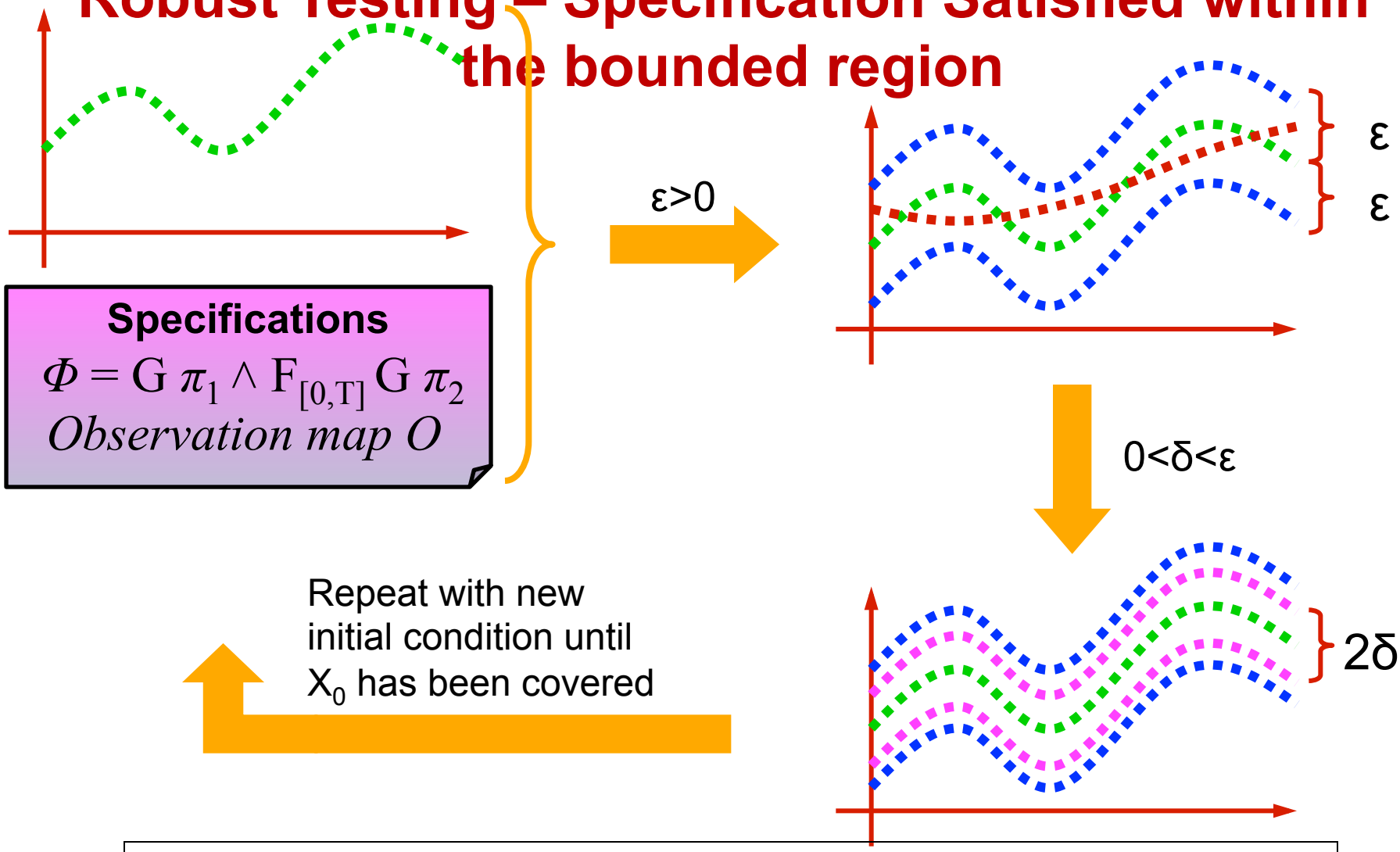
Fainekos, Girard and Pappas, *Temporal logic verification using simulation*, FORMATS 2006
Julius, Fainekos, Anand, Lee, Pappas, *Robust Test Generation and Coverage for Hybrid Systems,* HSCC 2007
Fainekos, Pappas, *MTL Robust Testing and Verification for LPV Systems*, ACC 2009

Solution Overview



The key idea of our solution is: Given the original closed-loop system, where the matrix A depends on some time-varying uncertain parameters p(t), we first try to find a fixed linear system A' that is a close approximation of the original system, in the sense that the output traces of the original system always (despite uncertain p(t)'s) stay within Delta distance to the traces of the reduced system. Then if we can show that the specifications are satisfied by the reduced system with some robustness constraints (explained in the next slide), then we can infer that the original system also satisfy the properties

Robust Testing – Specification Satisfied within the bounded region

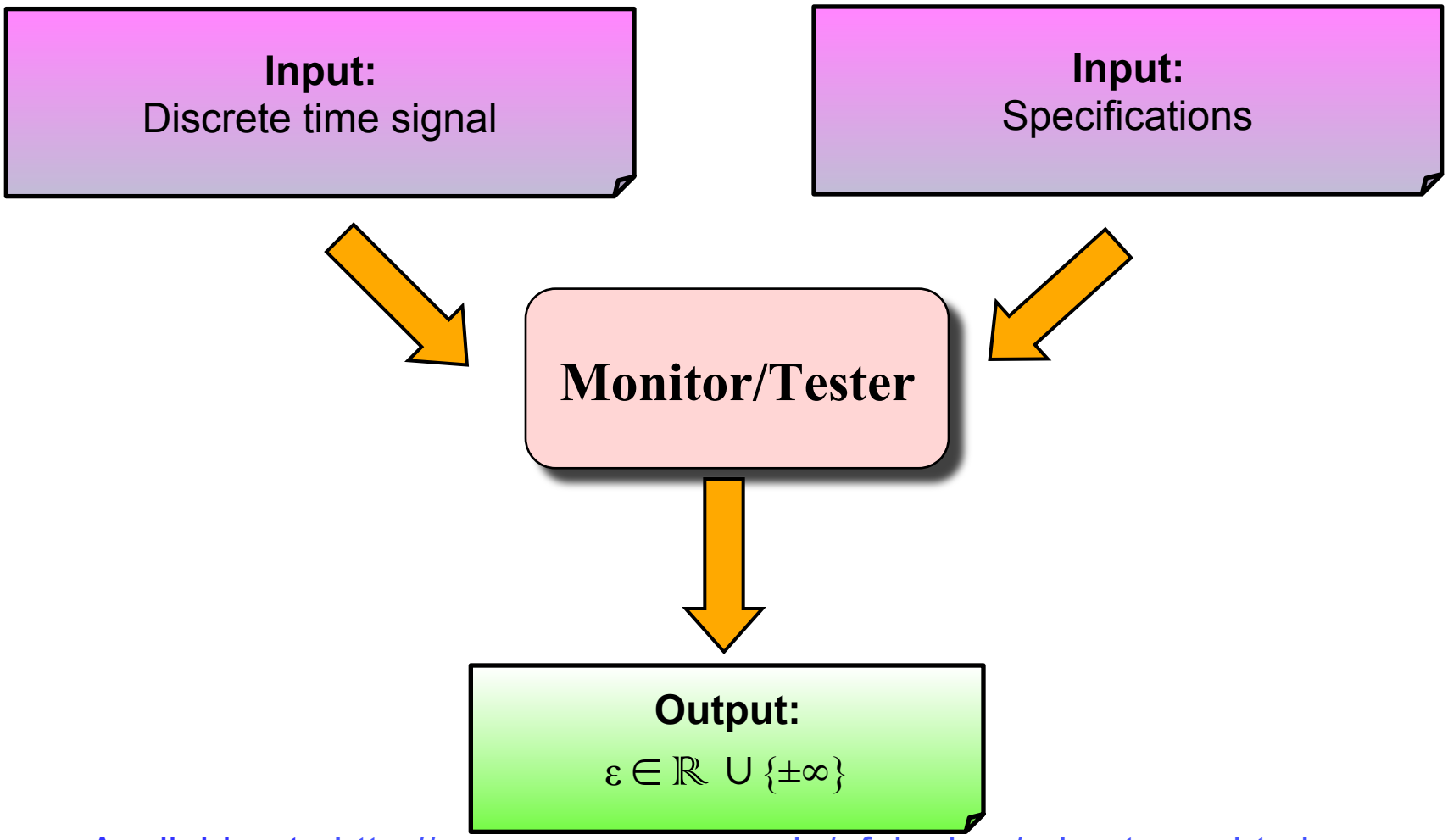


Fainekos, Girard and Pappas, *Temporal logic verification using simulation*, FORMATS 2006

This slide illustrates intuitively how the approach works: Given a system trace of the reduced system and the specifications, we have a software tool (next slide) to calculate a robustness bound eps, meaning that the specification is locally satisfied anywhere within the eps-"tube" (the region between to blue lines) around the given trace.

Next if we can show that the "difference" between the reduced system and the original system is always within the "tube", then it is inferred that the original system traces also satisfy the specifications

Software toolbox : TaLiRo



Available at : http://www.seas.upenn.edu/~fainekos/robustness.html

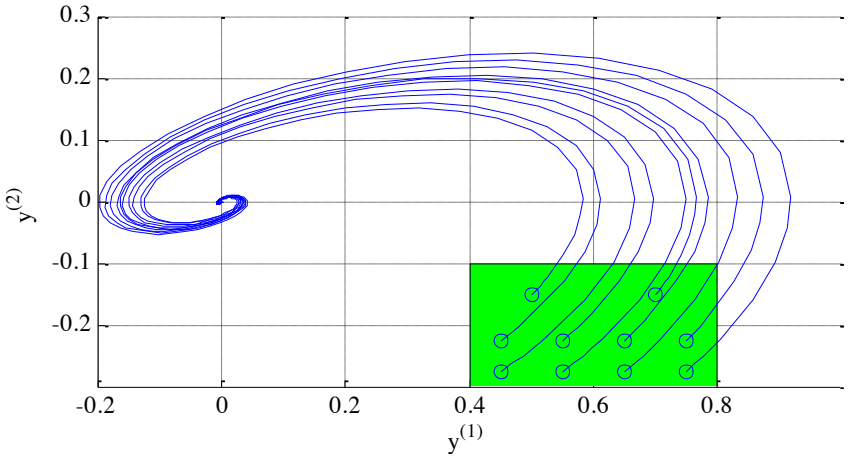
We have a software tool such that given a system trace and specifications, the tool calculate the robustness bound epsilon, meaning that the specifications are satisfied within a epsilon-tube around the given trace

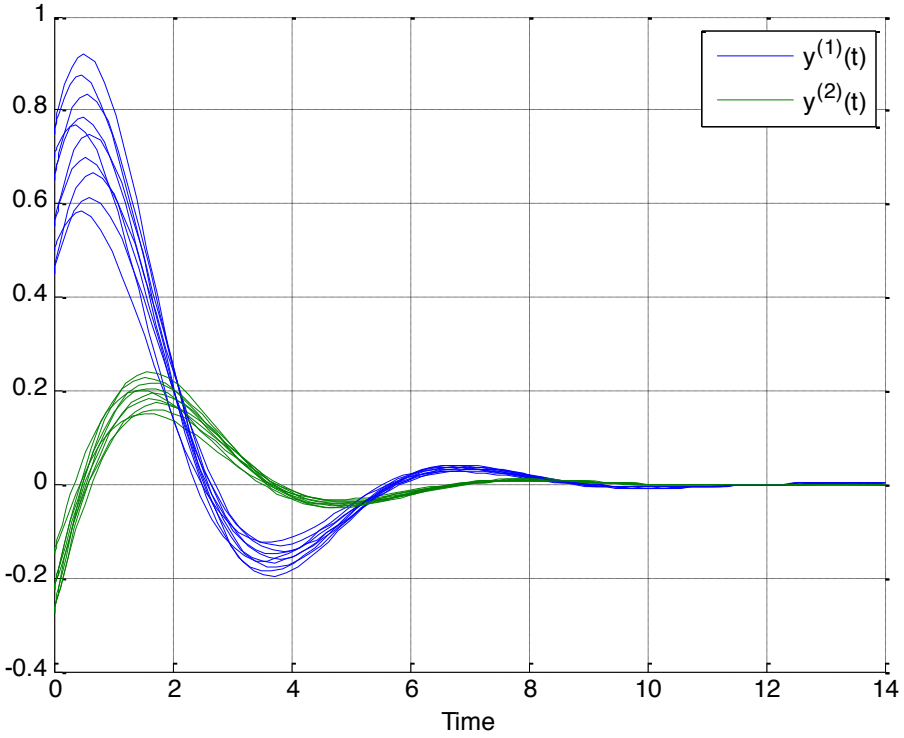
Example : Nonlinear systems

$$\dot{x}_1(t) = 0.05\sin^2(x_2(t))x_1(t) - 2.5x_2(t)$$
$$\dot{x}_2(t) = 0.5x_1(t) - x_2(t)$$
$$X_0 = [0.4, 0.8] \times [-0.3, -0.1]$$

$$\psi_2 = \Box\pi_3 \wedge \Box_{\geq 8}\pi_4$$
$$\mathcal{O}(\pi_3) = \mathbb{R} \times [-0.6, 0.6]$$
$$\mathcal{O}(\pi_4) = [-0.4, 0.4] \times [-0.4, 0.4]$$

$$\dot{x}(t) = \begin{bmatrix} 0.05p(t) & -2.5 \\ 0.5 & -1 \end{bmatrix} x(t)$$
$$p(t) \in P = [0, 1]$$





Here is an example of extending the approach to simple non-linear systems: In the model, the only non-linear term is sin2(x2), rather than dealing with the non-linear system, we transform the system into a linear system with uncertain parameters, by replacing sin2(x2) with a parameter P, which is unknown but bounded (within [0,1]). Next, if we can show that all possible traces of the linear system (with uncertain P) satisfy the specifications, we know the original system also satisfy the same specifications.

Adaptive/Robust Extension

- Possible to extend the robust verification results on linear systems to large non-linear systems
  - Partition parameter space into several regions
    - Example: highly insulin-sensitive, average, and insulin-resistant subjects

Insulin Sensitivity Coefficient: within [3,10]		
Insulin resistant	Average	Insulin sensitive
[3,5]	[5,8]	[8,10]

  - Robustness verification within each region
  - Adaptive exploration to determine which region a newly admitted subject belongs to
    - Exploration phase must satisfy safety properties

Safety Analysis

- Identify platform hazards in the networked control setting
  - Develop mitigation strategies
    - Unlike the closed-loop PCA system, where only overdosing is undesirable, in the BG system, both hypo- and hyper-glycaemia need to be avoided
    - **No trivial fail-safe mode** for closed-loop BG control
  - System-level safe
  - ty verification and validation to show that patient safety is guaranteed in the networked system, even under failure conditions