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

Already demonstrated:
- Lane following & Active cruise control
- Fully autonomous driving
- ...
Embedded in: Automotive Systems

- 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]

[Continental Cooperation: The Cloud as sensor]
"A software error may prevent the transmission from downshifting, such as shifting from 5th to 4th gear when coasting," said NHTSA in its recalls summary of the problem. "This may result in decreased engine RPMs and possible engine stall, increasing the risk of a crash."

... the software that “allows the ECU to establish a ‘handshake’ with the engine is in error. The ECU monitors certain driving conditions, and when the engine is found to be out of tolerance, it triggers a fault code. When this happens, the ECU tries to find an optimal driving condition outside its prescribed tolerances, a rough idle or stalling situation ensues.”

... to update the software that controls the hybrid electric motor. Under certain circumstances, it is possible, according to the company, “for the electric motor to rotate in the direction opposite to that selected by the transmission.”

If the fault occurs, cruise control can only be disabled by turning off the ignition while driving - which also disables power steering. Braking or pressing the cancel button will not work.

Many more ...
How serious this problem is?

Software-Related Vehicle Recalls

- Number of recalls
- Number of affected vehicles

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

The same holds for the medical device industry!

Is it always a software error?!?

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

Tesla cars: Clearly a marvel of modern engineering!

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

A Tesla somewhere in Switzerland

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

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.
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.

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!

Enginuity™ Modeling Approach

**Orifice Flow**
Isentropic Flow Model

\[ m_1 = A \frac{p}{\sqrt{RT}} \psi \]

\[ \psi = \sqrt{\ldots [\max(...)-\max(...)]} \]

\[ \vdots \]

**Intake and Exhaust Plenum**
Mass Conservation

\[ m_2 = \begin{cases} > 0 & \text{if } p_1 > p_2 \\ = 0 & \text{if } p_1 = p_2 \\ < 0 & \text{if } p_1 < p_2 \end{cases} \]

**Combustion Chamber**
Energy Conservation
Heat Transfer
Heat Release
Ignition Delay
Fuel Injection Dynamics

...
Develop controllers and generate code

Simplify model:
\[ \dot{x} = Ax + Bu \]

or
\[ \dot{x} = f(x, u), \#(x) \ll 60 \]

Design control laws
e.g. idle speed control

Engine dynamics

Alternative path:
PID tuning

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?

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_Lin(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:


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**
- **Model Simplification**
  - What effects we can neglect?
- **Physical Phenomenon or device to be studied**
- **Idealized System**
- **Mathematical Model**
  - Expected to be a dynamic model
- **Model Simplification**
- **Mathematical Model**
  - Solutions to Math model: Analytical or Numerical
- **Model Validation**
- **Performance assessment**
  - Aid in redesign to satisfy specs

- **Analysis**
  - What are the properties of interest?
  - 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 \( \dot{T} = 0.1 T - 50 \)
With rod 1 \( \dot{T} = 0.1 T - 56 \)
With rod 2 \( \dot{T} = 0.1 T - 60 \)

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

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

**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.

$$\text{System} = \text{Train} \parallel \text{Gate} \parallel \text{Controller}$$
Train model

- far: $-50 \leq x \leq -40$, $x \geq 1000$
- near: $-50 \leq x \leq -30$, $x \geq 0$
- past: $-50 \leq x \leq -30$, $x \geq -100$

$x \geq 2000$

$x = 1000$

approach

$x = 0$

exit

$x = -100 \rightarrow x' \in [2000, \infty)$
Gate model

- **Raising**
  - $\dot{\theta} = 9$
  - $\theta \leq 90$
  - Transition to **Open** when $\theta = 90$

- **Open**
  - $\dot{\theta} = 0$
  - $\theta = 90$

- **Lowering**
  - $\dot{\theta} = -9$
  - $\theta \geq 0$
  - Transition to **Closed** when $\theta = 0$

- **Closed**
  - $\dot{\theta} = 0$
  - $\theta = 0$

Actions:
- **Raise**
- **Lower**

States:
- **Open**
- **Closed**
- **Raising**
- **Lowering**
Controller model

\[ \begin{align*}
\dot{y} &= 1 \\
y &\leq d
\end{align*} \]

\[ \begin{align*}
\dot{y} &= 1 \\
\text{true}
\end{align*} \]

\[ \begin{align*}
\dot{y} &= 1 \\
\text{true}
\end{align*} \]

\[ \begin{align*}
\dot{y} &= 1 \\
y &\leq d
\end{align*} \]
Synchronized transitions

\[
x \geq 2000
\]

\[
\begin{aligned}
&\text{far} \quad -50 \leq x \leq -40 \\
&\quad x \geq 1000
\end{aligned}
\]  
\[
\begin{aligned}
&\text{near} \quad -50 \leq x \leq -30 \\
&\quad x \geq 0
\end{aligned}
\]  
\[
\begin{aligned}
&\text{past} \quad -50 \leq x \leq -30 \\
&\quad x \geq -100
\end{aligned}
\]

\[
\begin{aligned}
&\text{approach} \\
&\text{exit}
\end{aligned}
\]

\[
x = -100 \rightarrow x' \in [2000, \infty)
\]

\[
y := 0
\]

\[
\begin{aligned}
&\text{Going to lower} \\
&y = 1 \\
&y \leq d
\end{aligned}
\]

\[
\begin{aligned}
&\text{idle} \\
&\dot{y} = 1 \\
&\text{true}
\end{aligned}
\]

\[
\begin{aligned}
&\text{Going to raise} \\
&y = 1 \\
&y \leq d
\end{aligned}
\]

\[
\begin{aligned}
&\text{approach} \\
&\text{lower}
\end{aligned}
\]

\[
\begin{aligned}
&\text{true} \\
&\text{raise}
\end{aligned}
\]

\[
\begin{aligned}
&\text{approach} \\
&\text{exit}
\end{aligned}
\]
Verifying the controller

System = Train || Gate || Controller

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

Parametric verification: 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 to 4th gear when coasting." said NHTSA in its recalls summary of the problem. "This may result in decreased engine RPMs and possible engine stall, increasing the risk of a crash."

- ... the software that “allows the ECU to establish a ‘handshake’ with the engine is in error. The ECU monitors certain driving conditions and when the engine is found to be out of tolerance, the software triggers a fault code. As the engine tries to find an optimal driving condition outside its prescribed tolerances, a rough idle or stalling situation ensues.”

- ... to update the software that controls the hybrid electric motor. Under certain circumstances, it is possible, according to the company, "...for the electric motor to rotate in the direction opposite to that selected by the transmission."

- If the fault occurs, cruise control can only be disabled by turning of the ignition while driving - which would mean a loss of some control and in many cars also disables power steering. Braking or pressing the cancel button will not work.

- ...
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).

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 $s_1$ 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

\[ \phi ::= T \mid p \mid \neg \phi \mid \phi_1 \lor \phi_2 \mid G_I \phi \mid F_I \phi \mid \phi_1 U_I \phi_2 \]

**Ga** - always a

**F_{[1,3]}a** - eventually a

**a U b** - a until b

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

\[ \phi \text{ now} \]

\[ \text{time} \]
Possible formalizations?

\[ G\left(\text{Orange} \land P_{[0,y]} \text{ Black}\right) \rightarrow F_{[0,x]}\left( (s2 \text{ in red}) \cup G \left(s1 \text{ in blue}\right) \right) \]
\[ G\left(\text{Orange} \land P_{[0,y]} \text{ Black}\right) \rightarrow G_{[x,\infty]}\left( (s2 \text{ in red}) \lor G \left(s1 \text{ in blue}\right) \right) \]
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]
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

<table>
<thead>
<tr>
<th>Generated input</th>
<th>Defect condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gas pedal[%]</td>
<td>① Specific logic on</td>
</tr>
<tr>
<td>Brake[%]</td>
<td>② Engine revolution around 4000rpm</td>
</tr>
<tr>
<td>Shift{P,N,D}</td>
<td>③ Satisfy ①,② and specific accelerator operation</td>
</tr>
<tr>
<td>Water temp[℃]</td>
<td></td>
</tr>
<tr>
<td>Air temp[℃]</td>
<td></td>
</tr>
<tr>
<td>Air pressure[kPa]</td>
<td></td>
</tr>
<tr>
<td>Air conditioner SW</td>
<td></td>
</tr>
</tbody>
</table>

Tried 6 large-scale models, 5 models were falsified.
(Past defect case, intentional defect by logic developer)

S-Taliro could generate the complicated scenario including the defect

Figure Generated signals automatically

Shunsuke Kobuna
WHAT IS THE CHALLENGE IN FORMALIZING REQUIREMENTS?
Student homework (Graduate class): Formalizing requirements

- Traditional section of the class (31 students)

<table>
<thead>
<tr>
<th>Problem difficulty</th>
<th>Very Easy</th>
<th>Very Easy</th>
<th>Easy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>9.4</td>
<td>9.6</td>
<td>7.2</td>
</tr>
<tr>
<td>Median</td>
<td>10.0</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>Max</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Min</td>
<td>7.1</td>
<td>6.7</td>
<td>4</td>
</tr>
</tbody>
</table>

- On-line section (10 professional* students)

<table>
<thead>
<tr>
<th>Problem difficulty</th>
<th>Very Easy</th>
<th>Very Easy</th>
<th>Easy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>7.7</td>
<td>7.7</td>
<td>6.8</td>
</tr>
<tr>
<td>Median</td>
<td>8.6</td>
<td>7.8</td>
<td>6.0</td>
</tr>
<tr>
<td>Max</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Min</td>
<td>4.3</td>
<td>4.4</td>
<td>0</td>
</tr>
</tbody>
</table>

* 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:

\[
\varphi = \Diamond_{[0,30]}(\quad (v > 100) \implies \Box_{[0,20]}(v > 100))
\]

\(\varphi\) is a tautology

- \((v > 100) = \bot\) at any time in \([0,30]\)
  
  \[((v > 100) \implies \Box_{[0,20]}(v > 100)) = T\]

- \((v > 100) = T\) for all the time in \([0,30]\)
  
  \[\Box_{[0,20]}(v > 100) = T\text{ between } [0,10]\]

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

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:

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

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

User Input → ViSPEC Tool → MITL → Debugging → Specification

Revision Necessary

3-Levels of Specification Debugging

MITL → Validity → Redundancy → Vacuity → Passed

Revision Necessary
Problem Formulation

Given an MITL formula $\varphi$, find whether $\varphi$ 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.

Runtime Overhead

Validity

Redundancy

Vacuity
WRAPPING UP
As seen in ...
Vision: a complete theory for MBD for CPS

Transparent from the user perspective:
1. Automated synthesis
2. Testing and verification support with guarantees

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.
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]
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

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.

We build systems you can trust your life on!

Special Thanks: S. Vrudhula (ASU)