Teaching theoretical foundations of Cyber-Physical Systems

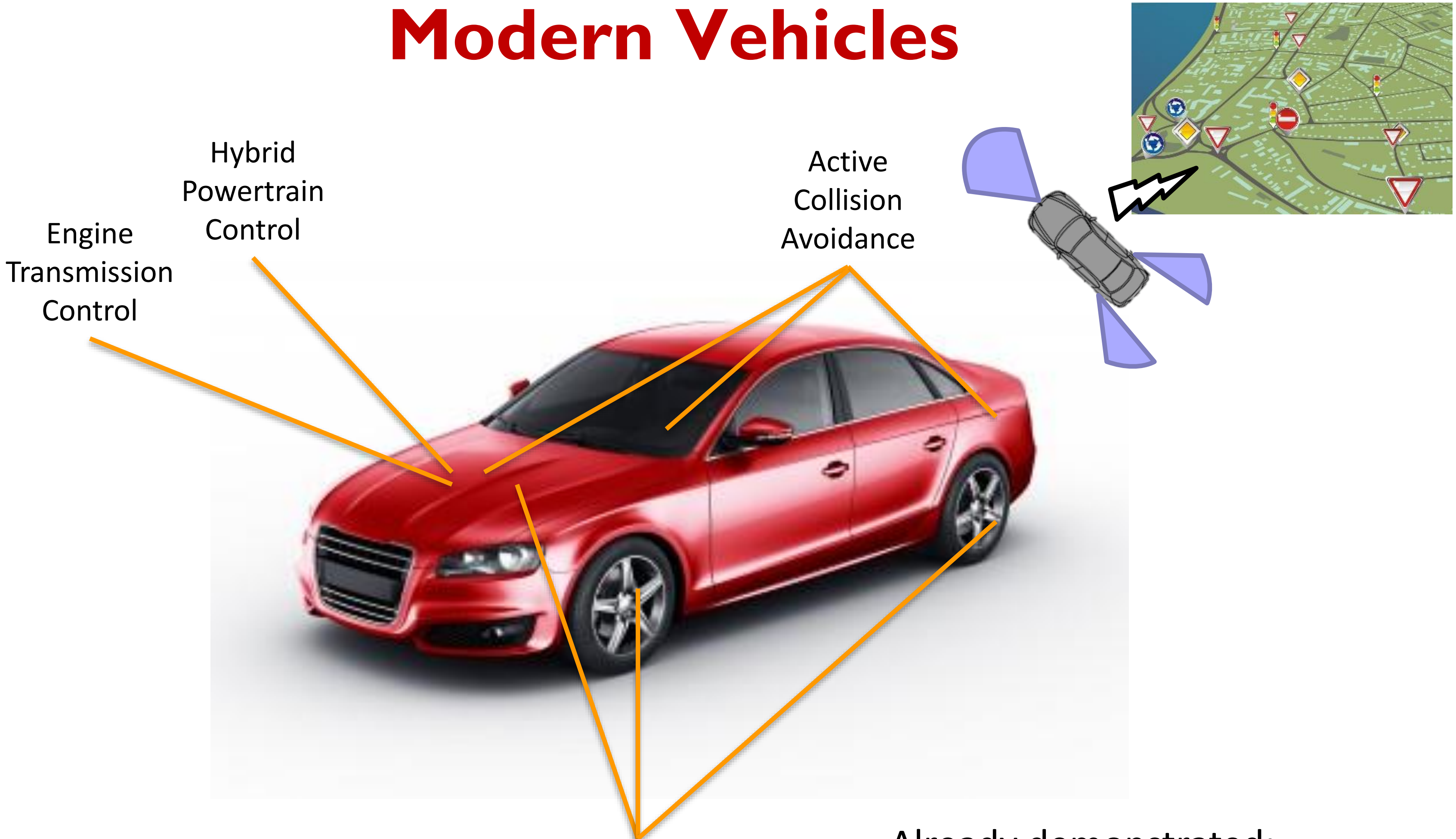
Georgios Fainekos

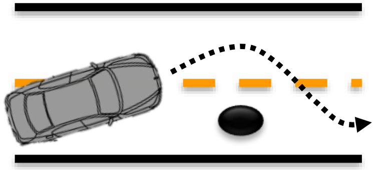
July 2017 @ CPS Ed 2017

✉ fainekos at asu edu

💻 http://www.public.asu.edu/~gfaineko

Modern Vehicles

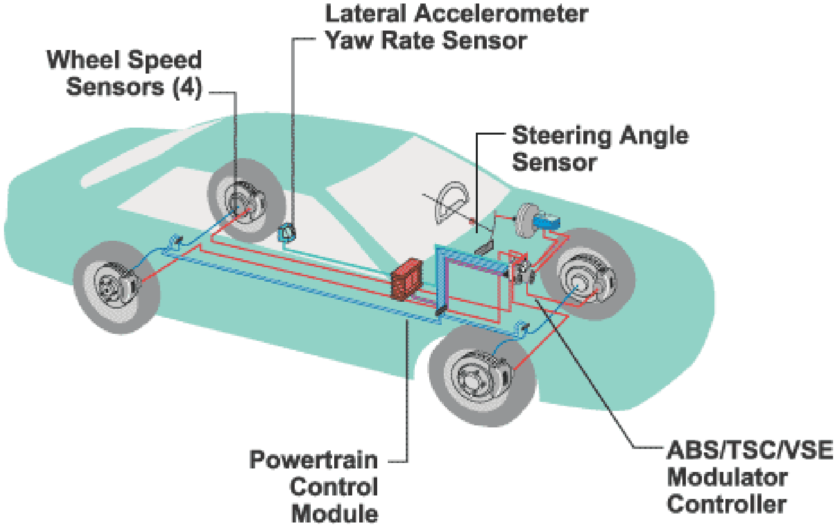

Hybrid Powertrain Control
Engine Transmission Control
Active Collision Avoidance
Electronic Stability Control

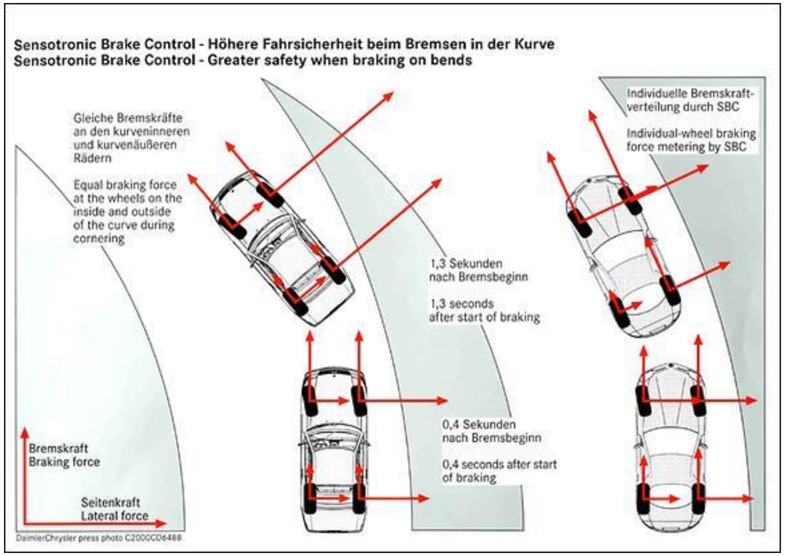


Already demonstrated:

- Lane following & Active cruise control
- Fully autonomous driving
- …

Embedded in : Automotive Systems




Sensotronic Brake Control - Höhere Fahrsicherheit beim Bremsen in der Kurve
Sensotronic Brake Control - Greater safety when braking on bends
Gleiche Bremskräfte an den kurveninneren und kurvenäußeren Rädern
Equal braking force at the wheels on the inside and outside of the curve during cornering
Individuelle Bremskraftverteilung durch SBC
Individual-wheel braking force metering by SBC
1,3 Sekunden nach Bremsbeginn
1,3 seconds after start of braking
0,4 Sekunden nach Bremsbeginn
0,4 seconds after start of braking
Bremskraft
Braking force
Seitenkraft
Lateral force
DaimlerChrysler press photo C2000C06488

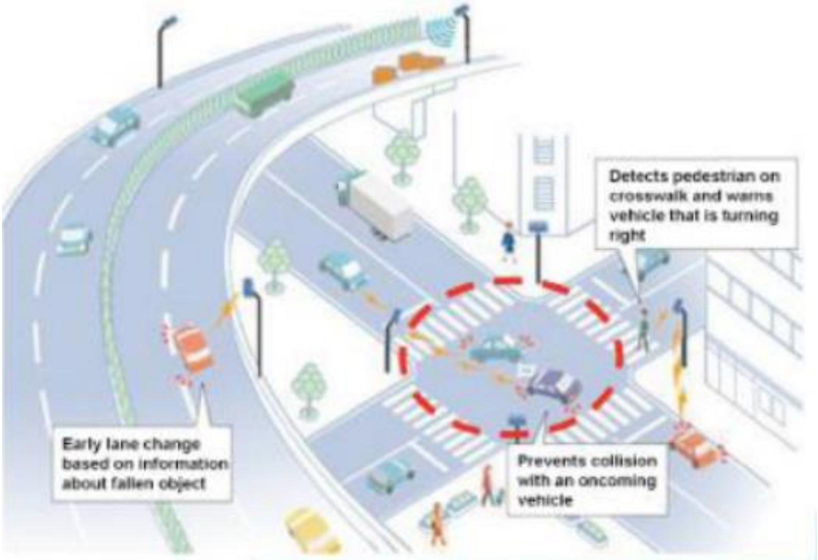
- Longitudinal dynamics : ABS (anti-lock brake system) and ASC (automatic stability control)
- Lateral dynamics : EDRC (engine drag reduction control) and CBC (corner braking control)
- DSC (dynamic stability control) is using all the above
- Autonomous parking
- Lane following and adaptive cruise control

“Soon” near you:

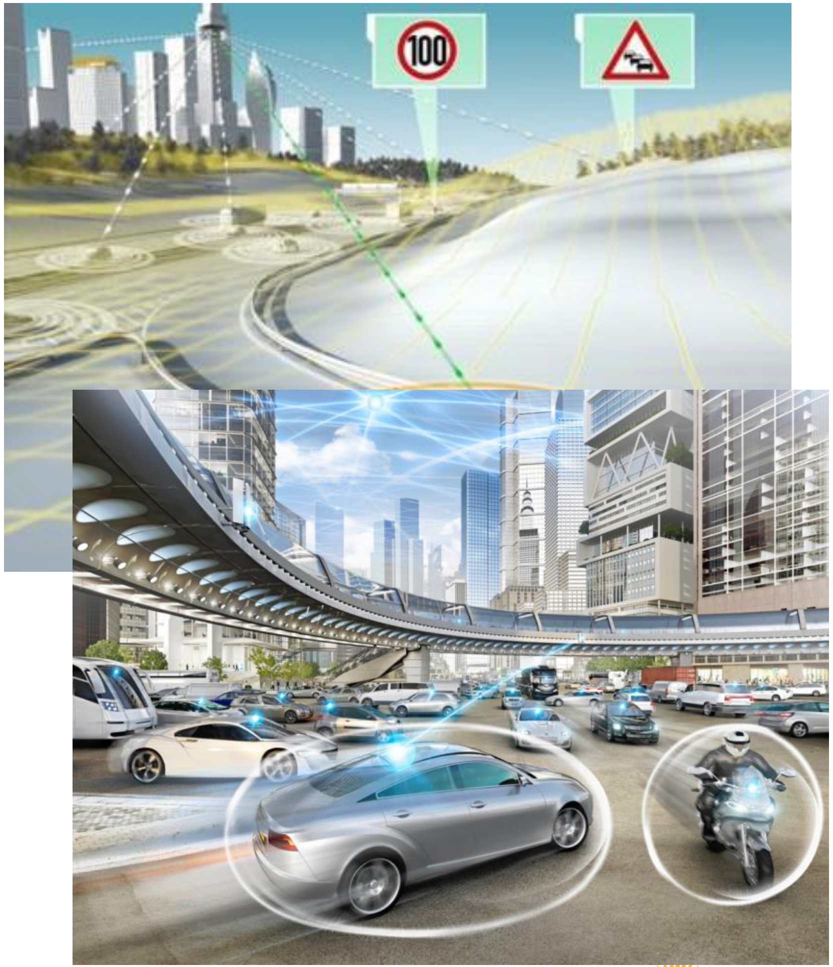
- Fully autonomous vehicles

Smart Road Infrastructure: Closing the loop at a higher level

[Image by Ken Butts, Toyota]


Detects pedestrian on crosswalk and warns vehicle that is turning right
Early lane change based on information about fallen object
Prevents collision with an oncoming vehicle

[Continental Cooperation: The Cloud as sensor]



Trust? : Sampling of automotive recalls (~2011-12) due to software errors …

- "A software error may prevent the transmission from downshifting, such as shifting from 5th to 4t … ry of the problem. "This … ne stall, increasing the risk of a crash."

No downshifting from 5th to 4th

- … the software that "allows the ECU to establish a 'handshake' with the engine is in error. The EC … gine is found to be out of toler … pens, the ECU triggers a fault … tion outside its prescribed tolerances, a rough idle or stalling situation ensues."

Rough idling or stalling due to complicated adaptive ECU

- … to u … tain circums … tor to rotate in the direction opposite to that selected by the transmission.

Electric motor to rotate in the direction opposite to that selected by the transmission

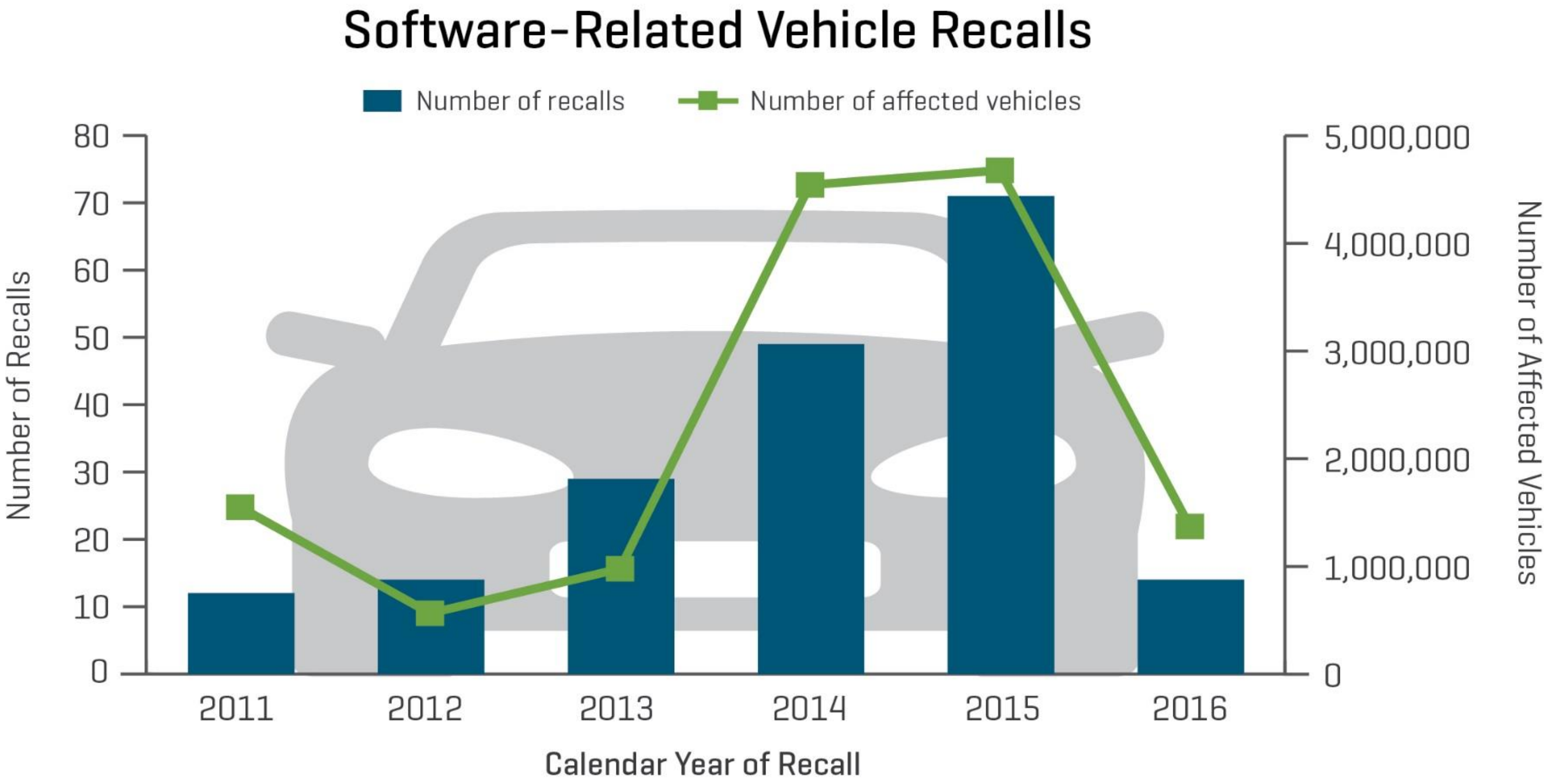
- If the fault occ … e ignition while driving - which … so disables power steering. Braking or pressing the cancel button will not work.

Cruise control does not disengage unless turning off the ignition

- …

Many more …

How serious this problem is?



Software-Related Vehicle Recalls

Source: J.D. Power SafetyIQ and NHTSA's safecar.gov

The same holds for the medical device industry!

http://www.jdpower.com/press-releases/jd-power-safetyiq-may-2016

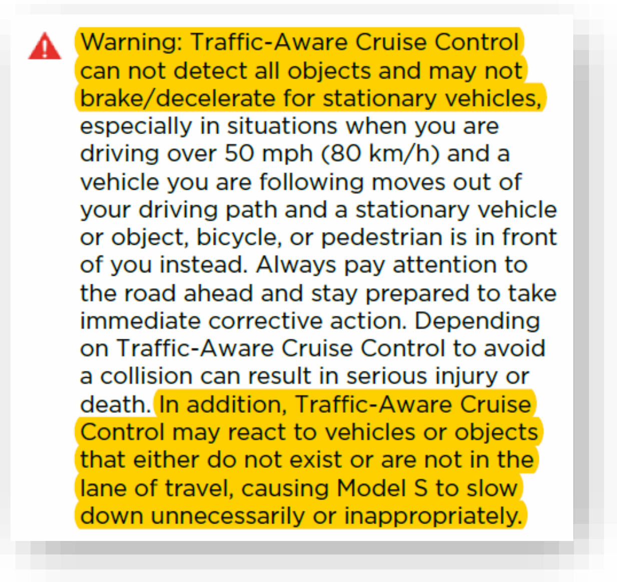
Is it always a software error?!?

https://www.youtube.com/watch?v=qQkx-4pFjus

A Tesla somewhere in Switzerland

Tesla cars: Clearly a marvel of modern engineering!

From the Tesla Model X Owner's manual (Not a bug!):



⚠ Warning: Traffic-Aware Cruise Control can not detect all objects and may not brake/decelerate for stationary vehicles, especially in situations when you are driving over 50 mph (80 km/h) and a vehicle you are following moves out of your driving path and a stationary vehicle or object, bicycle, or pedestrian is in front of you instead. Always pay attention to the road ahead and stay prepared to take immediate corrective action. Depending on Traffic-Aware Cruise Control to avoid a collision can result in serious injury or death. In addition, Traffic-Aware Cruise Control may react to vehicles or objects that either do not exist or are not in the lane of travel, causing Model S to slow down unnecessarily or inappropriately.

- Why the engineers cannot guarantee correct operation under all conditions?
- Can you prove / formally verify correctness?
- How do you even test such a system?

Are these just programming errors?!?

Could these be logical / design errors?!?

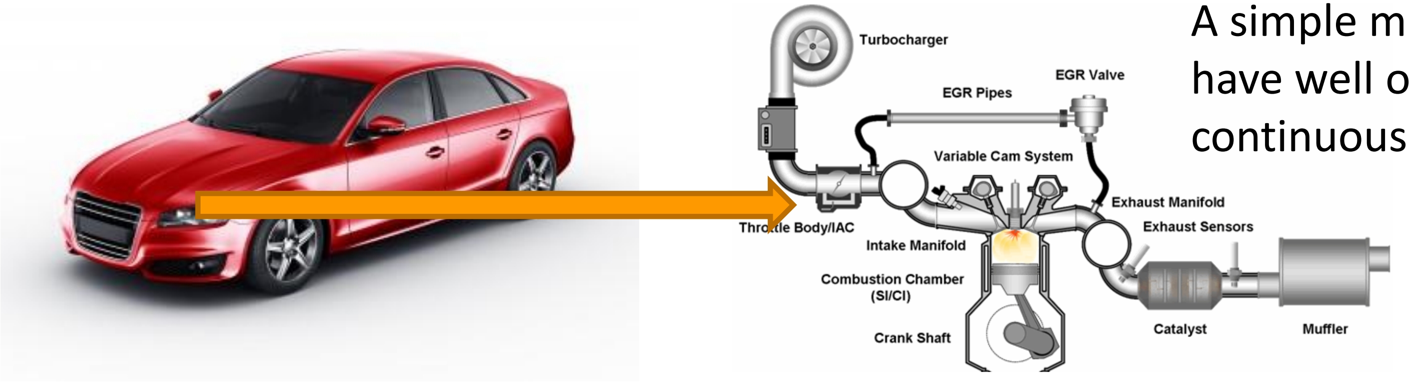
Can we even answer these questions efficiently and effectively?

WHY IS THE PROBLEM CHALLENGING?

Control design for powertrain

Vehicle dynamics & Environment

Engine dynamics



A simple model could have well over 60 continuous state variables.

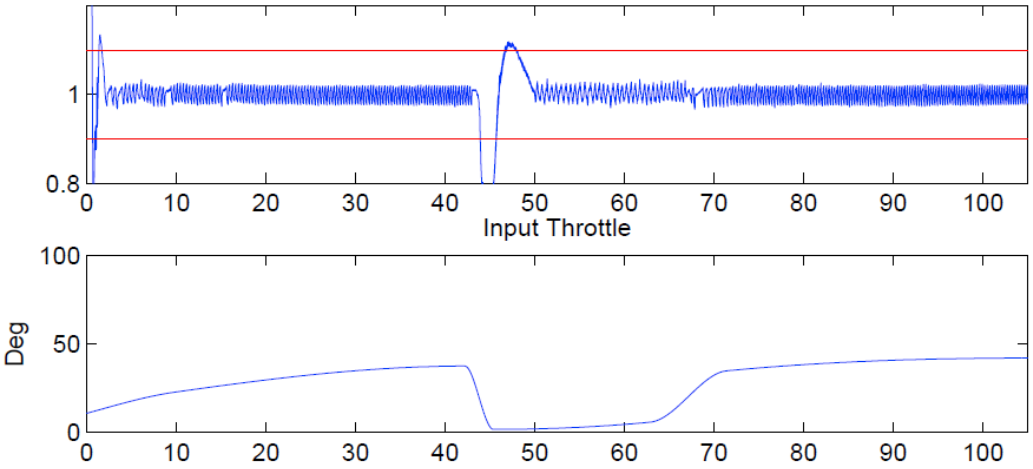
[Image: SimuQuest®]

Requirement: Whenever the normalized air-to-fuel ratio is outside [0.9,1.1], it will settle back inside the range within 1 sec, and stay there for at least 1 sec.

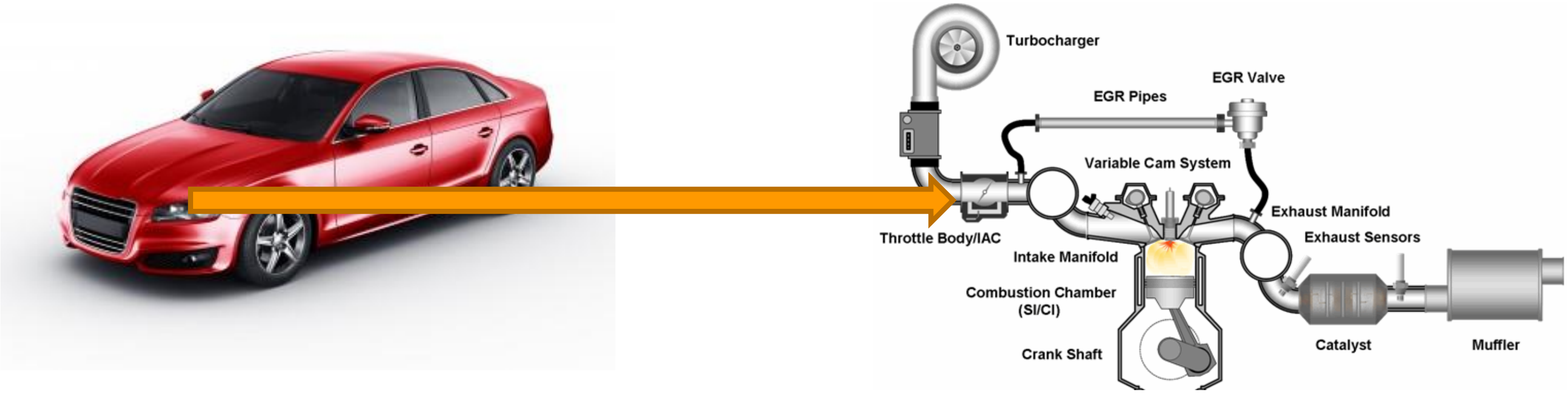
Controller design??

Challenges:

1. Noisy environment & high dim nonlinear dynamics
2. Hard real-time requirements <10ms



Engine models: Complex!



[Image: SimuQuest®]

Enginuity^{TM} Modeling Approach

Orifice Flow

Isentropic Flow Model

$$\dot{m}_1 = A \frac{p}{\sqrt{RT}} \psi$$

$$\psi = \sqrt{\dots [\max(\dots) - \max(\dots)]}$$

⋮

Intake and Exhaust Plenum

Mass Conservation

Energy Conservation

$$\dot{m}_2 = \begin{cases} > 0 & if\ p_1 > p_2 \\ = 0 & if\ p_1 = p_2 \\ < 0 & if\ p_1 < p_2 \end{cases}$$

Combustion Chamber

Energy Conservation

Heat Transfer

Heat Release

Ignition Delay

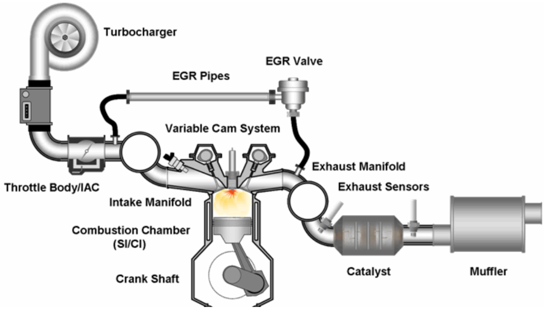
Fuel Injection Dynamics

⋮

…

Develop controllers and generate code

Engine dynamics



[Image: SimuQuest®]

Simplify model:

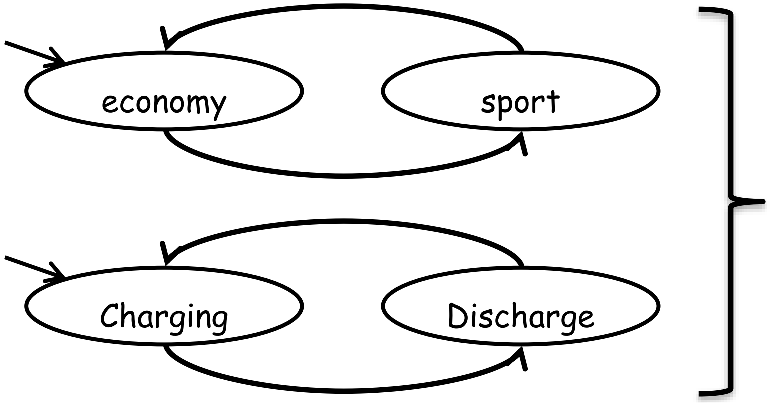
$$\dot{x} = Ax + Bu$$

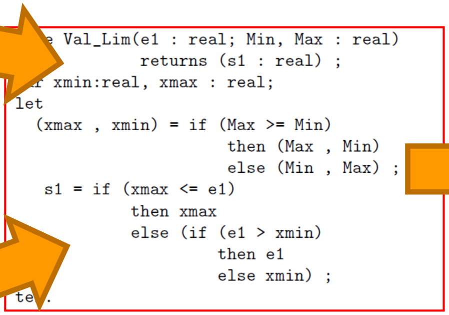
or

$$\dot{x} = f(x,u), \#(x) \ll 60$$

Alternative path:
PID tuning

Design control laws
e.g. idle speed control



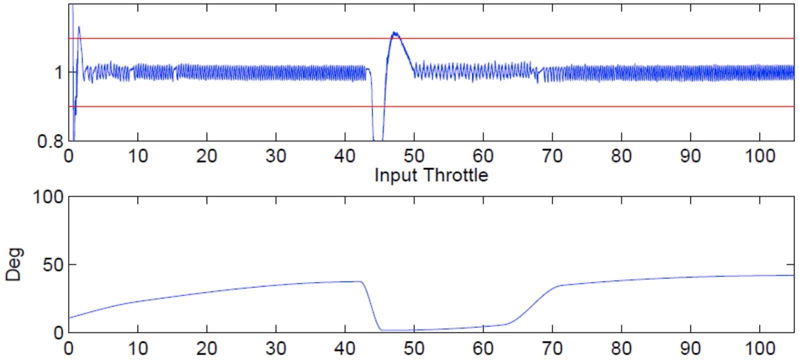


```
 e Val_Lim(e1 : real; Min, Max : real)
              returns (s1 : real) ;
 r xmin:real, xmax : real;
let
  (xmax , xmin) = if (Max >= Min)
                    then (Max , Min)
                    else (Min , Max) ;
   s1 = if (xmax <= e1)
            then xmax
            else (if (e1 > xmin)
                    then e1
                    else xmin) ;
te .
```

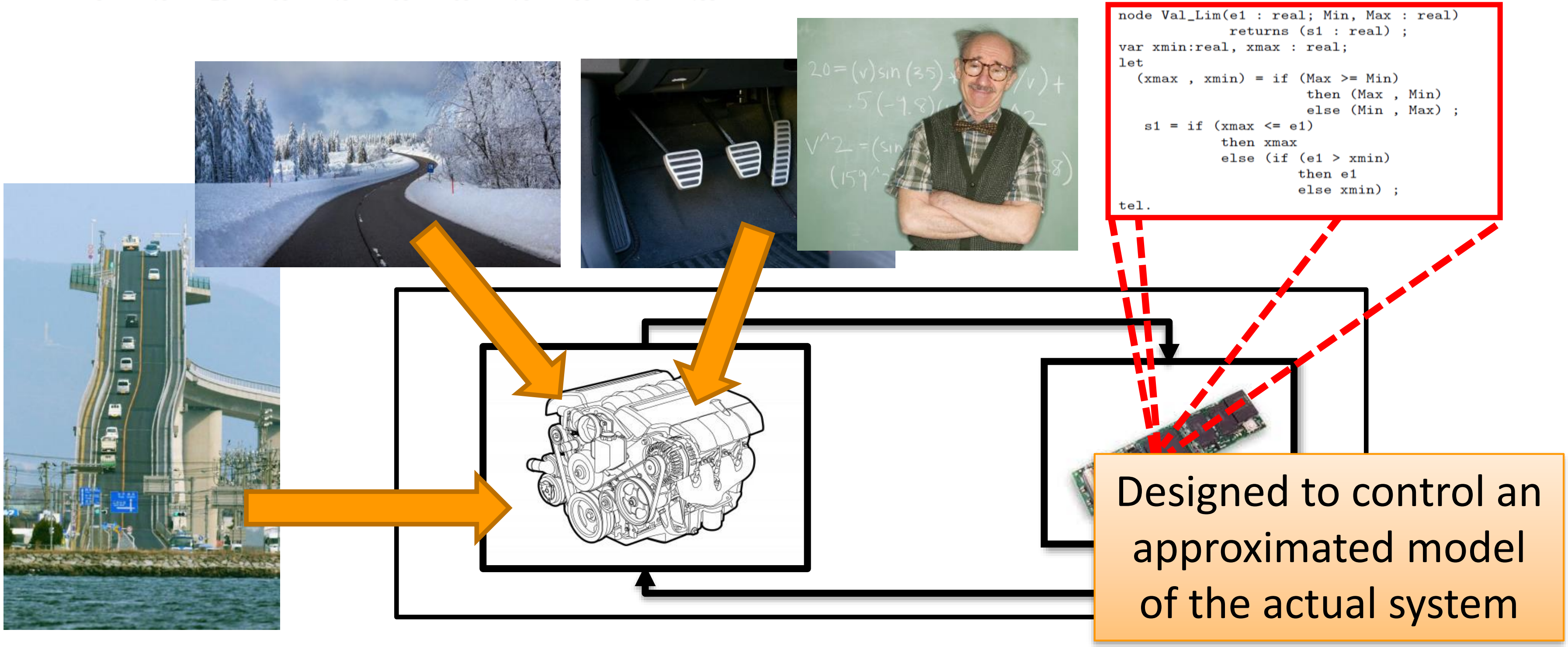
A mix of autocode and manual coding

Real-time execution guarantees

Control design for powertrain



How can we guarantee that the embedded control system will satisfy the design requirements?

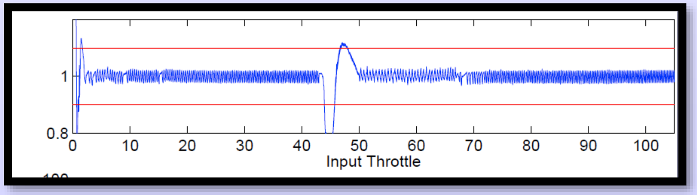


```
node Val_Lim(e1 : real; Min, Max : real)
          returns (s1 : real) ;
var xmin:real, xmax : real;
let
  (xmax , xmin) = if (Max >= Min)
                  then (Max , Min)
                  else (Min , Max) ;
   s1 = if (xmax <= e1)
        then xmax
        else (if (e1 > xmin)
              then e1
              else xmin) ;
tel.
```

Designed to control an approximated model of the actual system

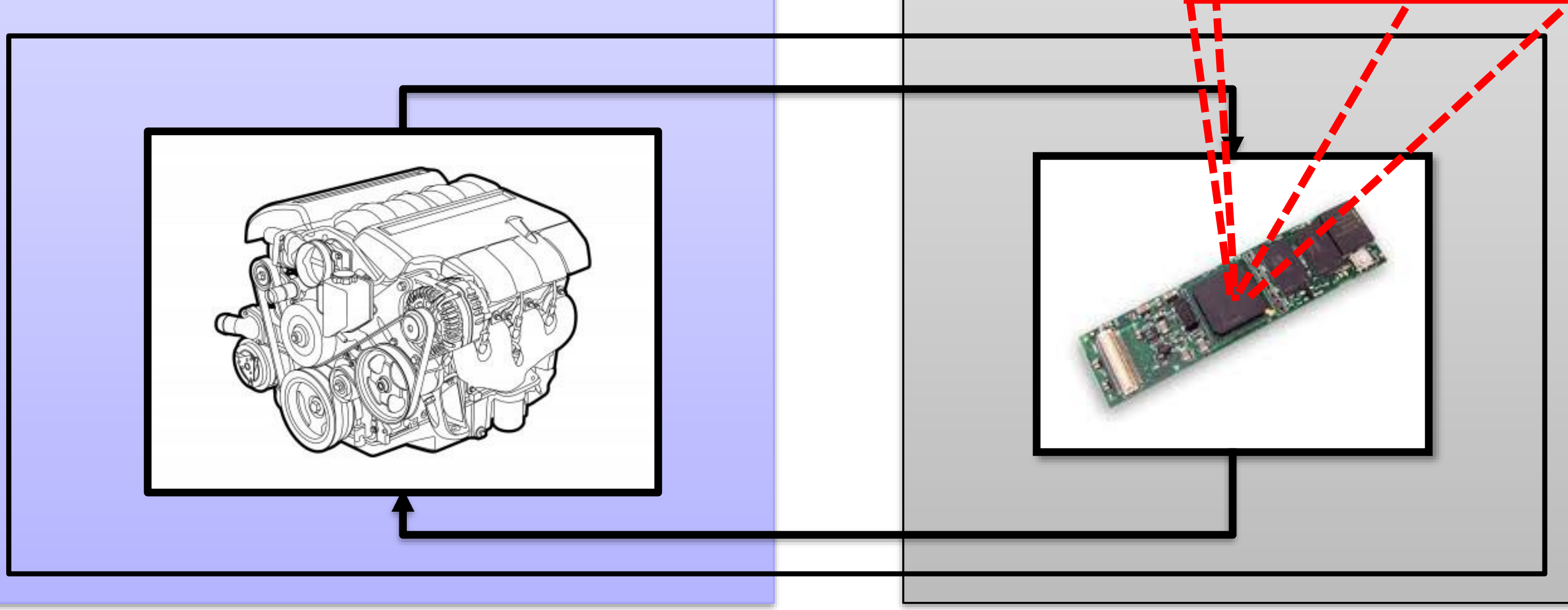
Control design for powertrain

Properties to check are typically on the physical side! (the domain of classical mechanical and electrical engineering)



Classical real-time systems and software engineering methods apply here! Still valuable, but …

```
node Val_Lim(e1 : real; Min, Max : real)
             returns (s1 : real) ;
var xmin:real, xmax : real;
let
  (xmax , xmin) = if (Max >= Min)
                   then (Max , Min)
                   else (Min , Max) ;
   s1 = if (xmax <= e1)
            then xmax
            else (if (e1 > xmin)
                     then e1
                     else xmin) ;
tel.
```

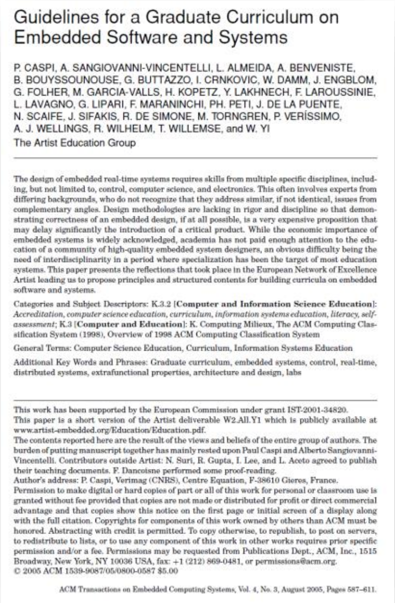


What are the mathematical foundations and algorithmic tools needed so that engineers can design such systems?

HOW CAN WE BRIDGE THE GAP?

Guidelines on CPS Education

- Planning your education in CPS? Then read the following:
  - Caspi et al, Guidelines for a Graduate Curriculum on Embedded Software and Systems, ACM Transactions on Embedded Computing Systems, Vol. 4, No. 3, August 2005, Pages 587–611
  - Henzinger & Sifakis, The Discipline of Embedded Systems Design, Computer, October 2007



Guidelines for a Graduate Curriculum on Embedded Software and Systems

P. CASPI, A. SANGIOVANNI-VINCENTELLI, L. ALMEIDA, A. BENVENISTE, B. BOUYSSOUNOUSE, G. BUTTAZZO, I. CRNKOVIC, W. DAMM, J. ENGBLOM, G. FOLHER, M. GARCIA-VALLS, H. KOPETZ, Y. LAKHNECH, F. LAROUSSINIE, L. LAVAGNO, G. LIPARI, F. MARANINCHI, PH. PETI, J. DE LA PUENTE, N. SCAIFE, J. SIFAKIS, R. DE SIMONE, M. TORNGREN, P. VERÍSSIMO, A. J. WELLINGS, R. WILHELM, T. WILLEMSE, and W. YI
The Artist Education Group

The design of embedded real-time systems requires skills from multiple specific disciplines, including, but not limited to, control, computer science, and electronics. This often involves experts from differing backgrounds, who do not recognize that they address similar, if not identical, issues from complementary angles. Design methodologies are lacking in rigor and discipline so that demonstrating correctness of an embedded design, if at all possible, is a very expensive proposition that may delay significantly the introduction of a critical product. While the economic importance of embedded systems is widely acknowledged, academia has not paid enough attention to the education of a community of high-quality embedded system designers, an obvious difficulty being the need of interdisciplinarity in a period where specialization has been the target of most education systems. This paper presents the reflections that took place in the European Network of Excellence Artist leading us to propose principles and structured contents for building curricula on embedded software and systems.

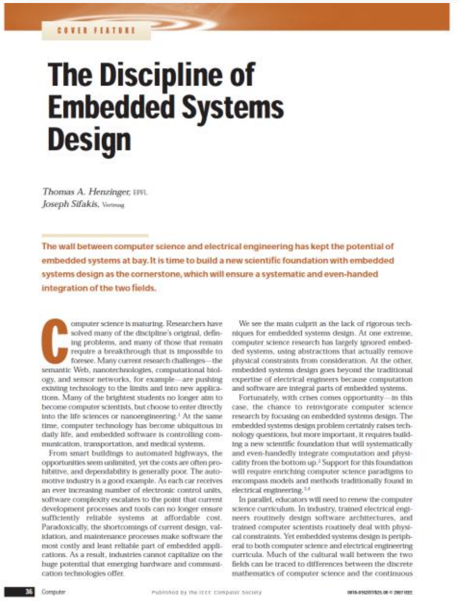
Categories and Subject Descriptors: K.3.2 [**Computer and Information Science Education**]: *Accreditation, computer science education, curriculum, information systems education, literacy, self-assessment*; K.3 [**Computer and Education**]: K. Computing Milieux, The ACM Computing Classification System (1998), Overview of 1998 ACM Computing Classification System

General Terms: Computer Science Education, Curriculum, Information Systems Education

Additional Key Words and Phrases: Graduate curriculum, embedded systems, control, real-time, distributed systems, extrafunctional properties, architecture and design, labs

This work has been supported by the European Commission under grant IST-2001-34820.
This paper is a short version of the Artist deliverable W2.All.Y1 which is publicly available at www.artist-embedded.org/Education/Education.pdf.
The contents reported here are the result of the views and beliefs of the entire group of authors. The burden of putting manuscript together has mainly rested upon Paul Caspi and Alberto Sangiovanni-Vincentelli. Contributors outside Artist: N. Suri, R. Gupta, I. Lee, and L. Aceto agreed to publish their teaching documents. F. Dancoisne performed some proof-reading.
Author's address: P. Caspi, Verimag (CNRS), Centre Equation, F-38610 Gieres, France.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 1515 Broadway, New York, NY 10036 USA, fax: +1 (212) 869-0481, or permissions@acm.org.
© 2005 ACM 1539-9087/05/0800-0587 $5.00

ACM Transactions on Embedded Computing Systems, Vol. 4, No. 3, August 2005, Pages 587–611.



COVER FEATURE

The Discipline of Embedded Systems Design

Thomas A. Henzinger, EPFL
Joseph Sifakis, Verimag

The wall between computer science and electrical engineering has kept the potential of embedded systems at bay. It is time to build a new scientific foundation with embedded systems design as the cornerstone, which will ensure a systematic and even-handed integration of the two fields.

Computer science is maturing. Researchers have solved many of the discipline's original, defining problems, and many of those that remain require a breakthrough that is impossible to foresee. Many current research challenges—the semantic Web, nanotechnologies, computational biology, and sensor networks, for example—are pushing existing technology to the limits and into new applications. Many of the brightest students no longer aim to become computer scientists, but choose to enter directly into the life sciences or nanoengineering.[1] At the same time, computer technology has become ubiquitous in daily life, and embedded software is controlling communication, transportation, and medical systems.

From smart buildings to automated highways, the opportunities seem unlimited, yet the costs are often prohibitive, and dependability is generally poor. The automotive industry is a good example. As each car receives an ever increasing number of electronic control units, software complexity escalates to the point that current development processes and tools can no longer ensure sufficiently reliable systems at affordable cost. Paradoxically, the shortcomings of current design, validation, and maintenance processes make software the most costly and least reliable part of embedded applications. As a result, industries cannot capitalize on the huge potential that emerging hardware and communication technologies offer.

We see the main culprit as the lack of rigorous techniques for embedded systems design. At one extreme, computer science research has largely ignored embedded systems, using abstractions that actually remove physical constraints from consideration. At the other, embedded systems design goes beyond the traditional expertise of electrical engineers because computation and software are integral parts of embedded systems.

Fortunately, with crises comes opportunity—in this case, the chance to reinvigorate computer science research by focusing on embedded systems design. The embedded systems design problem certainly raises technology questions, but more important, it requires building a new scientific foundation that will systematically and even-handedly integrate computation and physicality from the bottom up.[2] Support for this foundation will require enriching computer science paradigms to encompass models and methods traditionally found in electrical engineering.[3,4]

In parallel, educators will need to renew the computer science curriculum. In industry, trained electrical engineers routinely design software architectures, and trained computer scientists routinely deal with physical constraints. Yet embedded systems design is peripheral to both computer science and electrical engineering curricula. Much of the cultural wall between the two fields can be traced to differences between the discrete mathematics of computer science and the continuous

36 Computer Published by the IEEE Computer Society 0018-9162/07/$25.00 © 2007 IEEE

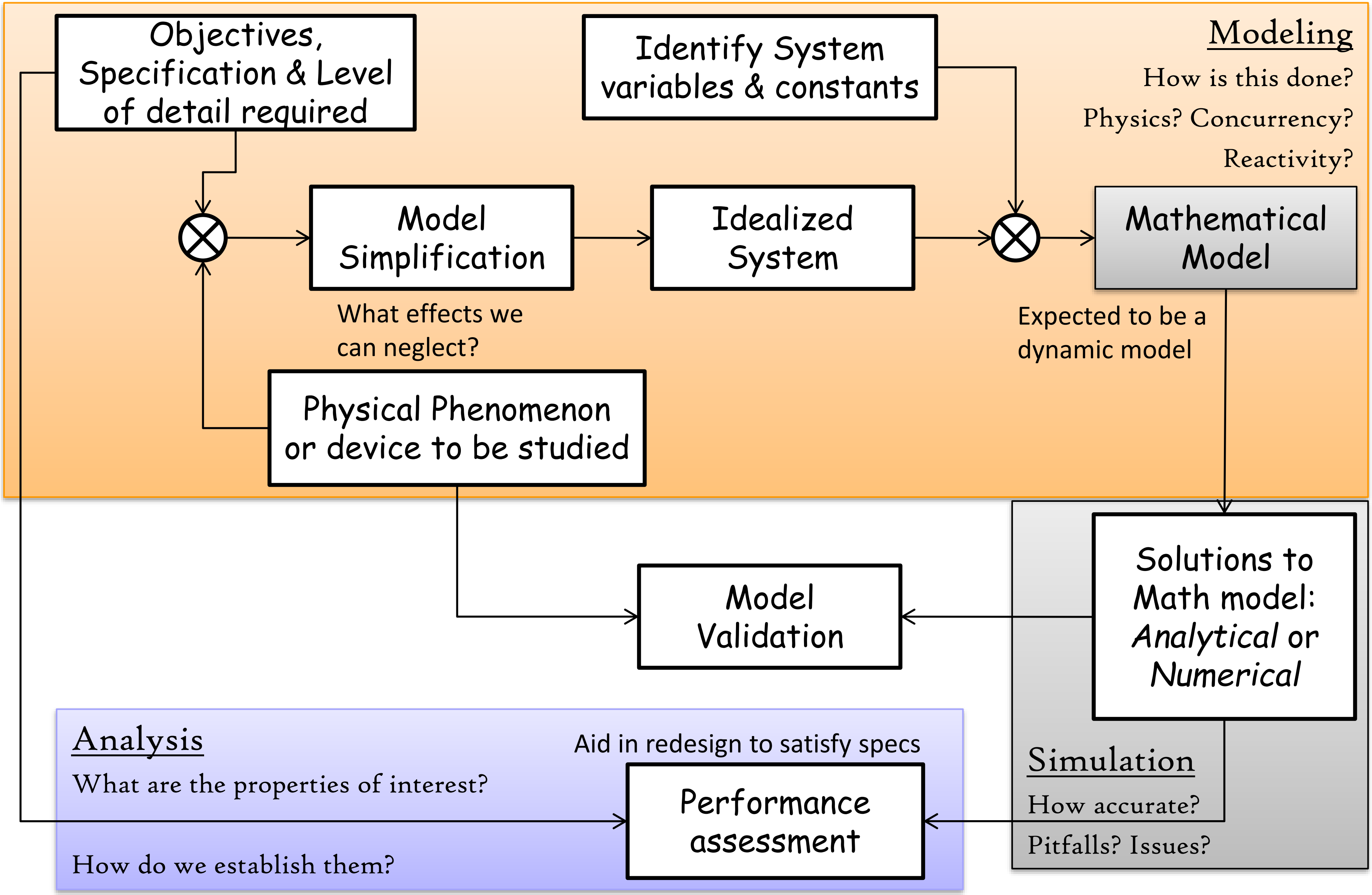
Recommended Curriculum

1. Foundations of Computer Science and Engineering
   - Algorithms, Computer architecture, Language theory (automata, etc), Programming languages, Operating systems, and Software engineering

2. Control, Signal processing, and Communication
   - Modeling, Control design, Signal processing, Discrete event systems

3. Hybrid systems (CS + Control + Communication)

Model Based Development for CPS


Objectives, Specification & Level of detail required
Identify System variables & constants
Modeling
How is this done?
Physics? Concurrency?
Reactivity?
Model Simplification
Idealized System
Mathematical Model
What effects we can neglect?
Expected to be a dynamic model
Physical Phenomenon or device to be studied
Solutions to Math model: *Analytical* or *Numerical*
Model Validation
Analysis
What are the properties of interest?
Aid in redesign to satisfy specs
Performance assessment
How do we establish them?
Simulation
How accurate?
Pitfalls? Issues?

Adapted from T. D. Burton: Introduction to Dynamic System Analysis

What is an appropriate model?

What are properties of interest?

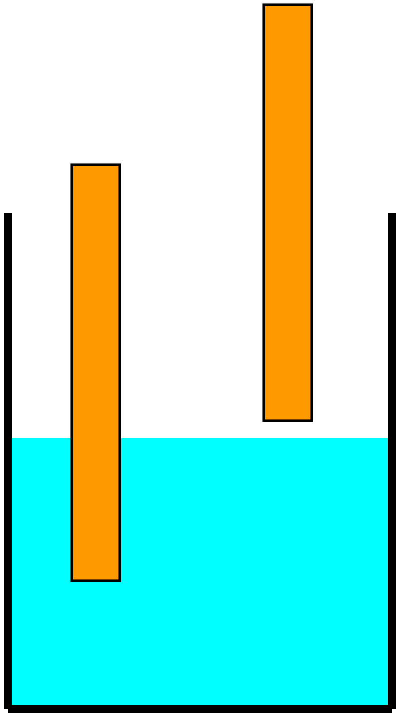
EXAMPLES OF MODEL BASED DESIGN FOR CPS: NUCLEAR REACTOR

Nuclear reactor example

Without rods $\dot{T} = 0.1T - 50$

With rod 1 $\dot{T} = 0.1T - 56$

With rod 2 $\dot{T} = 0.1T - 60$



Requirements:

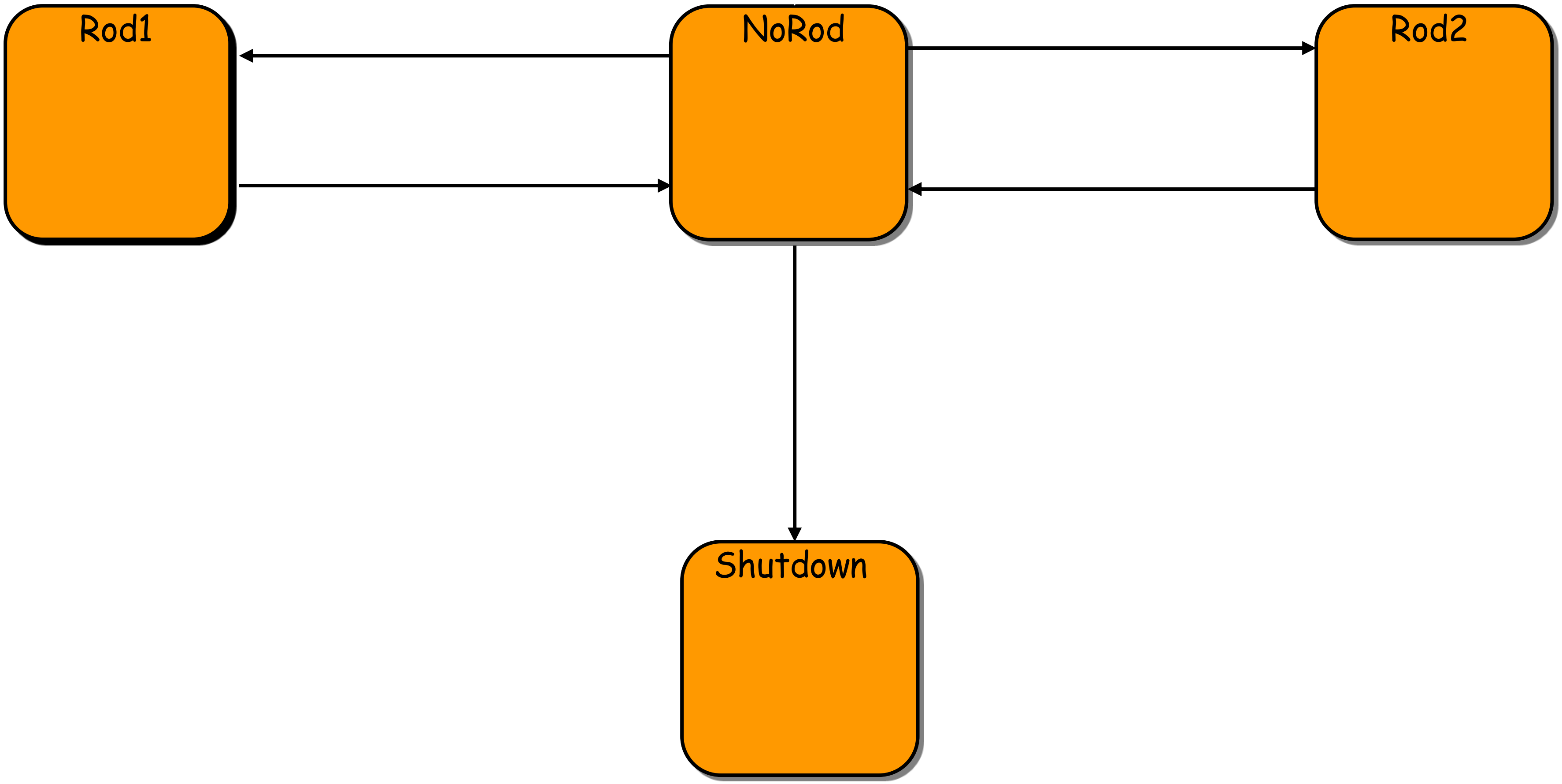
Rod 1 and 2 cannot be used simultaneously

Once a rod is removed, you cannot use it for 10 minutes

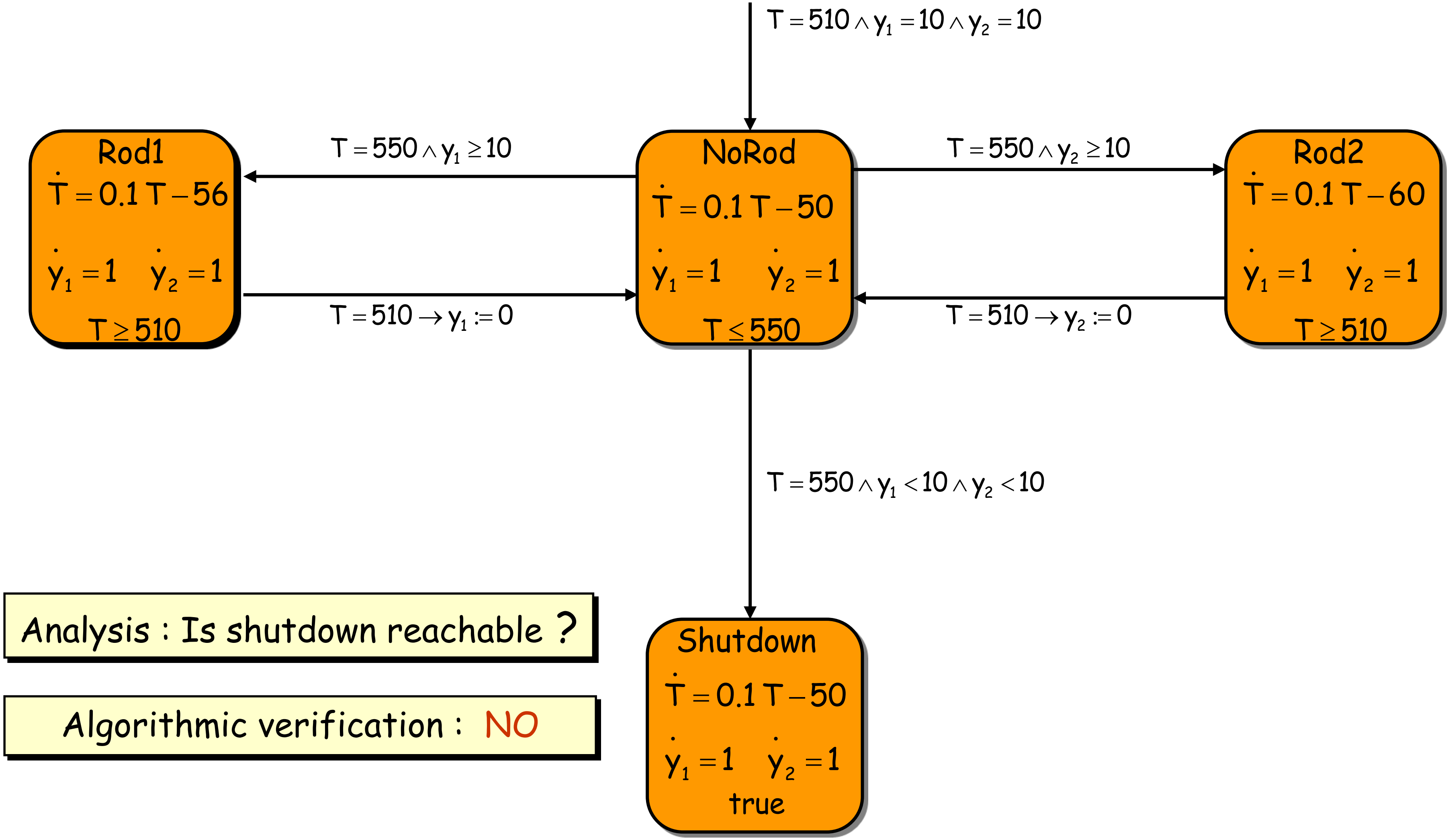
Specification : Keep temperature between 510 and 550 degrees.

If T=550 then either a rod is available or we shutdown the plant.

Software model of nuclear reactor



Hybrid model of nuclear reactor



$T = 510 \wedge y_1 = 10 \wedge y_2 = 10$

Rod1
$\dot{T} = 0.1T - 56$
$\dot{y}_1 = 1 \quad \dot{y}_2 = 1$
$T \geq 510$

NoRod
$\dot{T} = 0.1T - 50$
$\dot{y}_1 = 1 \quad \dot{y}_2 = 1$
$T \leq 550$

Rod2
$\dot{T} = 0.1T - 60$
$\dot{y}_1 = 1 \quad \dot{y}_2 = 1$
$T \geq 510$

Shutdown
$\dot{T} = 0.1T - 50$
$\dot{y}_1 = 1 \quad \dot{y}_2 = 1$
true

$T = 550 \wedge y_1 \geq 10$

$T = 510 \rightarrow y_1 := 0$

$T = 550 \wedge y_2 \geq 10$

$T = 510 \rightarrow y_2 := 0$

$T = 550 \wedge y_1 < 10 \wedge y_2 < 10$

Analysis : Is shutdown reachable ?

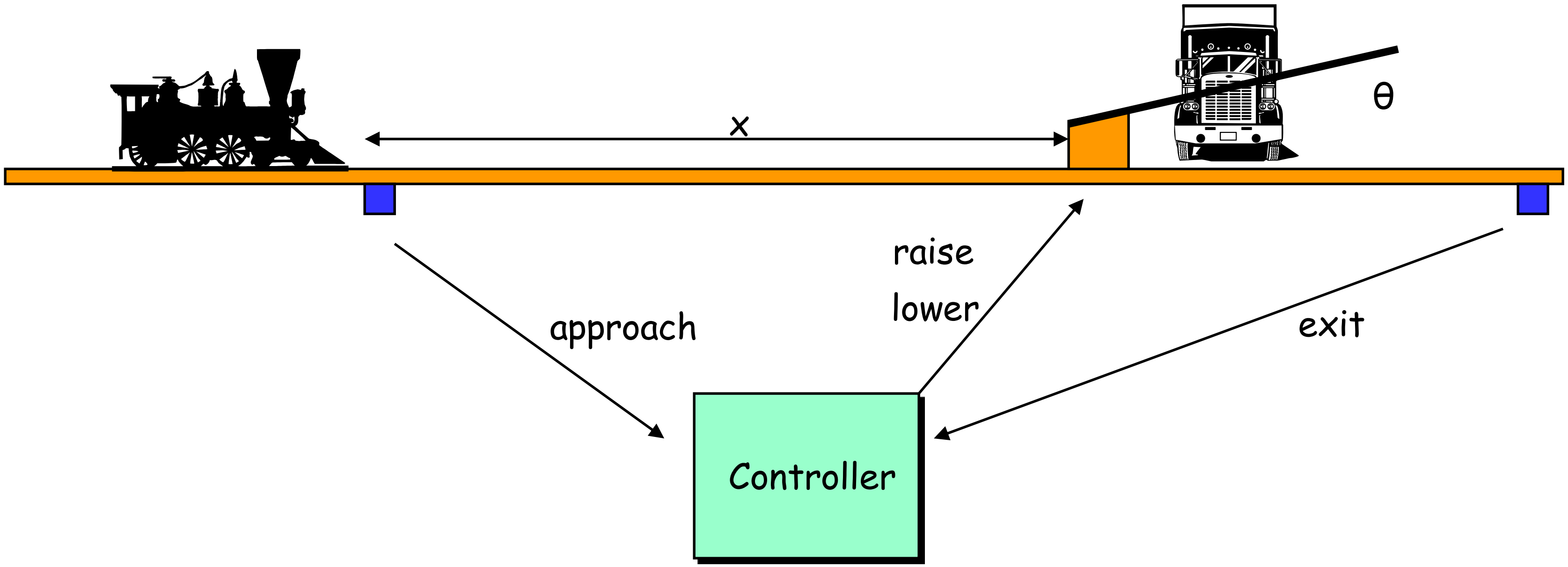
Algorithmic verification : NO

What is an appropriate model?

What are properties of interest?

EXAMPLES OF MODEL BASED DESIGN FOR CPS: TRAIN GATE CONTROLLER

The train gate example

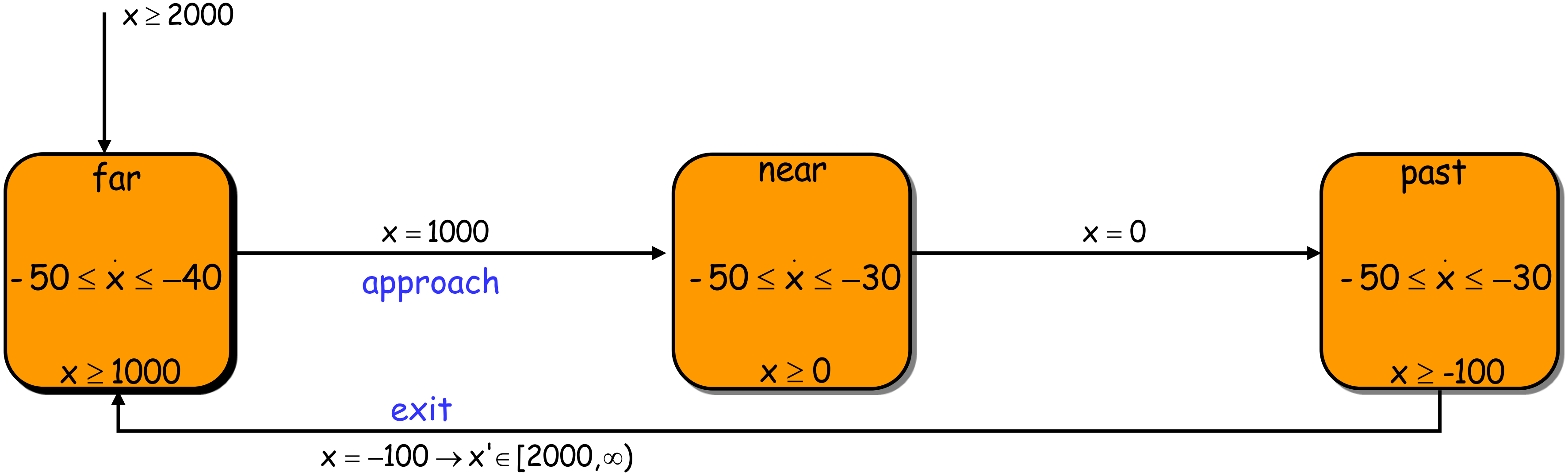


$$\text{System} = \text{Train} \,||\, \text{Gate} \,||\, \text{Controller}$$

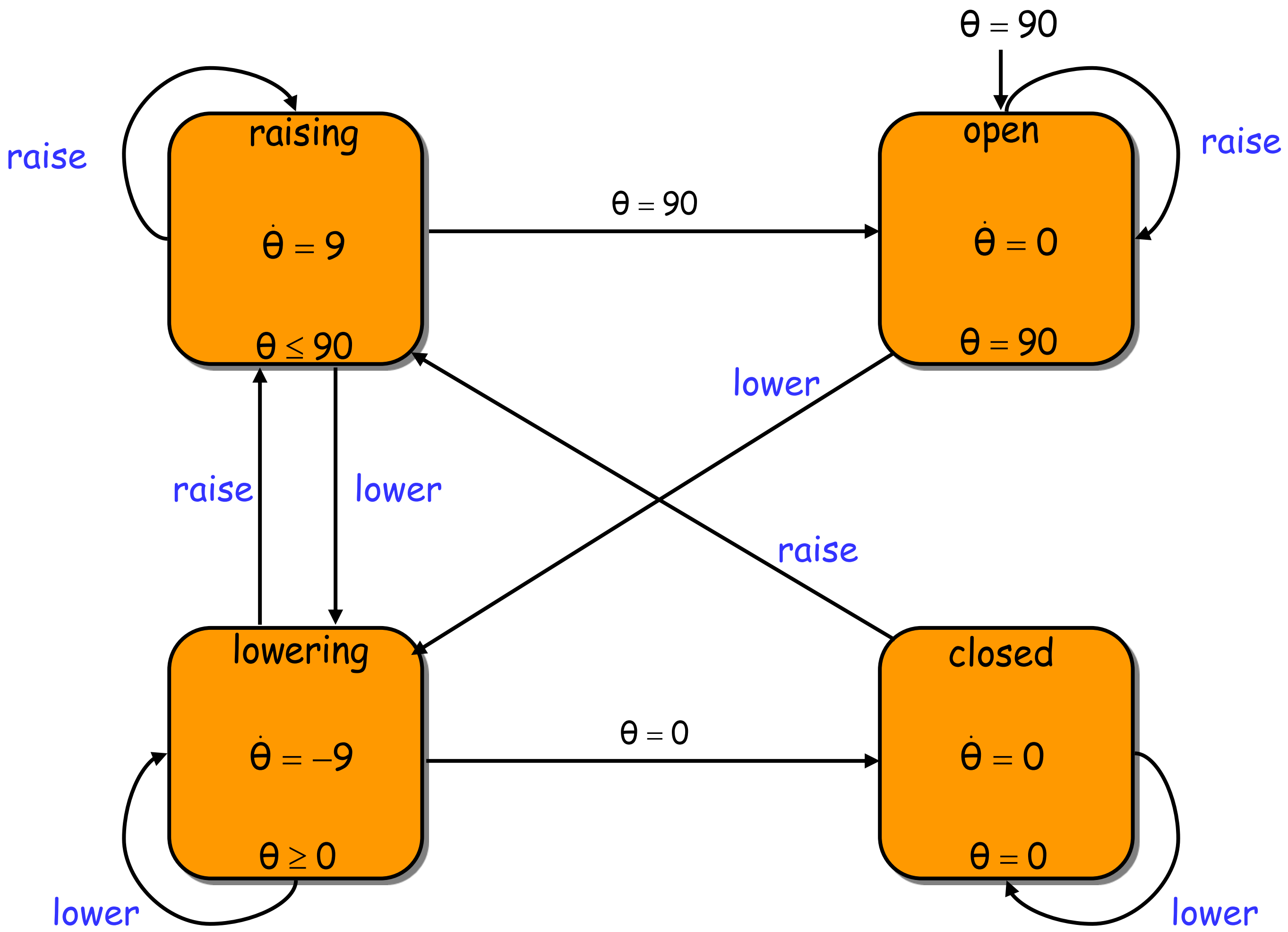
Safety specification : If train is within 10 meters of the crossing, then the gate should be completely closed.

Liveness specification : Keep gate open as much as possible.

Train model

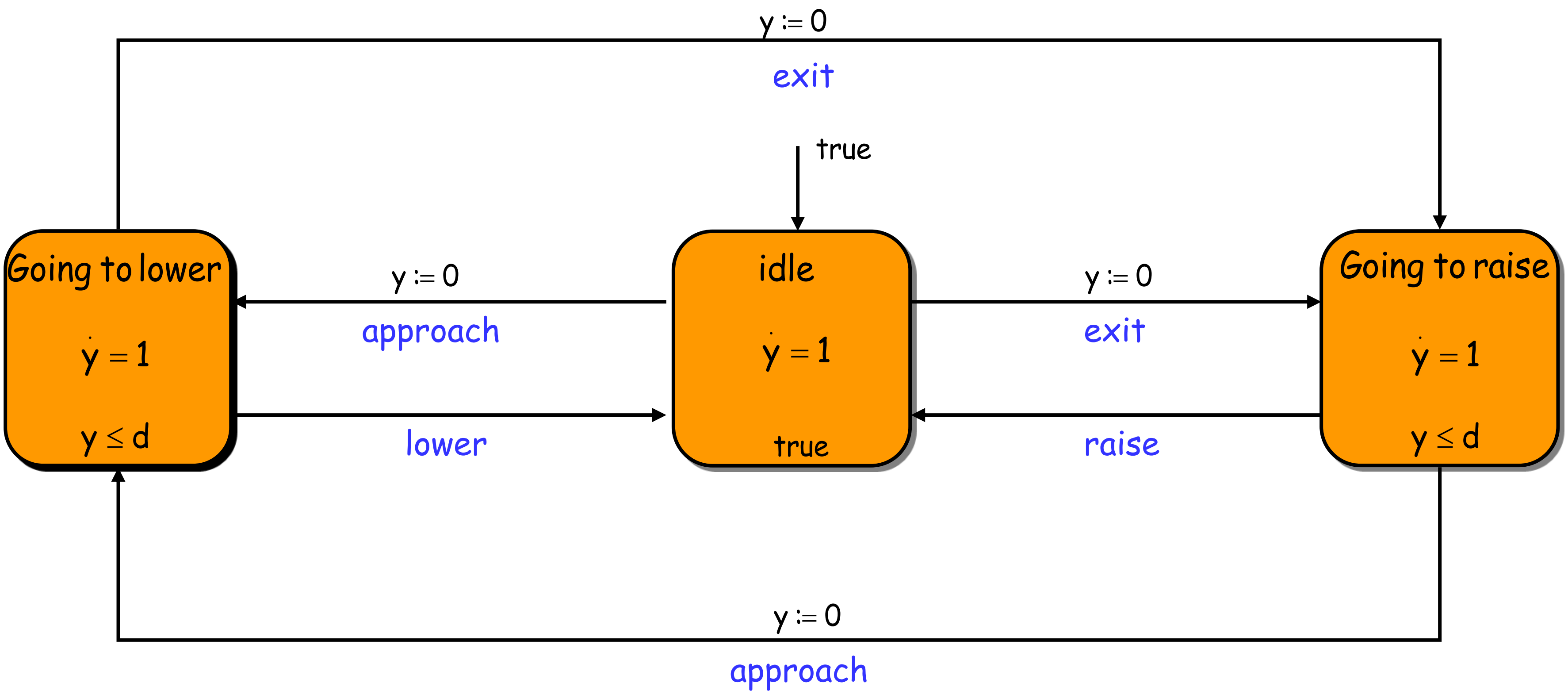

$x \geq 2000$
far
$-50 \leq \dot{x} \leq -40$
$x \geq 1000$
$x = 1000$
approach
near
$-50 \leq \dot{x} \leq -30$
$x \geq 0$
$x = 0$
past
$-50 \leq \dot{x} \leq -30$
$x \geq -100$
exit
$x = -100 \rightarrow x' \in [2000, \infty)$

Gate model

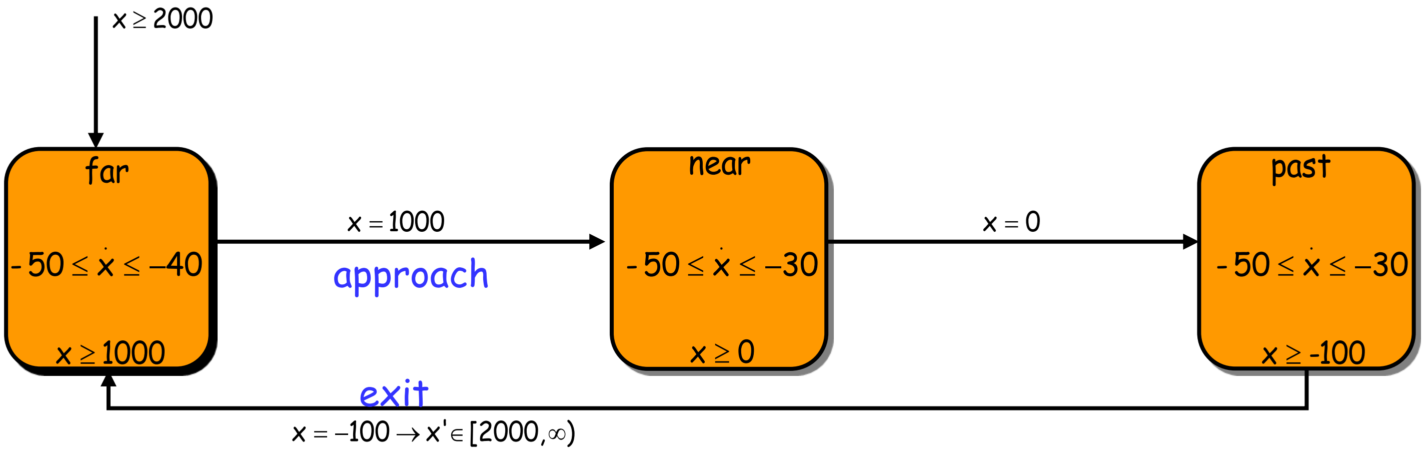


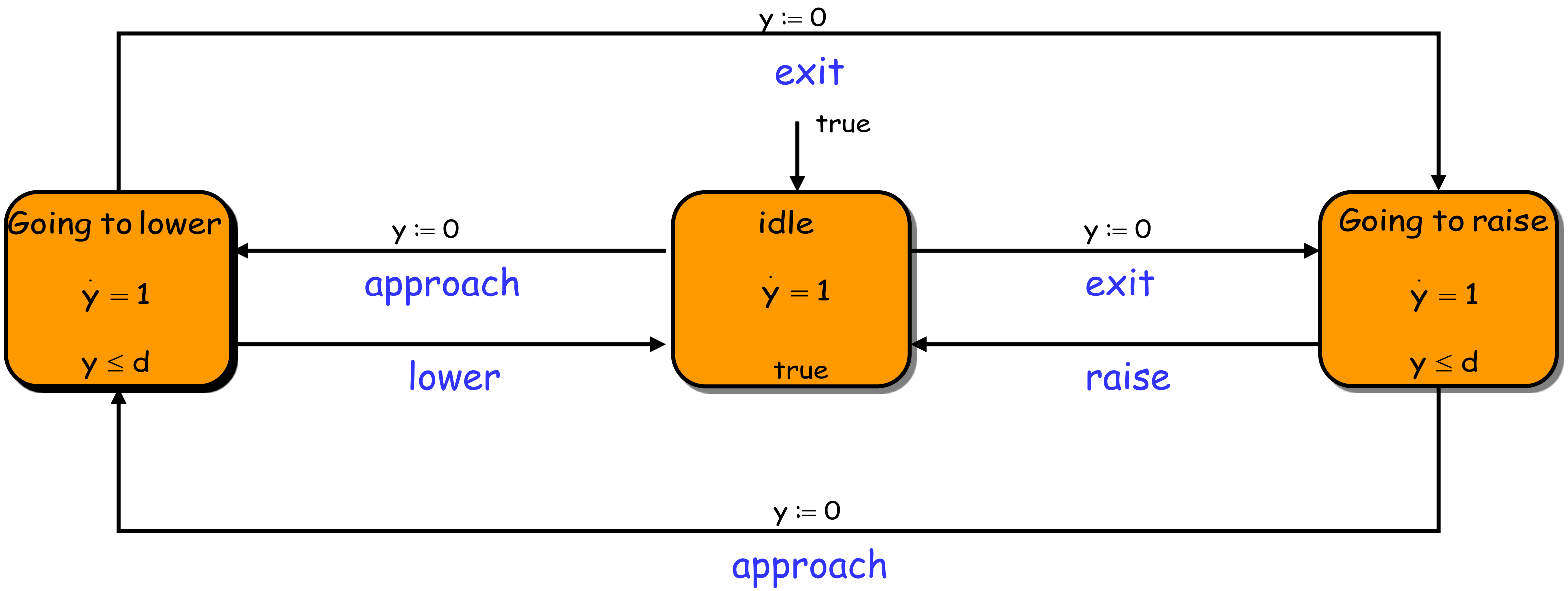
$\theta = 90$
raising
$\dot{\theta} = 9$
$\theta \leq 90$
raise
$\theta = 90$
open
$\dot{\theta} = 0$
$\theta = 90$
raise
lower
raise
lower
raise
lowering
$\dot{\theta} = -9$
$\theta \geq 0$
$\theta = 0$
closed
$\dot{\theta} = 0$
$\theta = 0$
lower
lower

Controller model

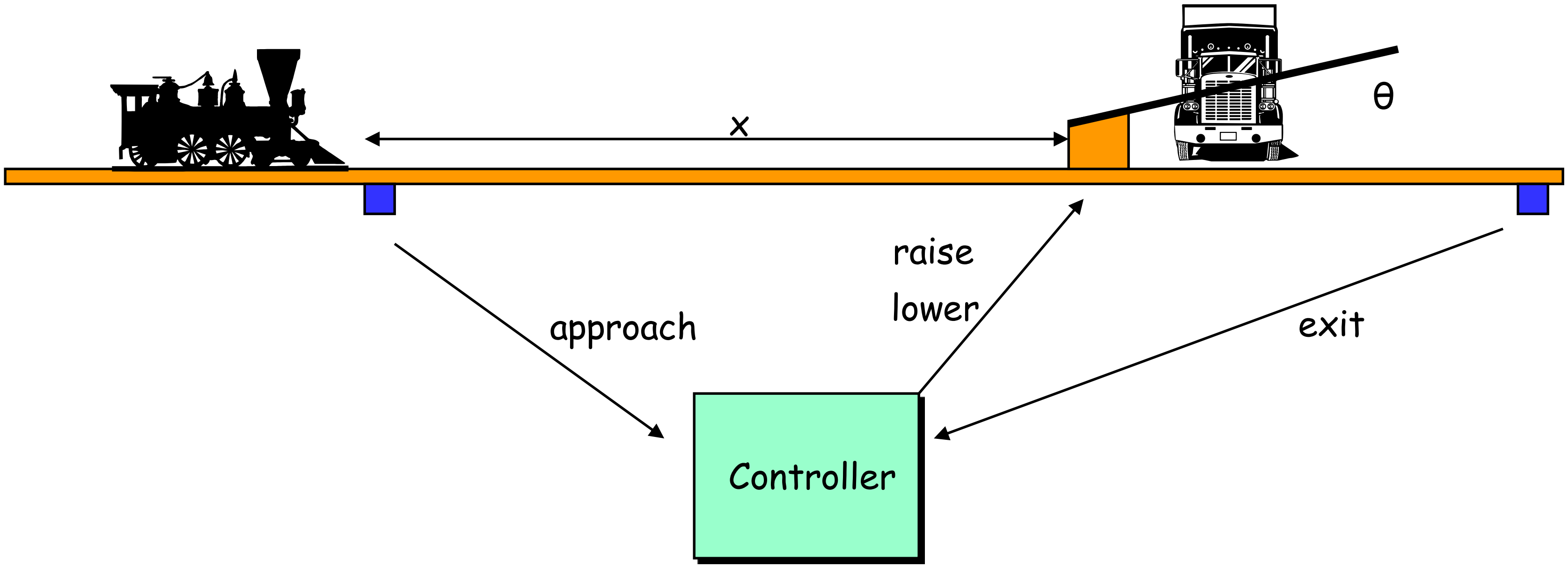


Synchronized transitions


$x \geq 2000$
far
$-50 \leq \dot{x} \leq -40$
$x \geq 1000$
$x = 1000$
approach
near
$-50 \leq \dot{x} \leq -30$
$x \geq 0$
$x = 0$
past
$-50 \leq \dot{x} \leq -30$
$x \geq -100$
exit
$x = -100 \to x' \in [2000, \infty)$



Verifying the controller



$$System = Train \,||\, Gate \,||\, Controller$$

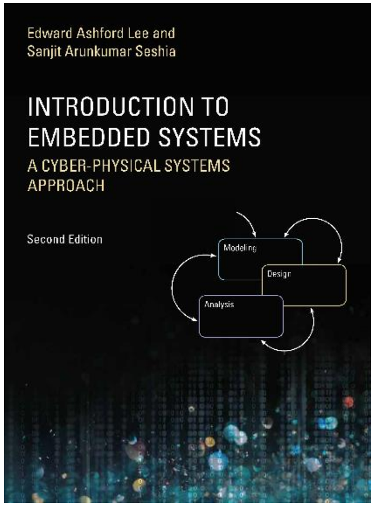
Safety specification : Can we avoid the set $\theta > 0 \wedge (-10 \leq x \leq 10)$?

Parametric verification : $\text{YES if } d \leq \frac{49}{5}$

Which textbooks support such an MBD approach to teaching foundations of CPS?

TEXTBOOKS

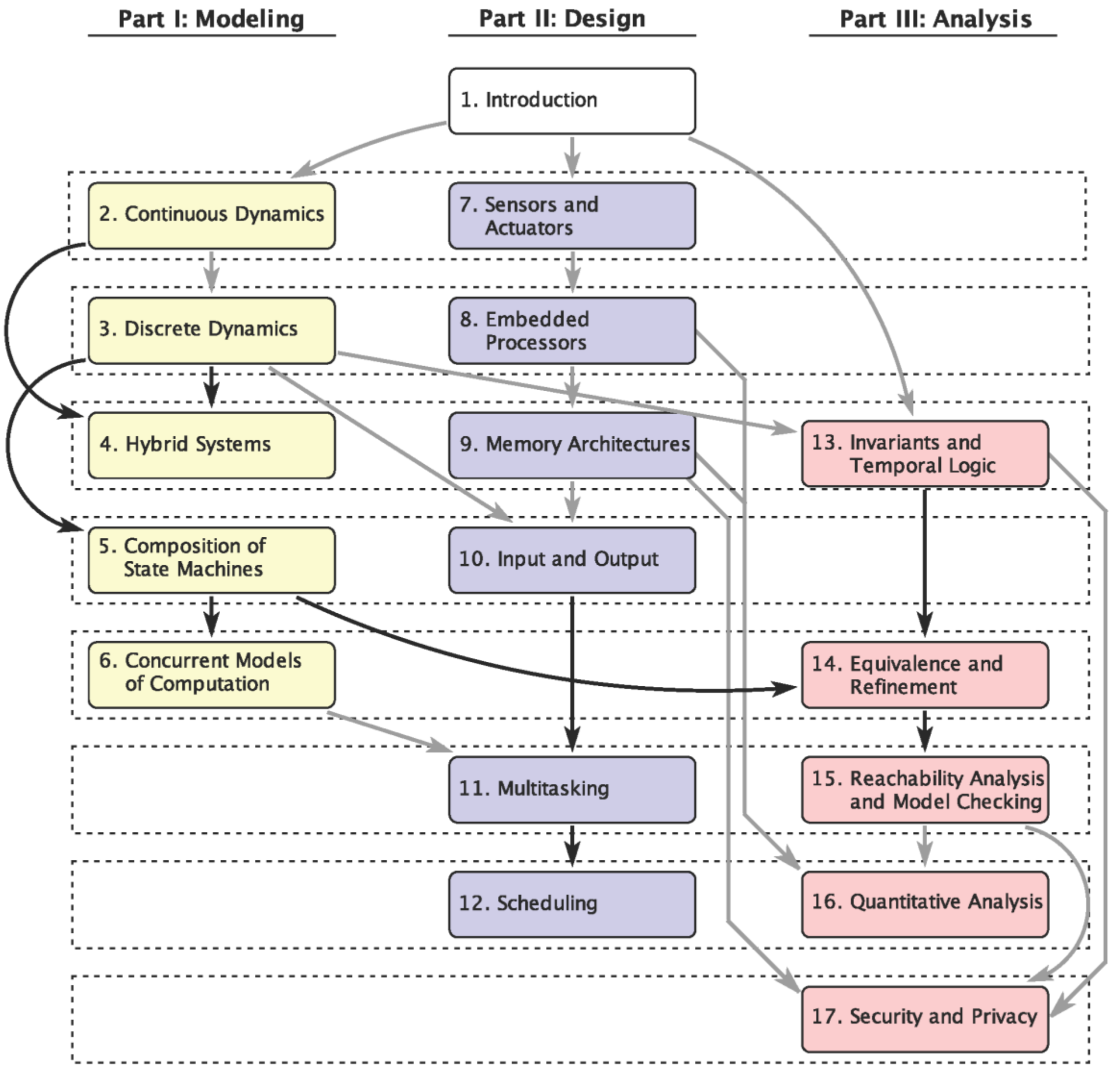
Senior undergraduate and graduate level



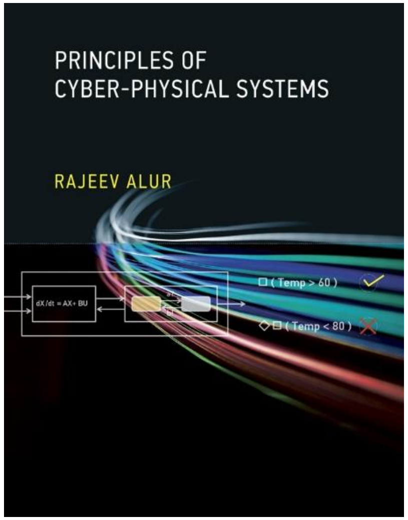
Lee and Seshia

Introduction to Embedded Systems

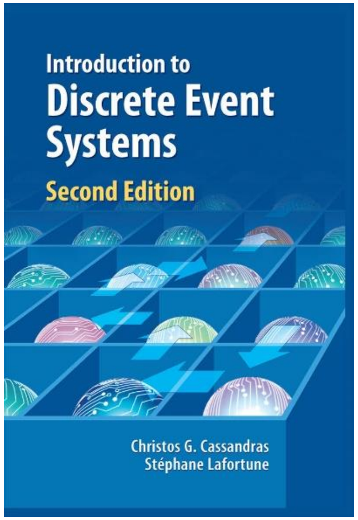
— A Cyber-Physical Systems Approach



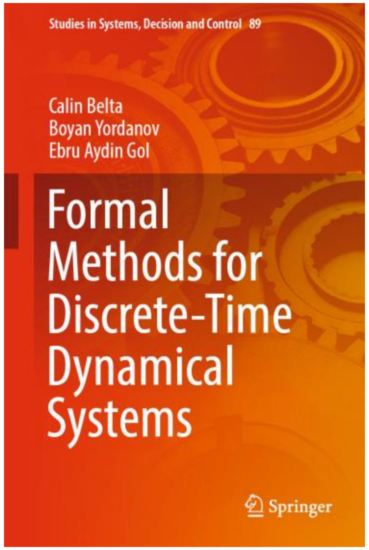
Graduate level



Rajeev Alur
Principles of Cyber-Physical Systems
By MIT Press

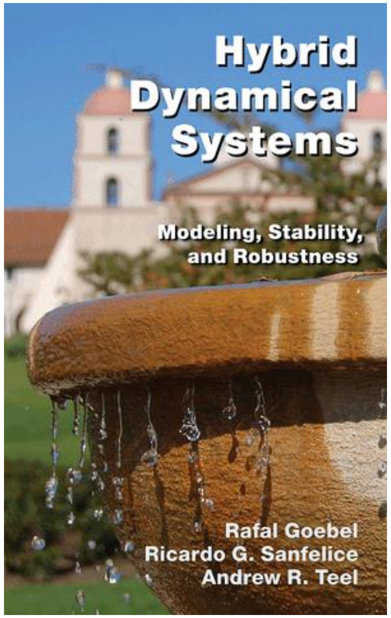


Cassandras and Lafortune,
Introduction to Discrete Event Systems
Springer

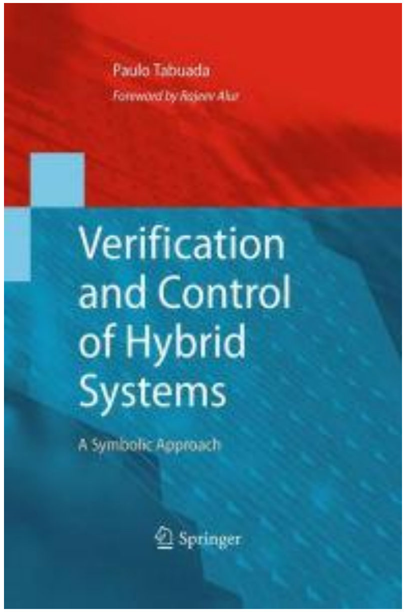


Belta, Yordanov & Gol
Formal Methods for Discrete-Time Dynamical Systems
Springer

Graduate level



Goebel, Sanfelice & Teel
Hybrid Dynamical Systems:
Modeling, Stability, and Robustness
Princeton University Press



P. Tabuada,
Verification and control of hybrid systems:
a symbolic approach,
Springer-Verlag

Why is it important?

TEACHING FORMAL REQUIREMENTS

Trust? : Sampling of automotive recalls (~2011-12) due to software errors …

- "A software error may prevent the transmission from downshifting, such as shifting from 5th y of the problem. e stall, increasing

When in 5th gear and RPM drops below x, then the system should always switch from 5th to 4th gear.

- … the software that "allows the ECU to establish a 'handshake' with the engine is in error. The E ngine is found to be out of tol ppens, the ECU triggers a fau lition outside its prescribed tolerances, a rough idle or stalling situation ensues."

The engine should never stall while idle.

- … to u tain circums tor to rotate in the direction opposite to that selected by the transmission.

The electric motor should always rotate in the direction selected by the transmission.

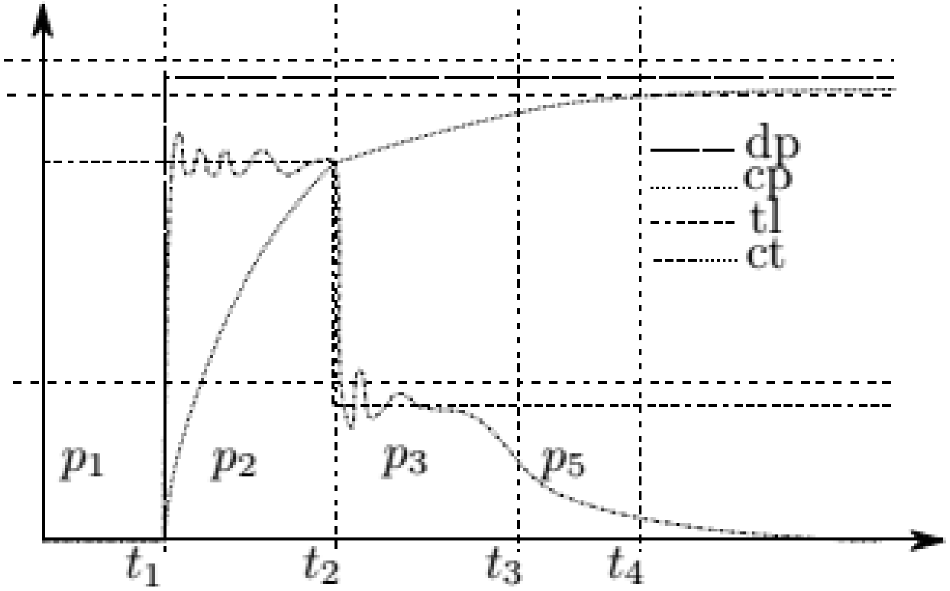
- If the fault gnition while driving - w disables power ste

The cruise control should always disengage when the "turn off" button is pressed.

- …

How complex can specifications be*?

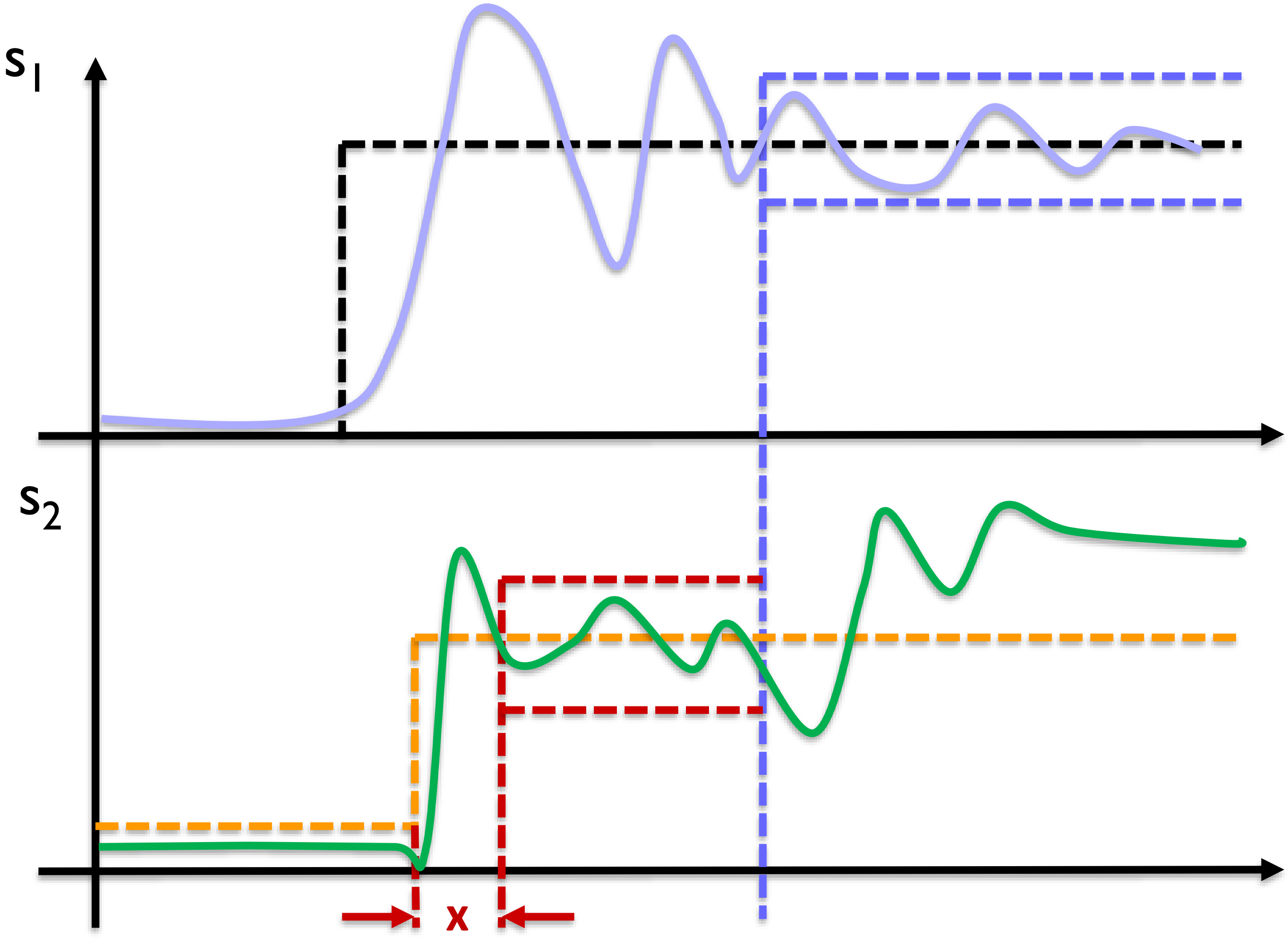
NL: During the position (cp) regulation after a step input on demand (dp), when the absolute value of the maximum torque limit (tl) decreases with a step (precondition), the absolute value of the actuator response in torques (ct) must be less than the torque limit plus 10% in less than 10 ms (postcondition)



* *H. Roehm, R. Gmehlich, T. Heinz, J. Oehlerking and M. Woehrle: Industrial Examples of Formal Specifications for Test Case Generation, ARCH 2015*

Specification: When ORANGE event happens after the BLACK EVENT, signal s_2 should stabilize in the RED region within x time units. Signal s_2 should only stay in the RED region only until signal s1 has stabilized in the BLUE region.

How do we mathematically capture such requirements so that we can automatically verify/test a system?



Example adapted from Bosch requirements

Metric Interval Temporal Logic: Semantic Intuition

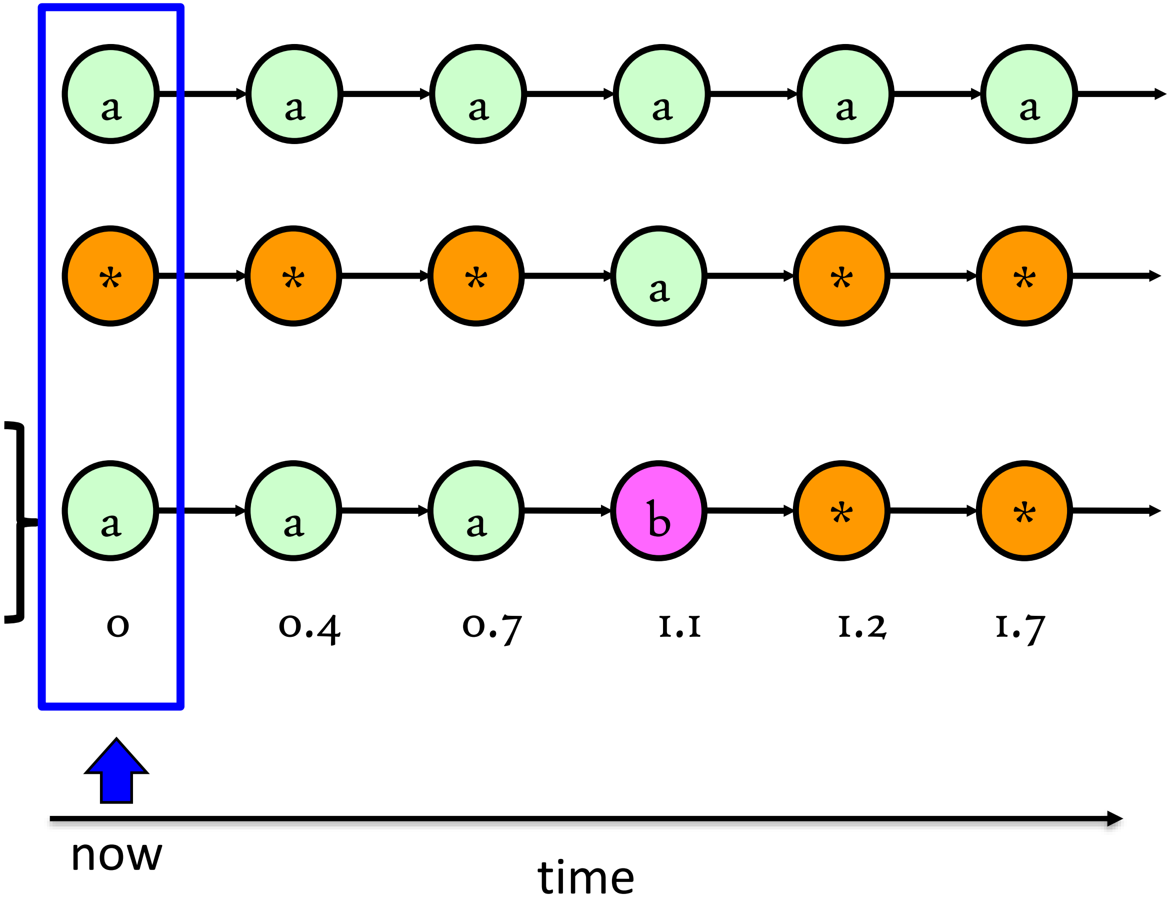
$$\phi \ ::= \top \ | \ p \ | \ \neg\phi \ | \ \phi_1 \vee \phi_2 \ | \ \mathrm{G}_I\phi \ | \ \mathrm{F}_I\phi \ | \ \phi_1 U_I \phi_2$$

$\mathrm{G}a$ - always a

$\mathrm{F}_{[1,3]}a$ - eventually a

$a\ U\ b$ - a until b

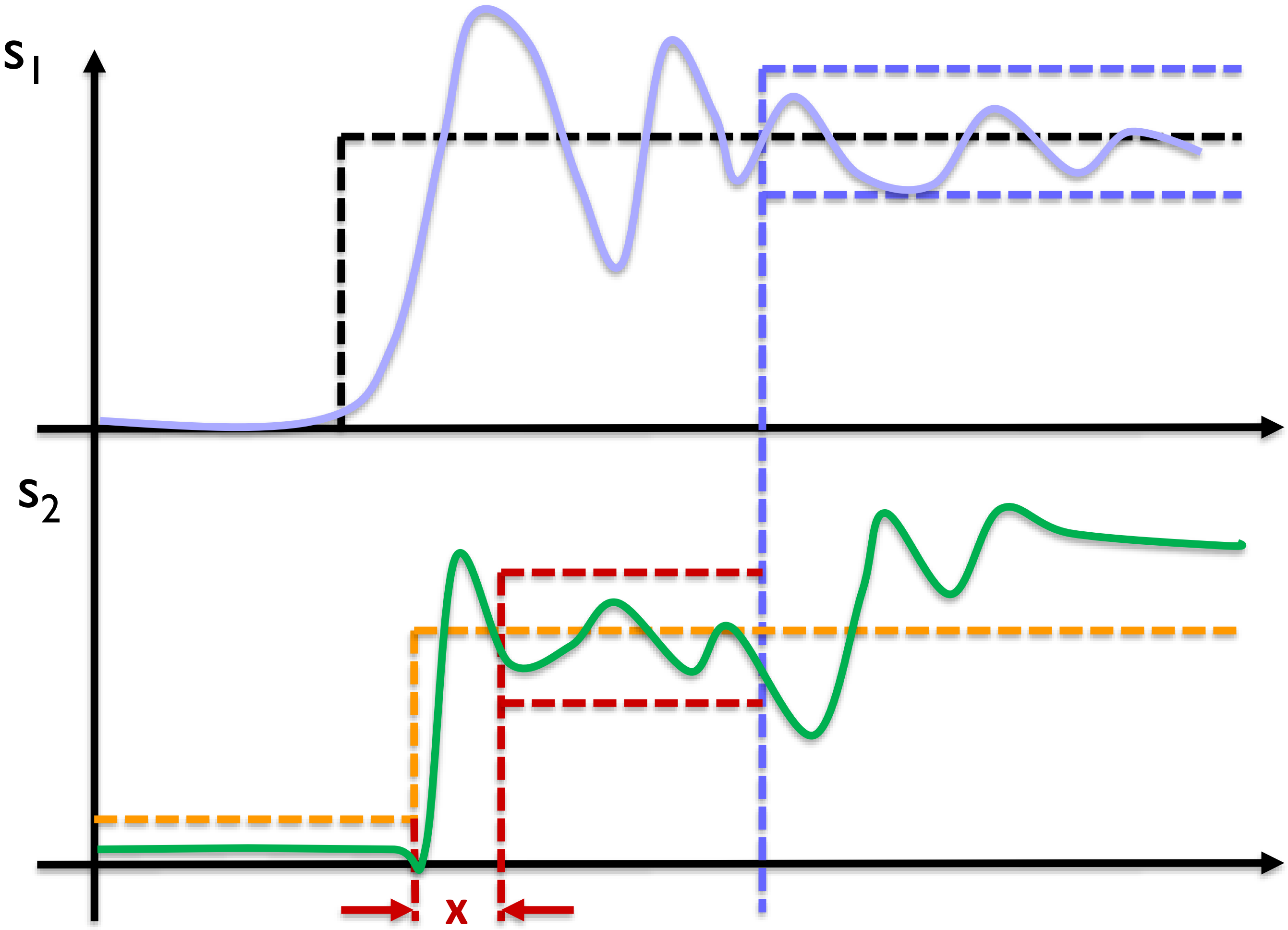
$a\ U_{[1,1.5]}\ b$ - a until b



Possible formalizations?

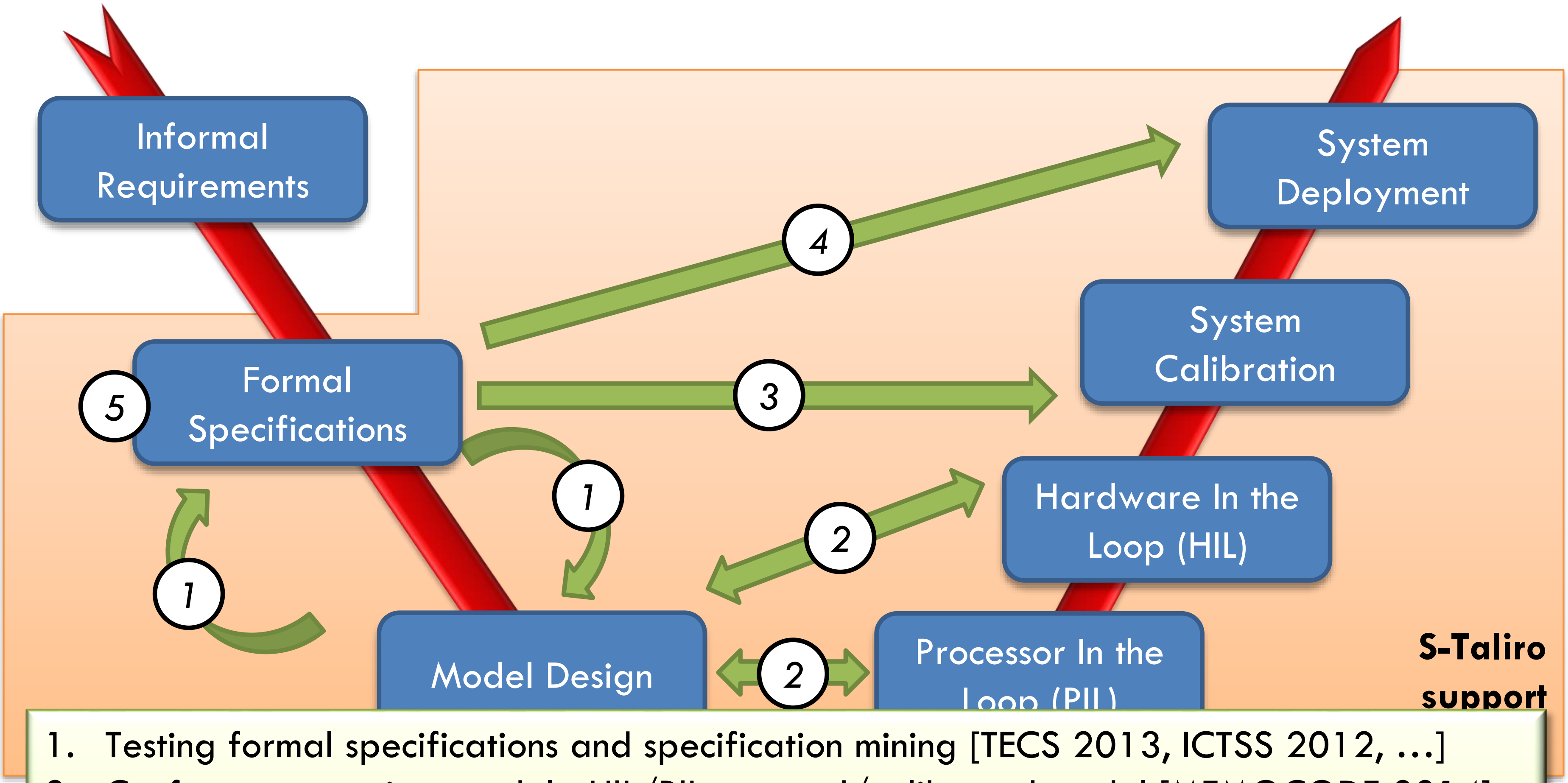
$$G(\ (\text{Orange} \wedge P_{[0,y]}\ \text{Black}) \rightarrow F_{[0,x]}(\ (\text{s2 in red})\ U\ G\ (\text{s1 in blue})\ ))$$

$$G(\ (\text{Orange} \wedge P_{[0,y]}\ \text{Black}) \rightarrow G_{[x,\infty)}(\ (\text{s2 in red}) \vee G\ (\text{s1 in blue})\ ))$$



Example adapted from Bosch requirements

S-Taliro support in the V-process



1. Testing formal specifications and specification mining [TECS 2013, ICTSS 2012, …]
2. Conformance testing: models, HIL/PIL or tuned/calibrated model [MEMOCODE 2014]
3. Testing formal specifications on the HIL/PIL calibrated system [TECS 2013, …]
4. Runtime monitoring of formal requirements [RV 2014]
5. Specification visualization [IROS 2015] & Debugging [MEMOCODE 2015]

Trial in Actual Control Model (Past defect case)

Detect following defect on SiLS model including all engine control
“monitor value－request value>50” continue over 500msec

There are 75 Control point

Generated input
Gas pedal[%]
Brake[%]
Shift{P,N,D}
Water temp[℃]
Air temp[℃]
Air pressure[kPa]
Air conditioner SW

Defect condition
① Specific logic on
② Engine revolution around 4000rpm
③ Satisfy ①,② and specific accelerator operation

Tried 6 large-scale models,
5 models were falsified.

(Past defect case,intential defect by logic developer)

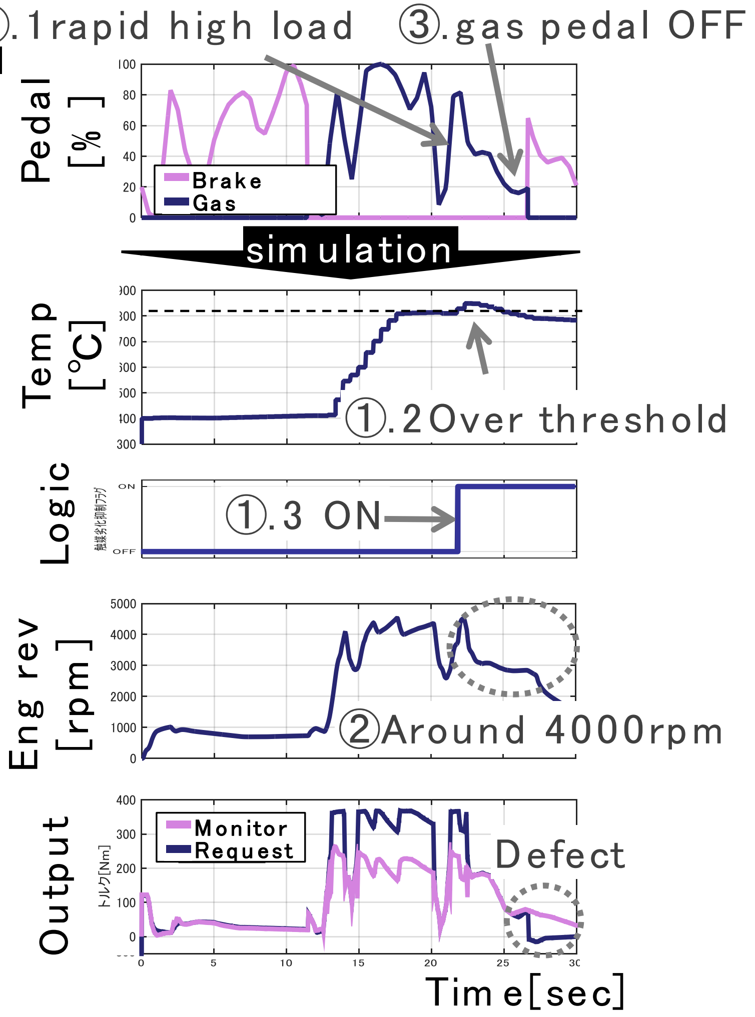


Figure Generated signals automatically

S-Taliro could generate the complicated scenario including the defect

Shunsuke Kobuna

TOYOTA

WHAT IS THE CHALLENGE IN FORMALIZING REQUIREMENTS?

Student homework (Graduate class): Formalizing requirements

- Traditional section of the class (31 students)

Problem difficulty	Very Easy	Very Easy	Easy
Average	9.4	9.6	7.2
Median	10.0	10	6
Max	10	10	10
Min	7.1	6.7	4

- On-line section (10 professional* students)

Problem difficulty	Very Easy	Very Easy	Easy
Average	7.7	7.7	6.8
Median	8.6	7.8	6.0
Max	10	10	10
Min	4.3	4.4	0

* Typically working engineers

Motivating Example: On-Line Survey

We asked:

"At some time in the first 30 seconds, the vehicle speed (v) will go over 100 and stay above 100 for 20 seconds"

Response:

$$\varphi = \Diamond_{[0,30]}(\boxed{(v > 100)} \Rightarrow \Box_{[0,20]}\boxed{(v > 100)})$$

φ is a tautology

- $(v > 100) = \perp$ at any time in [0,30]

$$((v > 100) \boxed{\Rightarrow} \Box_{[0,20]}(v > 100)) = \top$$

- $(v > 100) = \top$ for all the time in [0,30]

$$\Box_{[0,20]}(v > 100) = \top\text{between } [0,10]$$

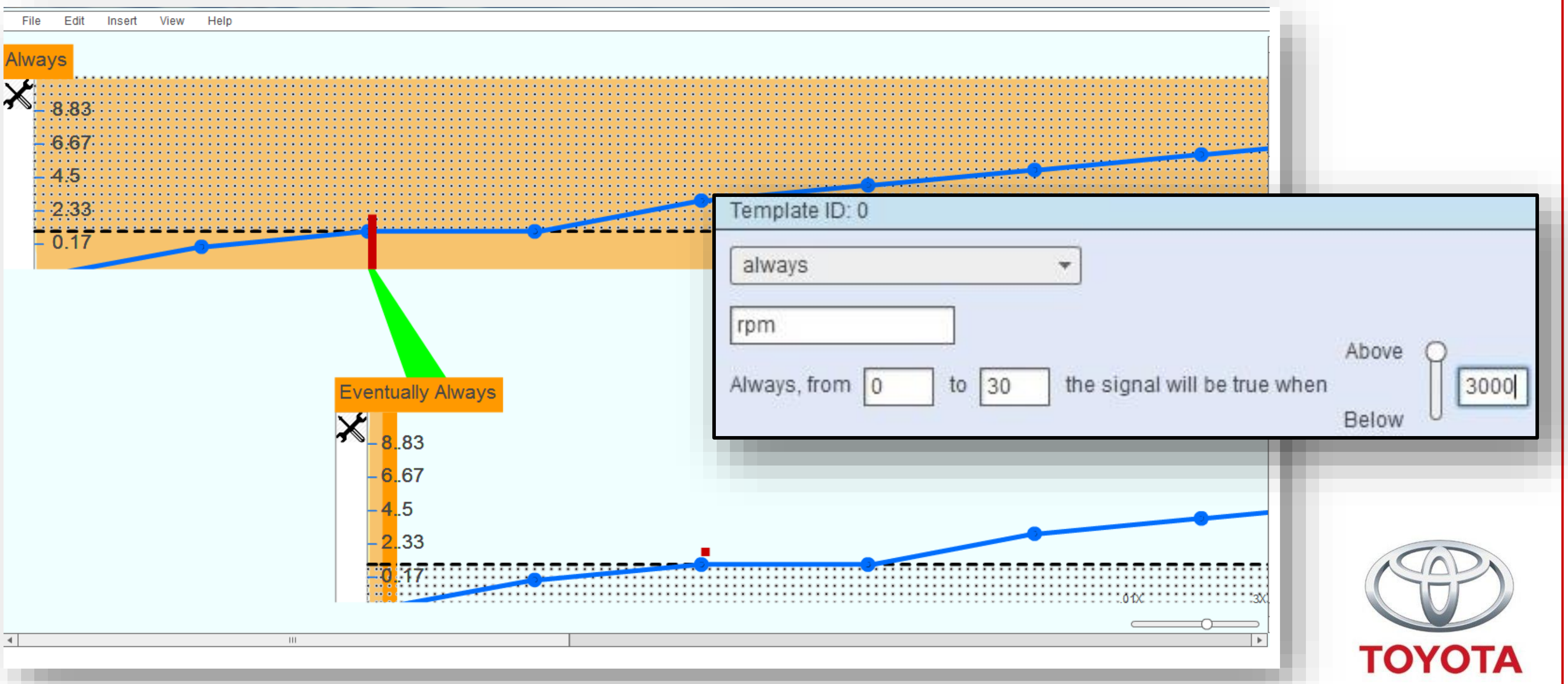
$$((v > 100) \boxed{\Rightarrow} \Box_{[0,20]}(v > 100)) = \top \text{ between } [0,10]$$

B. Hoxha, N. Mavridis and G. Fainekos, VISPEC: A graphical tool for easy elicitation of MTL requirements, IROS 2015

Visual Specification Language (ViSpec)

We have developed a graphical formalism for MTL specification elicitation. Example:

$$\phi_5 = G((\lambda_{diff} > 0.1) \rightarrow F_{[0,1]} G_{[0,1]}(\lambda_{diff} < 0.1))$$



Template ID: 0

always

rpm

Always, from 0 to 30 the signal will be true when Above/Below 3000

B. Hoxha and H. Bach and H. Abbas and A. Dokhanchi and Y. Kobayashi and G. Fainekos, **Towards Formal Specification Visualization for Testing and Monitoring of Cyber-Physical Systems,** DIFTS 2014

ViSpec – Usability Study

Each user received ten tasks:

- To formalize a NL specification in automotive industry through ViSpec

Group I: Non-expert users

No experience in working with requirements.

20 subjects from the student community at ASU

Group 2: Expert users

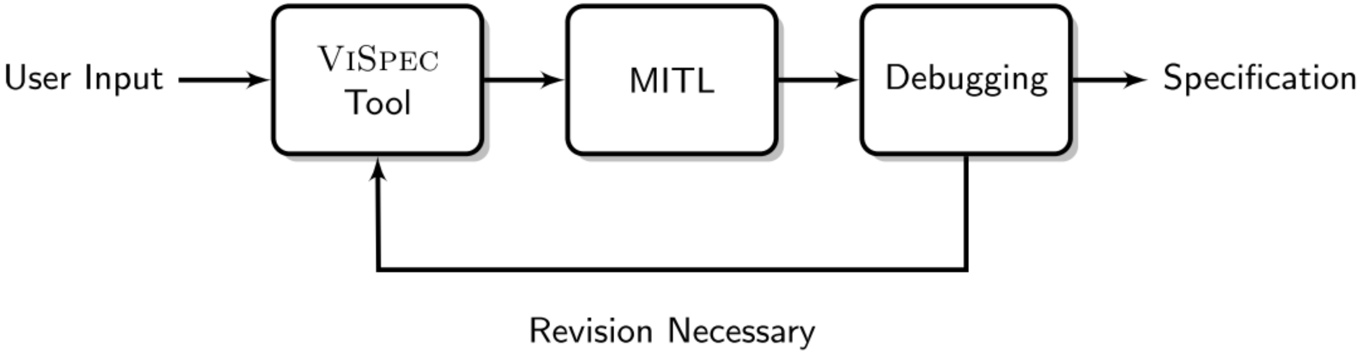
Experienced in working with requirements (not necessarily formal requirements)

10 subjects from the industry in the Phoenix area

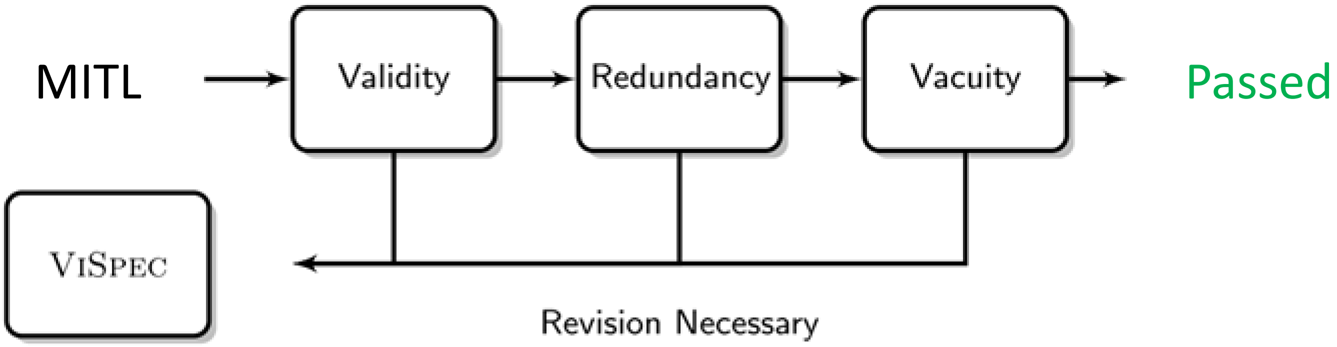
B. Hoxha, N. Mavridis and G. Fainekos, VISPEC: A graphical tool for easy elicitation of MTL requirements, IROS 2015

Debugging MITL Specification

Specification Elicitation Framework



3-Levels of Specification Debugging



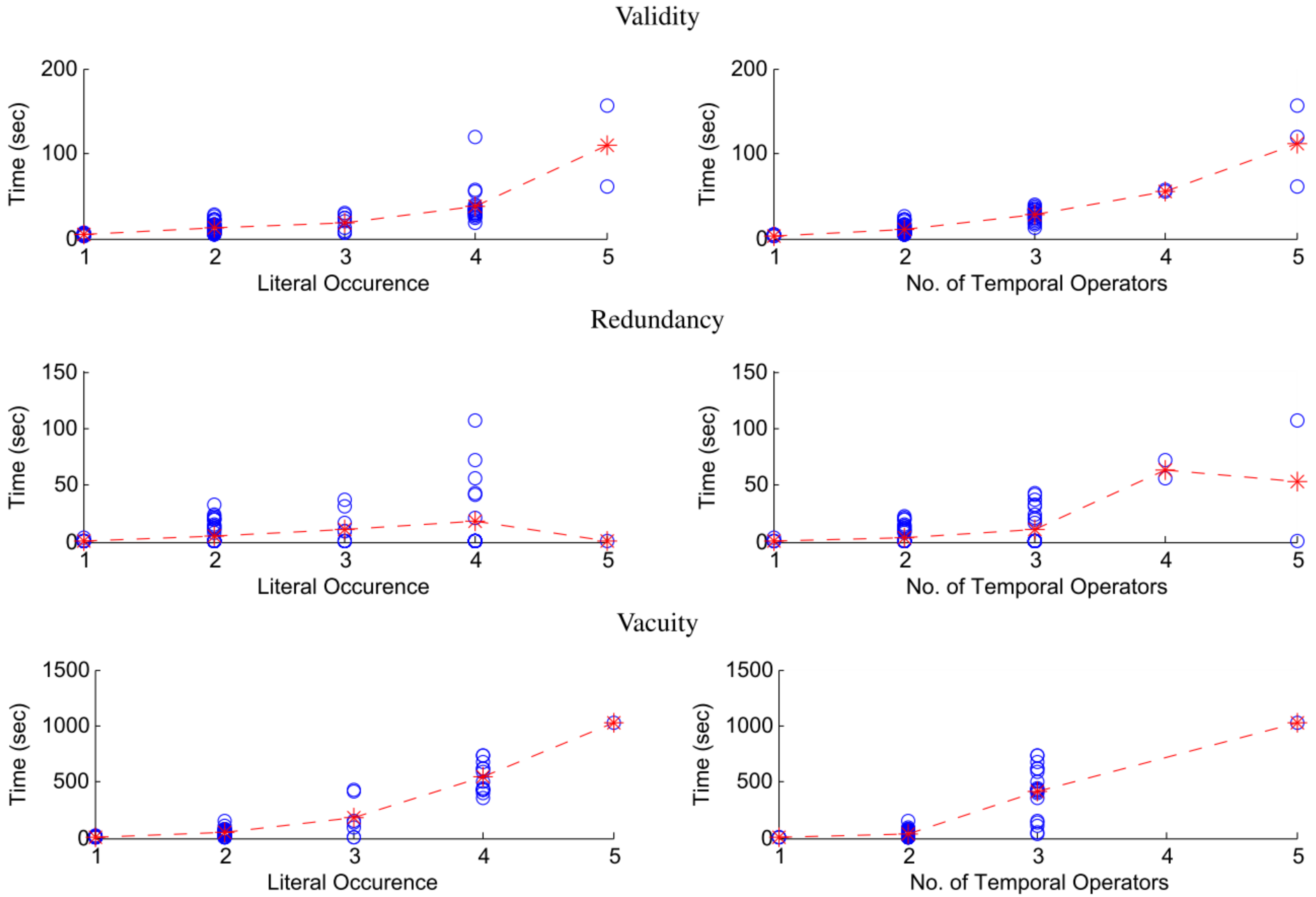
Problem Formulation

Given an MITL formula φ, find whether φ has any of the following logical issues:

- **Validity:** the specification is unsatisfiable or a tautology.
- **Redundancy:** the formula has redundant conjuncts.
- **Vacuity:** some subformulas do not contribute to the satisfiability of the formula.

A. Dokhanchi, B. Hoxha, and G. Fainekos, *Metric interval temporal logic specification elicitation and debugging*. MEMOCODE 2015

Runtime Overhead



WRAPPING UP

As seen in …

Embedded COMPUTING DESIGN

Industry-Nominated
Technology Breakthro
of NSF Industry/Unive
Cooperative Research C
2012

National Science Fou
Directorate for Engi
Division of Industrial Innovatio
Industry/University Cooperative R
4201 Wilson Bouleva
Arlington, Virgin
http://www.n
http://www.nsf.gov

Craig S. Scott, Editor
University of Washington School of Medicine

Marita Stevens Graube, Designer
Pixel Theory Inc.

Industry-Nominated
Technology Breakthroughs
of NSF Industry/University
Cooperative Research Center
2014

AUTOMOTIVE EMAG

ADAS

Improving automotive safety system effectiveness with model-based development

By Monique DeVoe, Managing Editor

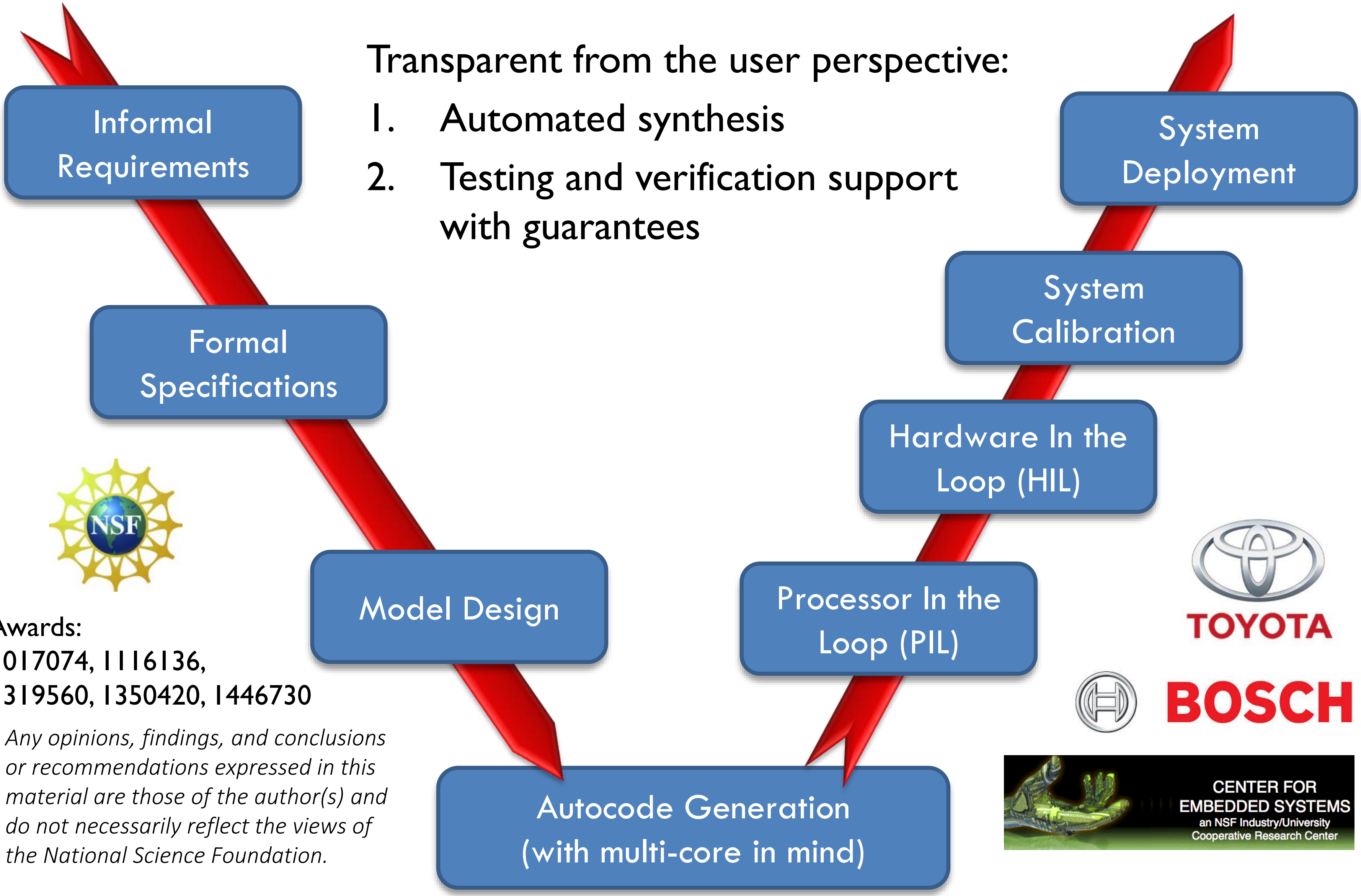
Embedded systems are controlling an increasing amount of the automotive experience, from infotainment systems and connectivity to drivers' portable devices to driver assist and vehicle safety/control features, making them smarter, but also more complex. Often contrary to the safety-critical nature of some embedded automotive systems are short development cycles; there isn't always time to make sure these systems are 100 percent bug free before they're deployed in vehicles. Arizona State University's Center for Embedded Systems (CES) is working on developing a software tool that can improve confidence in these designs.

An area of CES's research is based on the method of Model Based Development (MBD). In MBD, the system design and development always starts with a mathematical model of the system, with which system designers use to verify that the model satisfies the system's specifications. MBD has several advantages over traditional system development methods; particularly, by utilizing MBD practices, the main design choices are evaluated on the model of the system, removing the need for building costly prototypes.

Vision: *a complete theory for MBD for CPS*

Transparent from the user perspective:

1. Automated synthesis
2. Testing and verification support with guarantees


Informal Requirements

Formal Specifications

Model Design

Autocode Generation (with multi-core in mind)

Processor In the Loop (PIL)

Hardware In the Loop (HIL)

System Calibration

System Deployment

Awards:
1017074, 1116136,
1319560, 1350420, 1446730

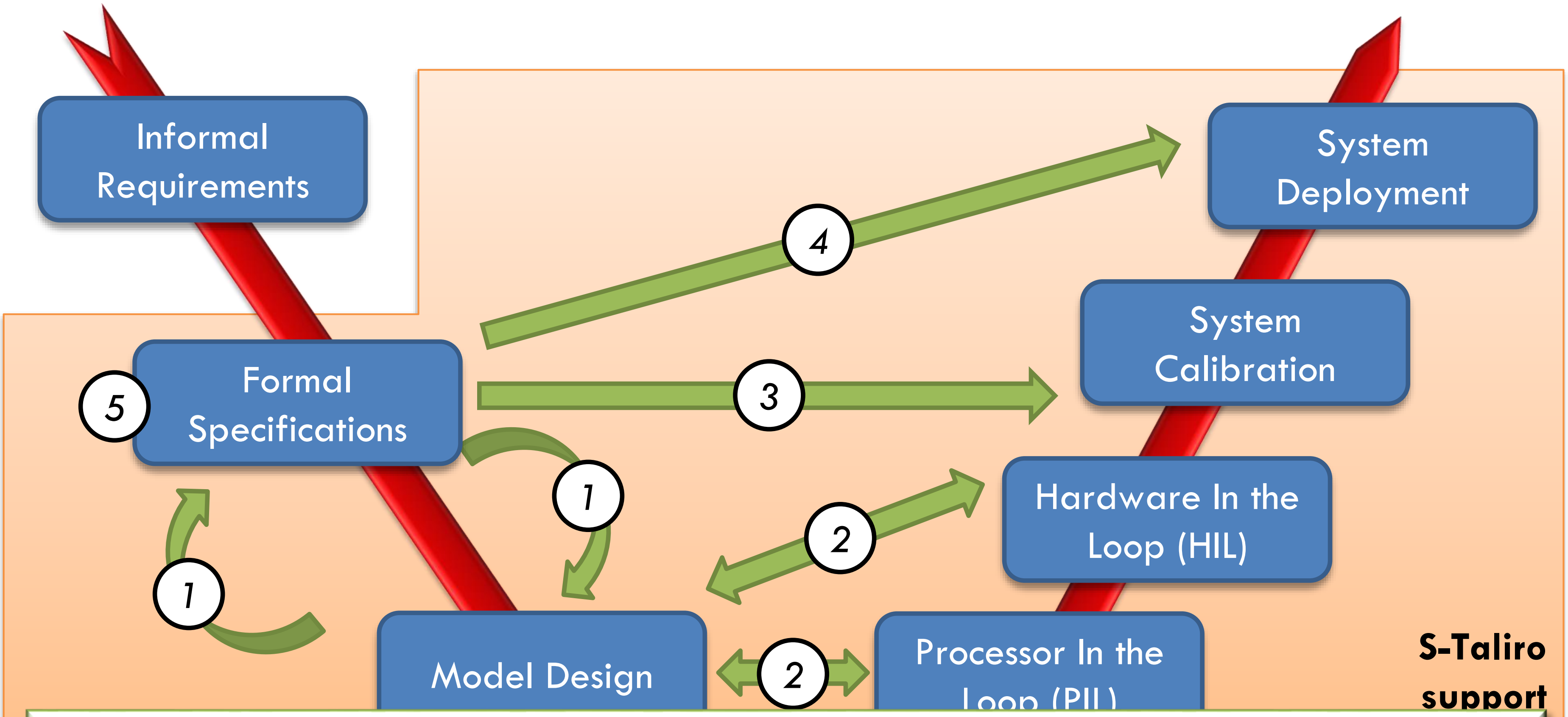
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

TOYOTA

BOSCH

CENTER FOR EMBEDDED SYSTEMS
an NSF Industry/University Cooperative Research Center

S-Taliro support in the V-process



1. Testing formal specifications and specification mining [TECS 2013, ICTSS 2012, …]
2. Conformance testing: models, HIL/PIL or tuned/calibrated model [MEMOCODE 2014]
3. Testing formal specifications on the HIL/PIL calibrated system [TECS 2013, …]
4. Runtime monitoring of formal requirements [RV 2014]
5. Specification visualization [IROS 2015] & Debugging [MEMOCODE 2015]

Acknowledgements

(Main contributors to the S-TaLiRo project)

Current Students

- Adel Dokhanchi – PhD
- Bardh Hoxha – PhD
- C. Erkan Tuncali – PhD
- Shakiba Yaghoubi – PhD

Former Students

- Houssam Abbas - PhD
- Y. Annapureddy - MS
- Rahul T. Srinivasa - MS
- Hengyi Yang – MS
- Hoang Bach – BS
- Jorge Mendoza – BS

Main collaborator

- CU, Boulder: S. Sankaranarayanan

Other collaborators

- ASU: Y. Kobayashi, Y-H Lee, H. Mittelmann
- NEC Labs: A. Gupta (now in Princeton), F. Ivancic (now in Google)
- RPI: Agung Julius
- Toyota: J. V. Deshmukh, J. Kapinski, K. Ueda, H. Yazarel (now in CareFusion), X. Jin

We build systems you can trust your life on!

NSF

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

BOSCH

MBD

TOYOTA TECHNICAL CENTER

Special Thanks: S. Vrudhula (ASU)

CENTER FOR EMBEDDED SYSTEMS
an NSF Industry/University Cooperative Research Center