From Models and Data to Proofs

For Improving Cyberphysical Systems

Sayan Mitra

mitras@Illinois.edu

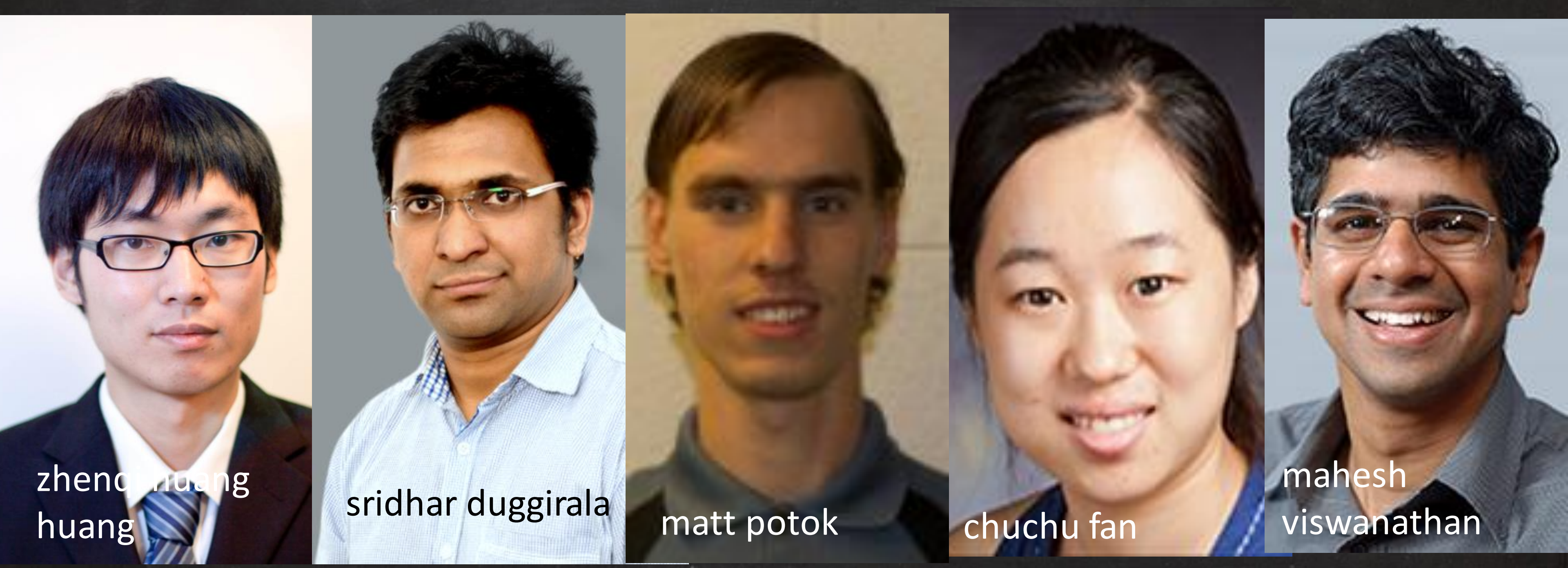
Department of Electrical & Computer Engineering

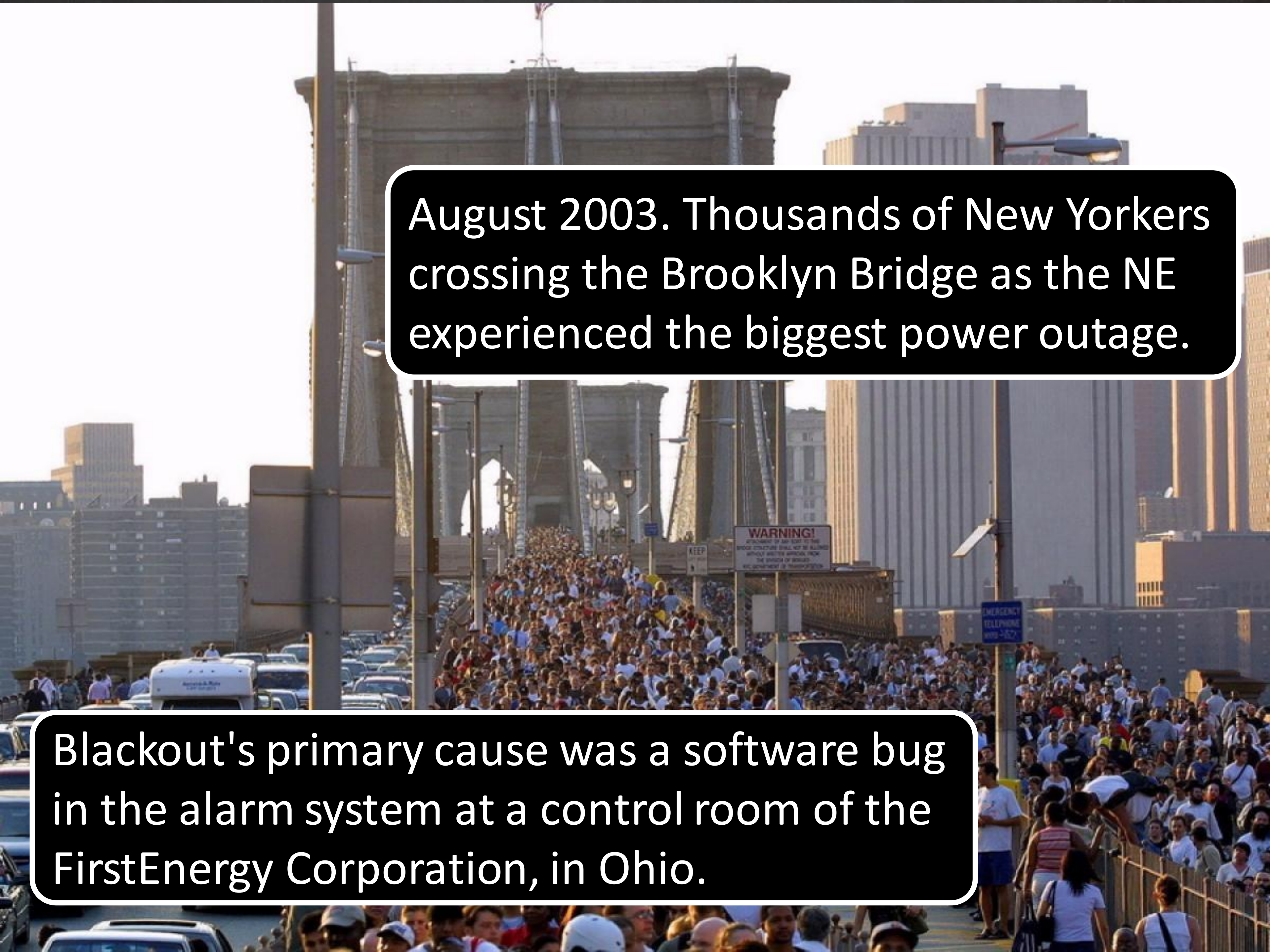
University of Illinois at Urbana Champaign

TENTH CARNEGIE MELLON CONFERENCE ON THE ELECTRICITY INDUSTRY

Testbeds for Smart Grids and Smart Cities, March 31st 2015

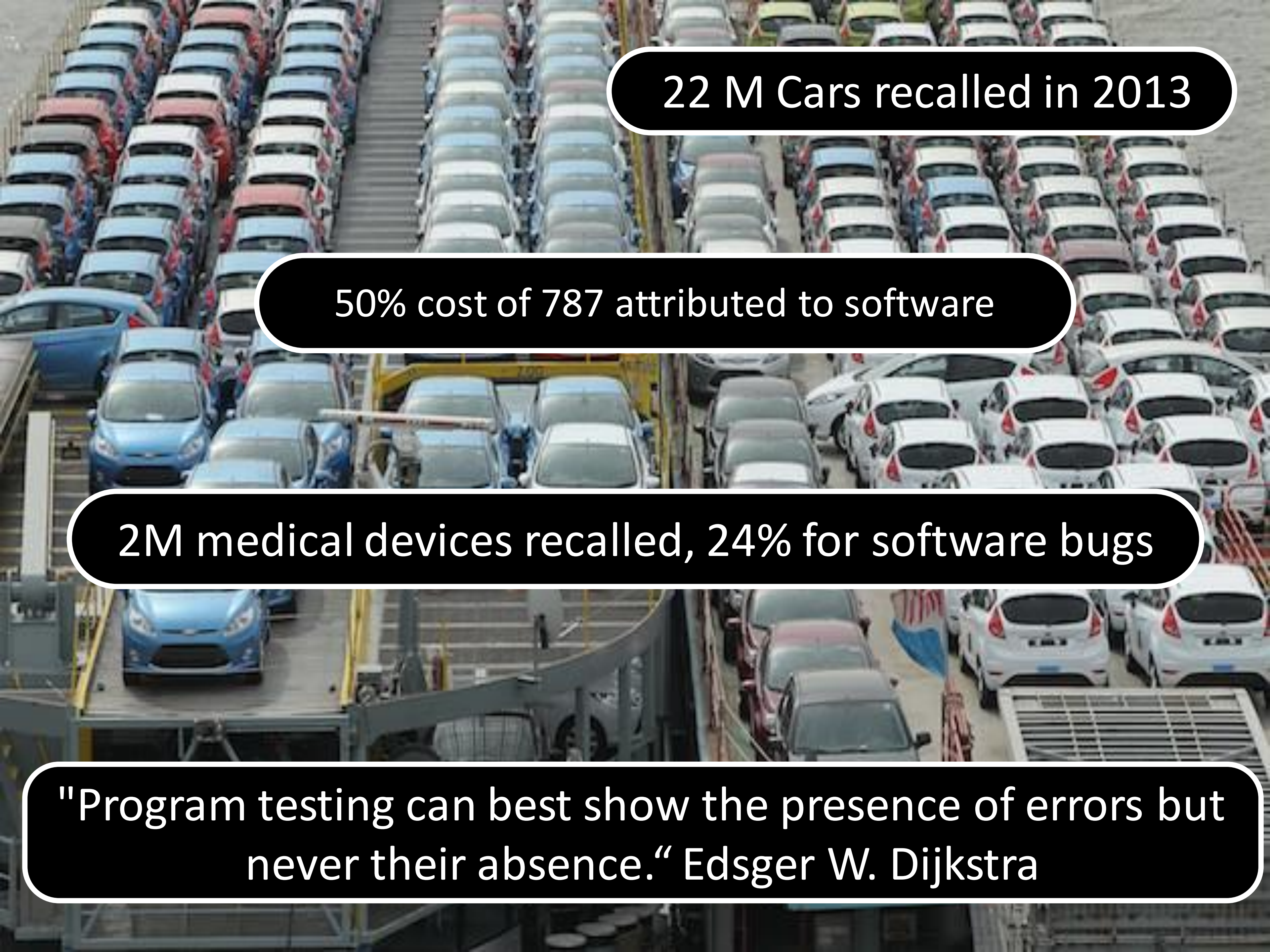
Collaborators





August 2003. Thousands of New Yorkers crossing the Brooklyn Bridge as the NE experienced the biggest power outage.

Blackout's primary cause was a software bug in the alarm system at a control room of the FirstEnergy Corporation, in Ohio.


22 M Cars recalled in 2013
50% cost of 787 attributed to software
2M medical devices recalled, 24% for software bugs
"Program testing can best show the presence of errors but never their absence." Edsger W. Dijkstra

Verification as a Search Problem

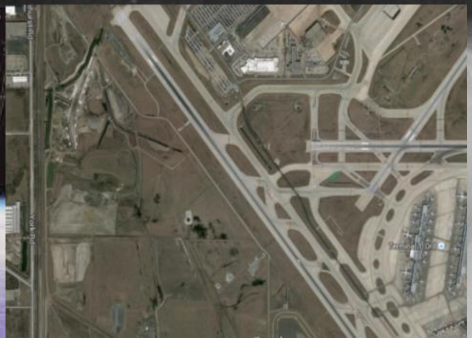
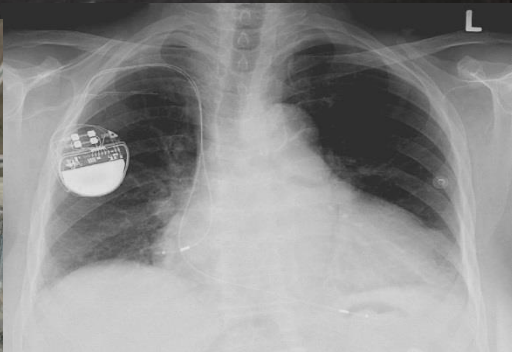
Sound and Complete
always gives correct answer

Given a system model and some requirements, find a behavior of the system that violates those requirements.

Yes (Bug-trace)

There is no such behavior (Safety certificate)

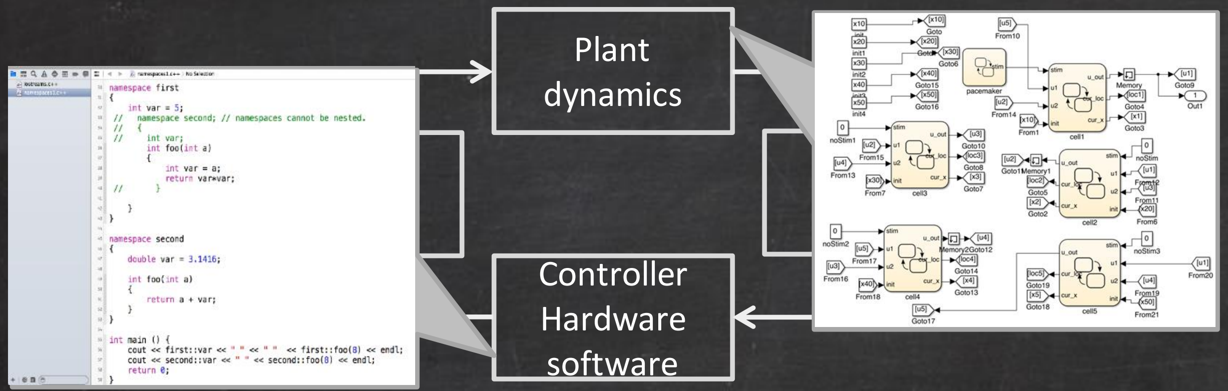
Model + Trace Data → Proof

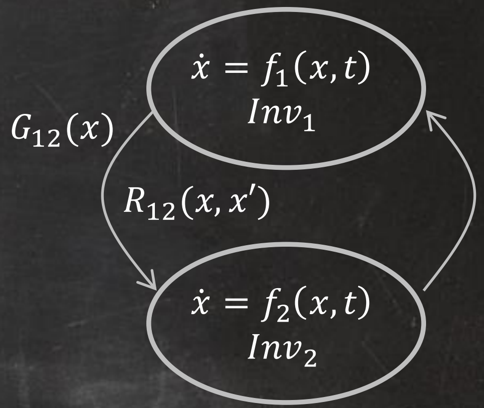




Outline

- Overview of Trace-based Verification
- Three recent case studies on
  - Alerting protocol (NASA/FAA)
  - Powertrain control system (Toyota)
  - Cardiac cells and Pacemaker Network
- Conclusions

Our Tools Handle a Class of Simulink/Stateflow Models





A generic hybrid systems with two modes

Early 90's: Exact unbounded verification: Decidable for $\dot{x} = 1$ [Alur Dill 92]
Undecidable even for $\dot{x}$=1 $\dot{y}$=2 [Henzinger 95]

Late 90'-00': Approximate, bounded, mostly linear: Hamilton-Jacobi-Bellman [Tomlin et al. 02], Polytopes [Henzinger 97], ellipsoids [Kurzhanski] zonotopes [Girard 05], support functions [Frehse 08], CEGAR [Clarke 03]

Today: Scalable, nonlinear: trace-based methods [Mitra 10-13][Donze 07]

Core Idea: Trace-based Verification

Given start S and target T

Compute finite cover of initial set

Execute/simulate from the center x_0 of each cover

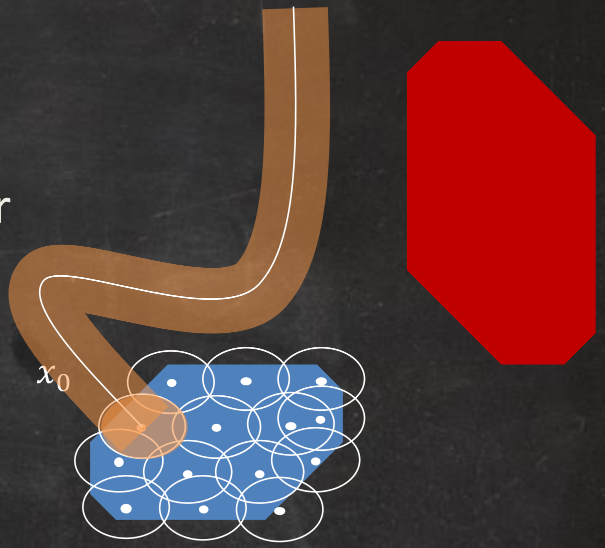
Bloat execution to contain all trajectories from the cover

If contained in T then UNSAFE

Union is an over-approximation of reach set

If Union is disjoint from T then SAFE

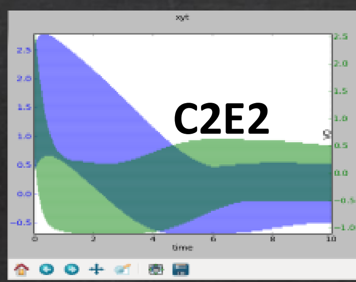
Otherwise, refine cover

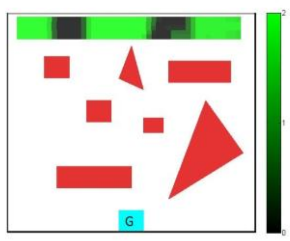


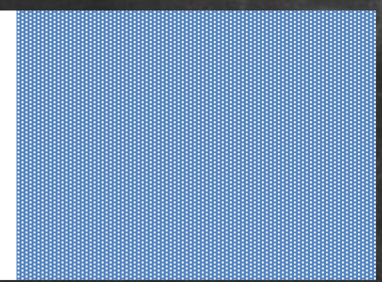
- How much to bloat? Use static analysis of model [EmSoft2013, FM 2014]
- How to handle mode switches? May-must analysis [TACAS 2015]
- How to handle large models? Compositional analysis [HSCC 2014, CAV 2014]

Discrepancy: a Layer Between Algorithms for (Verification | Synthesis | Monitoring) and (Models| Testbeds | Simulators)

verification ∘ synthesis ∘ monitoring

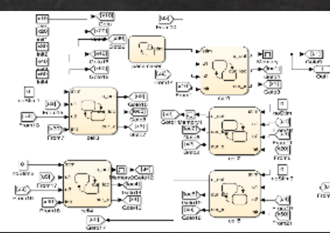






Discrepancy β

$$\dot{x} = f(x, u)$$
$$x \in Inv$$
$$y = g(x, u)$$

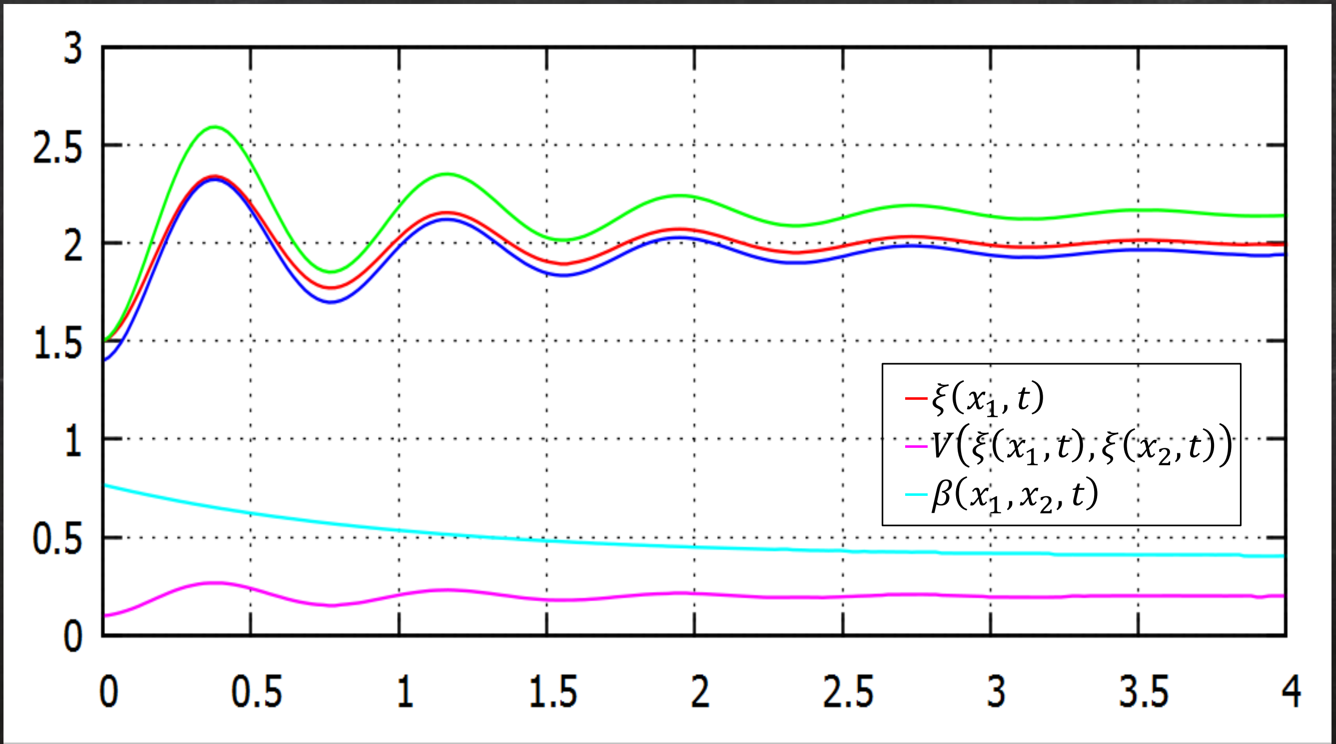


math model ∘ code ∘ hardware

A model characteristic extracted using static analysis: Discrepancy

Definition. $\beta: \mathbb{R}^{2n} \times \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}^{\geq 0}$ defines a discrepancy of the system if for any two states x_1 and $x_2 \in X$, For any t,

1. $|\xi(x_1, t) - \xi(x_2, t)| \leq \beta(x_1, x_2, t)$ and
2. $\beta \rightarrow 0$ as $x_1 \rightarrow x_2$



Algorithms are Sound & Relatively Complete

Theorem. (Soundness). If Algorithm returns safe or unsafe, then A is safe or unsafe.

Definition Given any HA $A = \langle V, Loc, A, D, T \rangle$, an **$\epsilon$-perturbation** of A is a new HA A' that is identical except, $\Theta' = B_\epsilon(\Theta)$, $\forall \ \ell \in Loc, Inv' = B_\epsilon(Inv)$ (b) a ∈ A, $Guard_a = B_\epsilon(Guard_a)$.

A is **robustly safe** iff $\exists \epsilon > 0$, such that A' is safe for U_ϵ upto time bound T, and transition bound N. Robustly unsafe iff $\exists \ \epsilon < 0$ such that A' is safe for U_ϵ.

Theorem. (Relative Completeness) Algorithm always terminates whenever the A is either robustly safe or robustly unsafe.

Outline

- Overview of Trace-based Verification
- Three recent case studies on
  - Alerting protocol (NASA/FAA)
  - Powertrain control system (Toyota)
  - Cardiac cells and Pacemaker Network
- Conclusions

SAPA-ALAS Parallel Landing Protocol

Air traffic is going to double in the next 20-25 years

Strong need to improve airport throughput

Cost of new runways: ~ $USD 15B+

Duggirala, Wang, Mitra, Munoz, Viswanathan FM 2014

SAPA-ALAS Parallel Landing Protocol

Air traffic is going to double in the next 20-25 years

Strong need to improve airport throughput

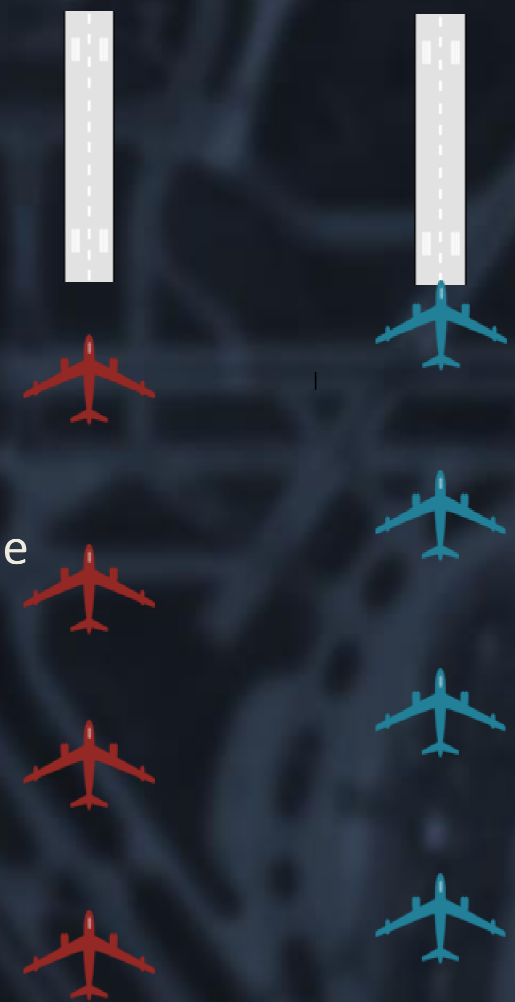
Cost of new runways: ~ $USD 15B+

Alternatively, pack more planes in shorter space & time

There are physical limits, e.g., wake vortices

But there is also human (co-pilot) in the loop

Solution: software!



Duggirala, Wang, Mitra, Munoz, Viswanathan FM 2014

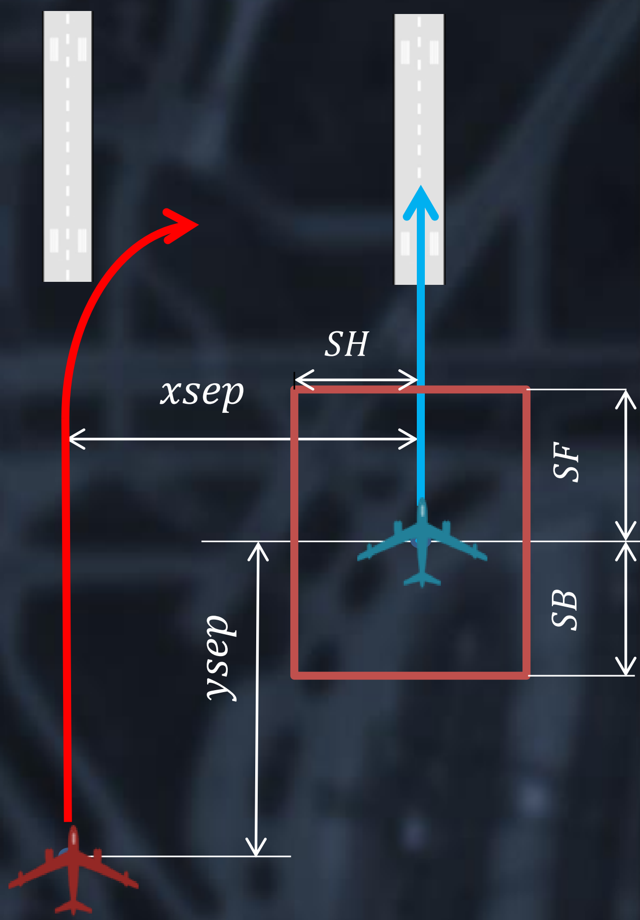
SAPA-ALAS Parallel Landing Protocol

Ownship and *Intruder* approaching parallel runways with small separation

ALAS (at ownship) NASA's protocol supposed to raise an alarm if within T time units the *Intruder* can violate safe separation

Can we trust ALAS? $Alert \prec_b Unsafe$?

Uncertainty: $xsep \in [.11, .12]$ Nm $ysep \in [.1, .21]$ Nm, $\phi \in [30^o, 45^o]$ vy_o= 136 Nmph, vy_i = 155 Nmph



Duggirala, Wang, Mitra, Munoz, Viswanathan FM 2014

C2E2 Verifies Alerting Protocol in Minutes

Our verification tool computes increasingly more precise over-approximations of the reachable states of the system and automatically proves $Alert \prec Unsafe$ properties for different scenarios in reasonable time

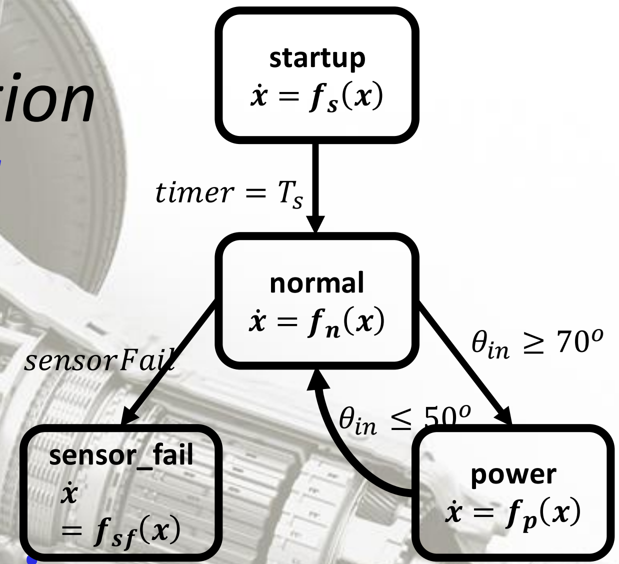
Shows that false alarms are possible

Finds scenarios where alarm may be missed

Scenario	Alert $\preccurlyeq_4$ Unsafe	Running time (mins:sec)	Alert $\preccurlyeq_?$ Unsafe
6	False	3:27	2.16
7	True	1:13	–
8	True	2:21	–
6.1	False	7:18	1.54
7.1	True	2:34	–
8.1	True	4:55	–
9	False	2:18	1.8
10	False	3:04	2.4
9.1	False	4:30	1.8
10.1	False	6:11	2.4

2. Powertrain Control System

Simulink model of a powertrain control system provided by ***Toyota*** *as a verification challenge.* ***Highly nonlinear polynomial differential*** *equations;* ***discrete mode switches***



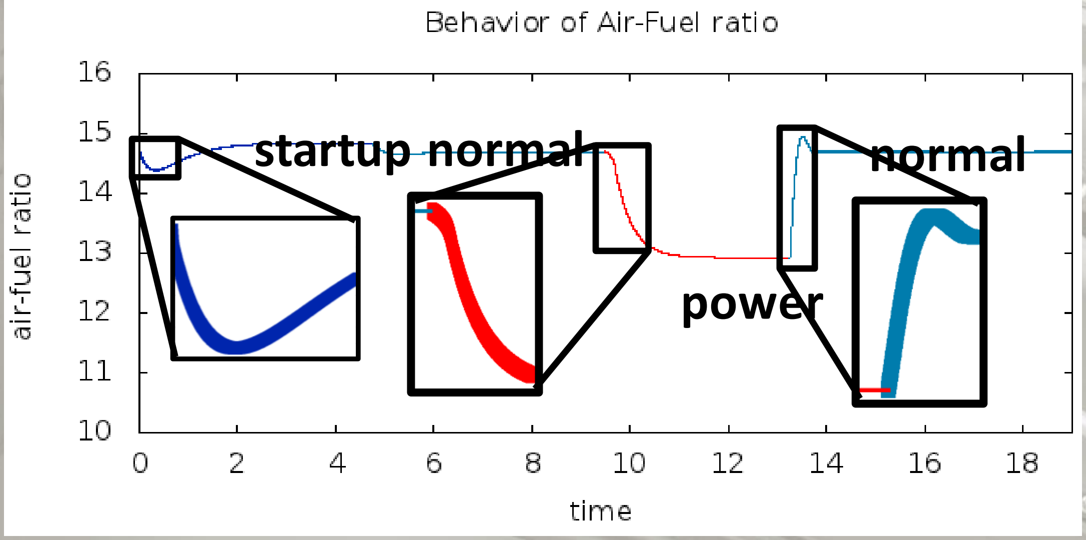
First to verify properties, e.g., that the **air-fuel ratio** remains within a given range for a set of driver behaviors

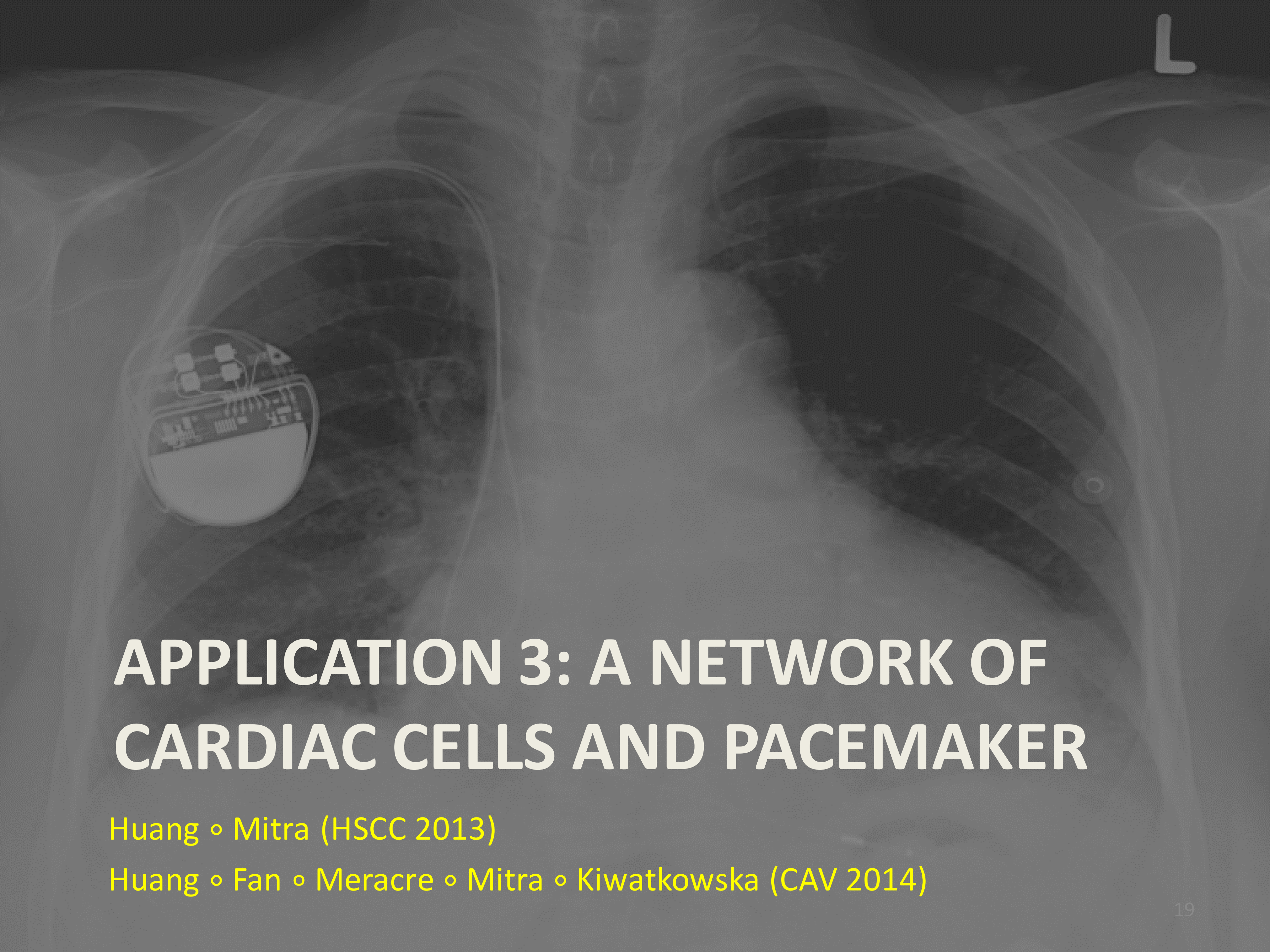
Discrepancy function β computed automatically using the local algorithm

2. Powertrain Control System

Simulink model of a powertrain control system provided by Toyota as a verification challenge. Highly nonlinear polynomial differential equations; discrete mode switches

We converted the model to Stateflow that can be processed by our tool; rest of the analysis was **completely automatic**. The **whole exercise took less than a month**





APPLICATION 3: A NETWORK OF CARDIAC CELLS AND PACEMAKER

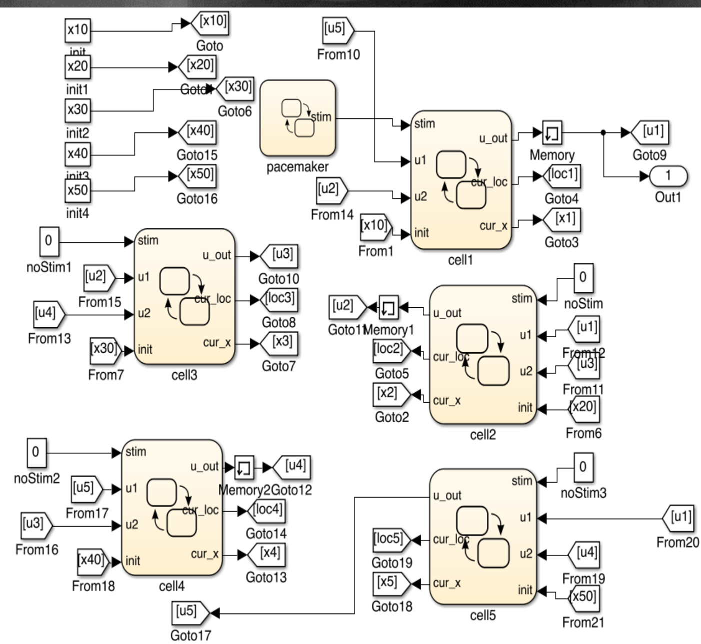
Huang ∘ Mitra (HSCC 2013)

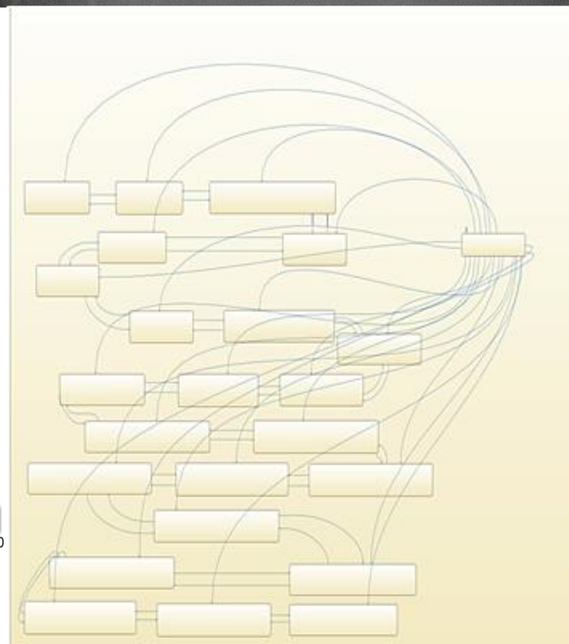
Huang ∘ Fan ∘ Meracre ∘ Mitra ∘ Kiwatkowska (CAV 2014)

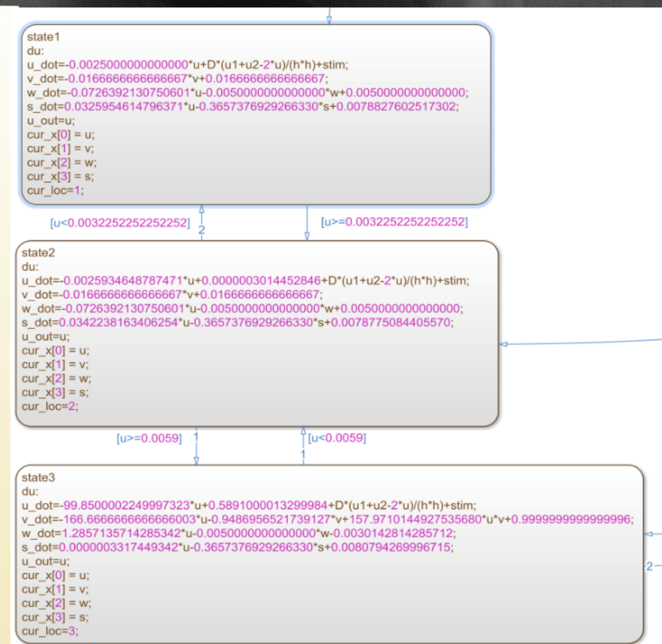
3. Pacemaker + Cardiac Network

Simulink model of a **network of cardiac cells** and a pacemaker; nonlinear differential equations; **30+ continuous variables**; many interacting components; uncertainty in timing and initial voltages

Key property: voltage range action potentials remain in specific interval and has periodicity



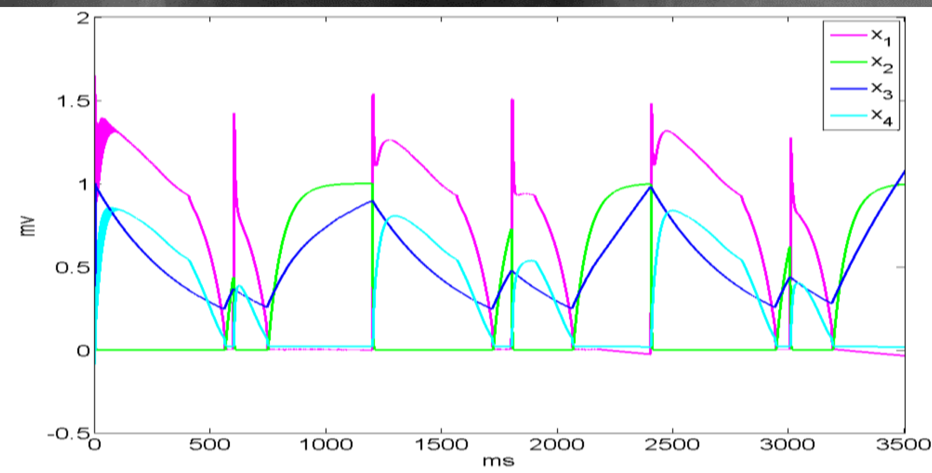




3. Pacemaker + Cardiac Network

Our tool **first** to verify properties of this model (running times shown below)

Compositional or modular analysis for computing the discrepancy



Variables	Thresh	Sims	Run time (s)	Property
15	2	16	104.8	TRUE
15	1.65	16	103.8	TRUE
25	2	3	208	TRUE
25	1.65	5	281.6	TRUE
25	1.5	NA	63.4	FALSE
40	2	3	240.1	TRUE
40	1.65	73	2376.5	TRUE

Conclusion

We have developed new algorithms and tools for analyzing complex, nonlinear hybrid models of control systems and software;

- **Use Traces + Discrepancy → algorithms**
- **Sound (guarantees coverage): Gives proof of correctness or finds a bug**
- **Relatively complete: Always gives an answer[1]**
- **Effective: Appears to work for large & interesting examples[2]**

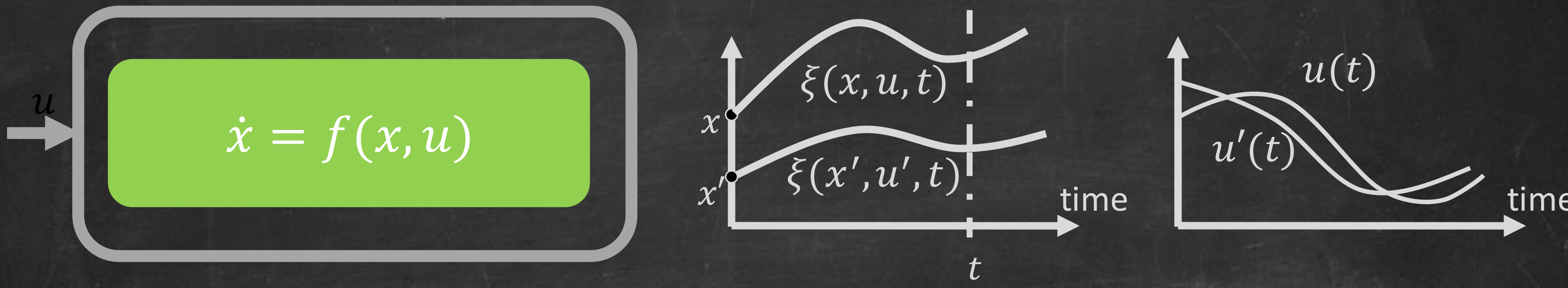
Can this technology be used in design of Smart Grids

- **Generating tests**
- **Finding parameters that satisfy properties**
- **Online monitoring**
- **Designing controllers**

1: Unless the system is fragile with respect to the property in question

2: Exploiting parallelism will make it scale to even larger models

Input-to-State (IS) Discrepancy

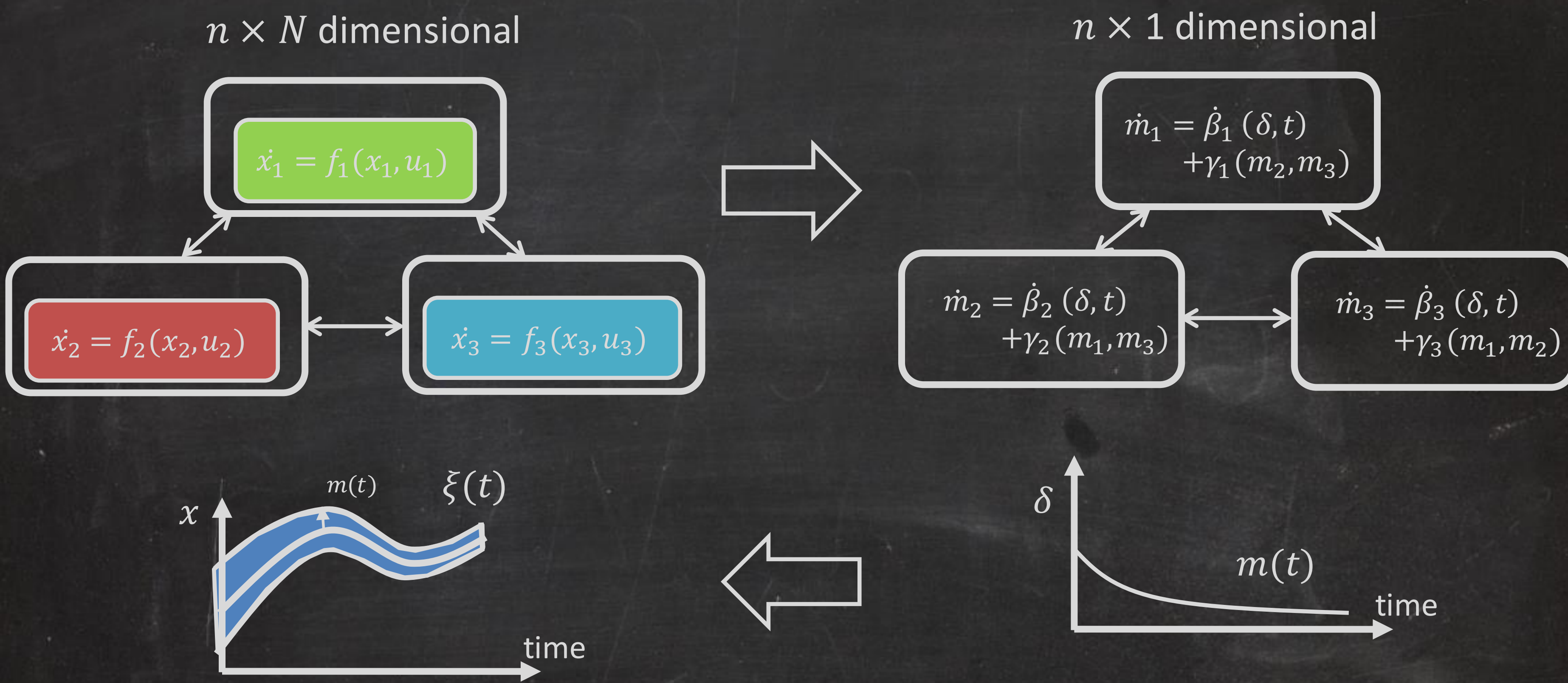


Definition. **IS discrepancy** is defined by β and γ such that for any initial states x, x' and any inputs u, u',

$$|\xi(x,u,t) - \xi(x',u',t)| \leq \beta(x,x',t) + \int_0^t \gamma(|u(s) - u'(s)|)ds$$

$\beta \rightarrow 0$ as $x \rightarrow x'$, and $\gamma \rightarrow 0$ as $u \rightarrow u'$

Bloating with Reduced Model



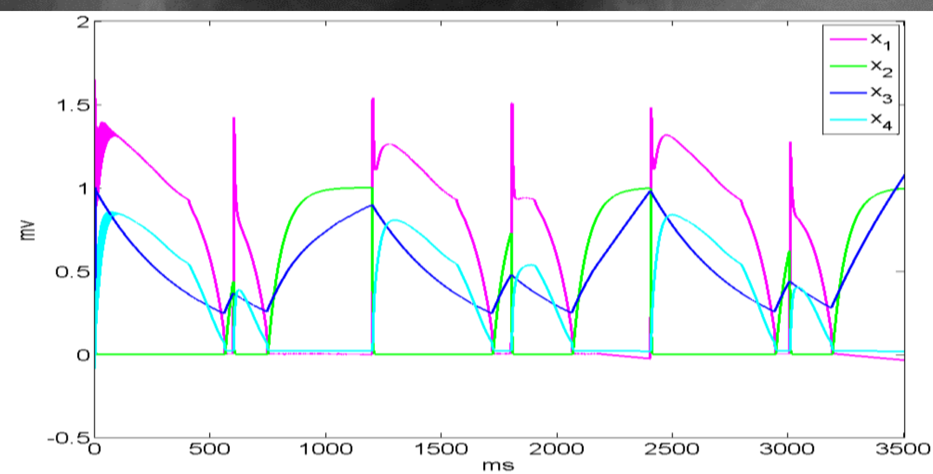
The bloated tube contains all trajectories start from the δ-ball of x.

The over-approximation can be computed arbitrarily precise.

3. Pacemaker + Cardiac Network

Our tool **first** to verify properties of this model (running times shown below)

Compositional or modular analysis for computing the discrepancy



Variables	Thresh	Sims	Run time (s)	Property
15	2	16	104.8	TRUE
15	1.65	16	103.8	TRUE
25	2	3	208	TRUE
25	1.65	5	281.6	TRUE
25	1.5	NA	63.4	FALSE
40	2	3	240.1	TRUE
40	1.65	73	2376.5	TRUE