Meeting A Powertrain Verification Challenge

Parasara **Sridhar** Duggirala,

Chuchu Fan,

Sayan Mitra, and

Mahesh Viswanathan

ILLINOIS
1867
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Powertrain Control Systems

- Fuel control and transmission subsystem
  - Software control: increasing complexity (100M LOC)
  - Constraints: Emissions, Efficiency, etc.
  - Strict performance requirements
  - Early bug detection using formal methods

Powertrain Control Systems

- Fuel control and transmission subsystem
  - Software control: increasing complexity (100M LOC)
  - Constraints: Emissions, Efficiency, etc.
  - Strict performance requirements
  - Early bug detection using formal methods

- Powertrain control benchmarks from Toyota Jin et.al. [HSCC'14]
- Complexity "*similar*" to industrial systems
- Benchmark tool/challenge problems for academic research

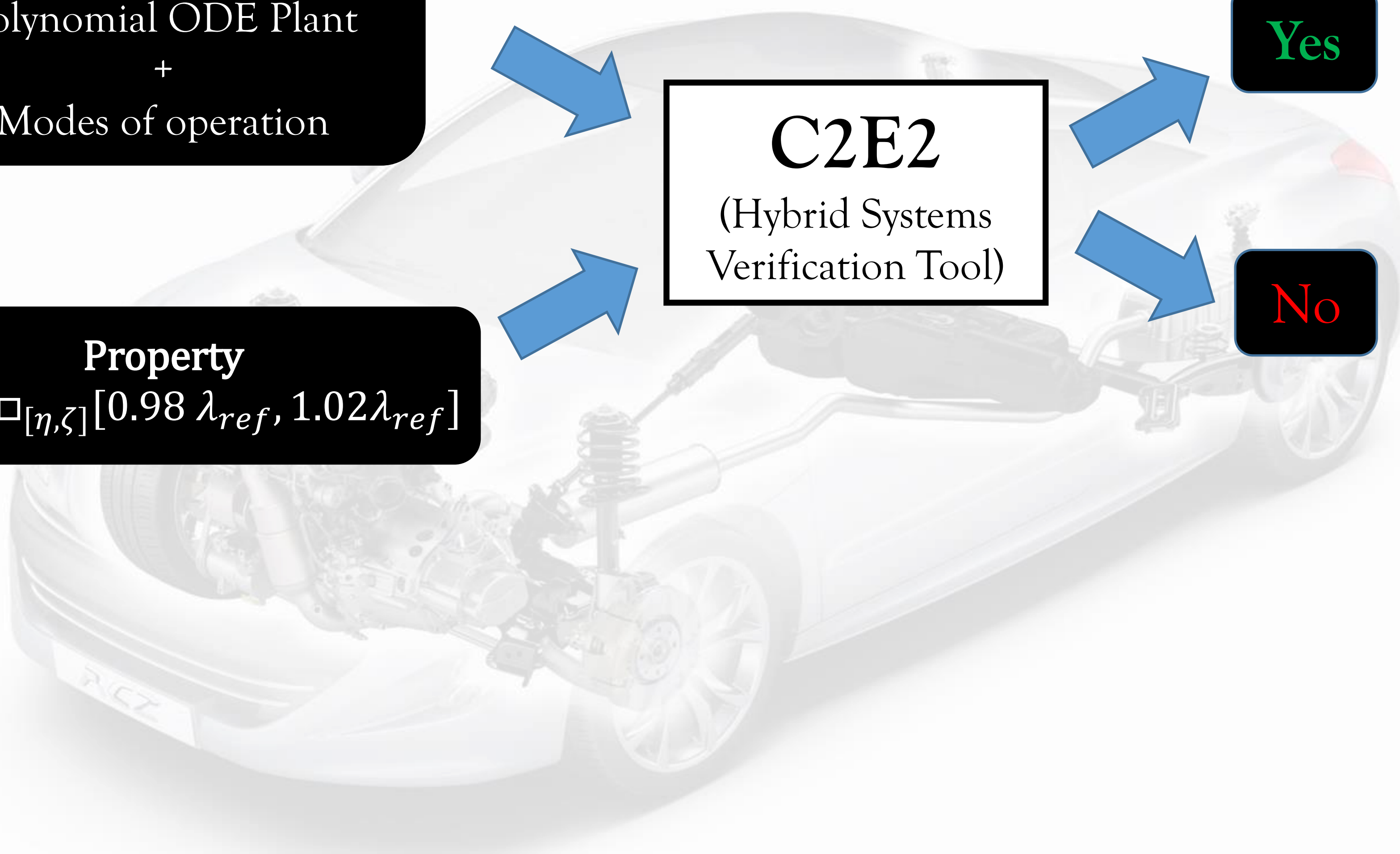
Powertrain Control Systems

- Fuel control and transmission subsystem
  - Software control: increasing complexity (100M LOC)
  - Constraints: Emissions, Efficiency, etc.
  - Strict performance requirements
  - Early bug detection using formal methods

- Powertrain control benchmarks from Toyota Jin et.al. [HSCC'14]
- Complexity "*similar*" to industrial systems
- Benchmark tool/challenge problems for academic research

This paper: Verifying one of the models in the powertrain control benchmark

Verifying Powertrain Control System (Challenges)



Hybrid Systems Model
Polynomial ODE Plant
+
Modes of operation

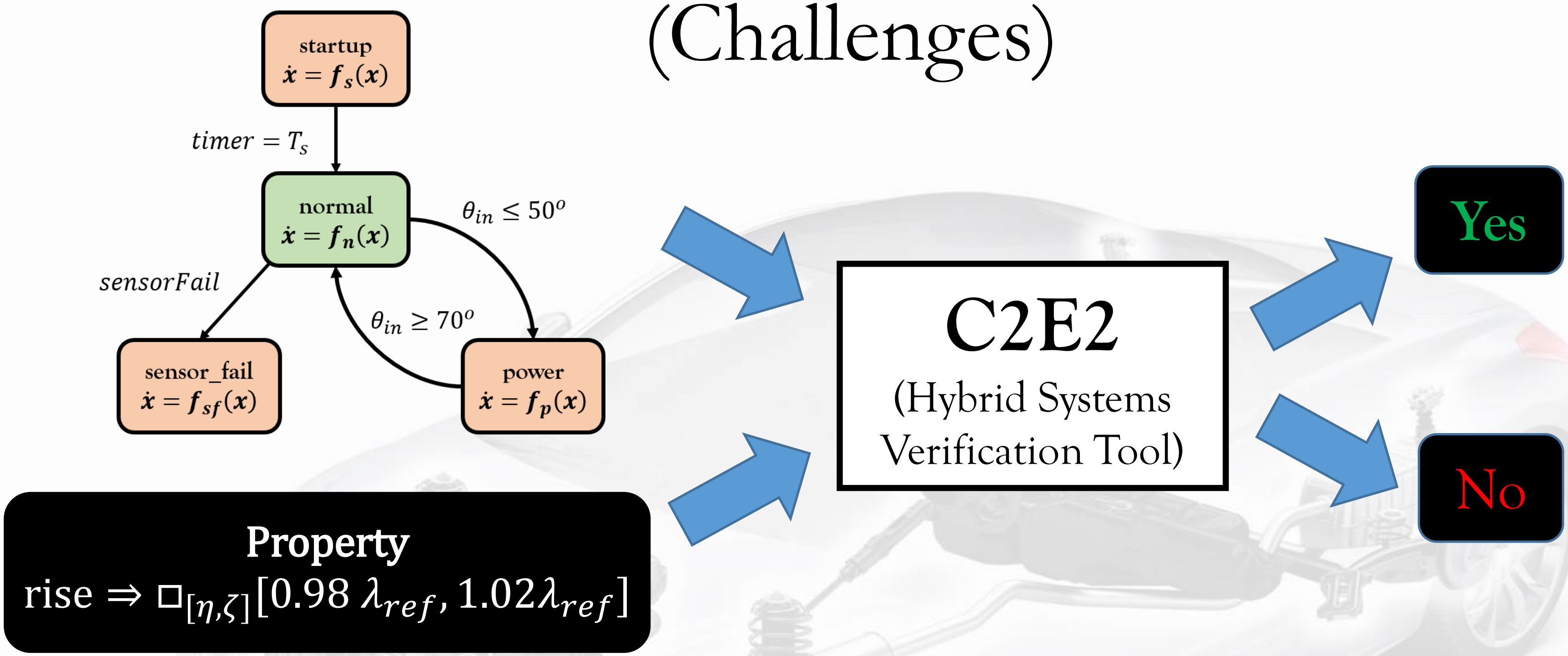
Property
rise $\Rightarrow \square_{[\eta,\zeta]}[0.98\ \lambda_{ref}, 1.02\lambda_{ref}]$

C2E2
(Hybrid Systems Verification Tool)

Yes

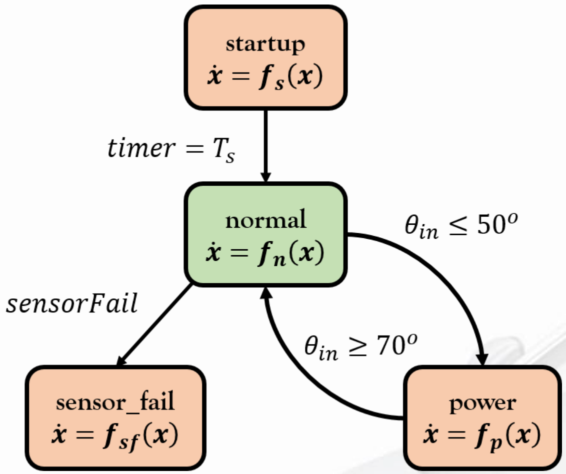
No

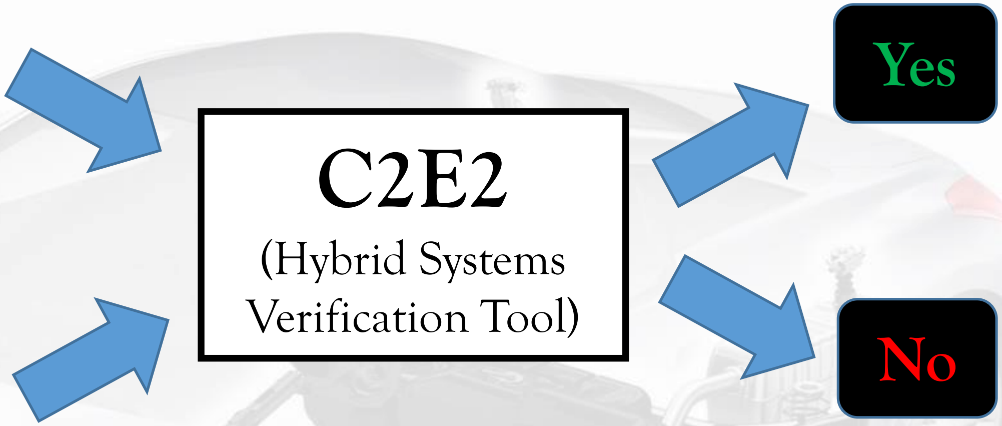
Verifying Powertrain Control System (Challenges)



- Hybrid systems verification
  - Undecidable in general [simple continuous dynamics $\dot{x} = 1, \dot{y} = 2$]

Verifying Powertrain Control System (Challenges)



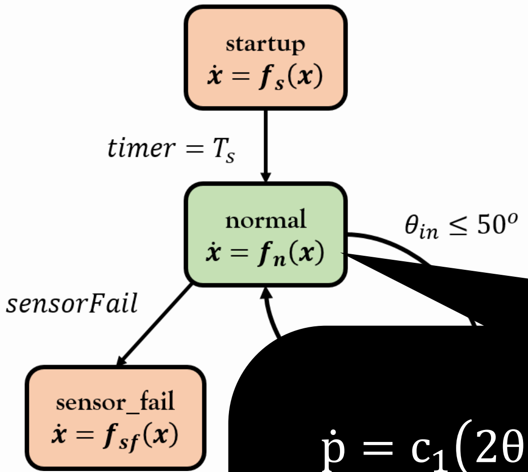


Property

rise $\Rightarrow \square_{[\eta,\zeta]}[0.98\,\lambda_{ref}, 1.02\lambda_{ref}]$

- Hybrid systems verification
  - Undecidable in general [simple continuous dynamics $\dot{x} = 1, \dot{y} = 2$]
  - Nonlinear Ordinary Diff. Eqns. – scalability problems

Verifying Powertrain Control System (Challenges)



Yes

Pro

rise ⇒ $\square_{[\eta,\zeta]}[0.$

$$\dot{p} = c_1\big(2\theta(c_{20}p^2 + c_{21}p + c_{22}) - c_{12}(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2)\big)$$
$$\dot{\lambda} = c_{26}(c_{15} + c_{16}c_{25}F_c + c_{17}c_{25}^2F_c^2 + c_{18}\dot{m}_c + c_{19}\dot{m}_c c_{25}F_c - \lambda)$$
$$\dot{p_e} = c_1\big(2c_{23}\theta(c_{20}p^2 + c_{21}p + c_{22}) - (c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2)\big)$$
$$\dot{i} = c_{14}(c_{24}\lambda - c_{11})$$

where

$$F_c = \frac{1}{c_{11}}(1 + i + c_{13}(c_{24}\lambda - c_{11}))(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2)$$
$$\dot{m}_c = c_{12}(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2)$$

- Hybrid systems verification
  - Undecidable in general [simple continuous dynamics $\dot{x} = 1, \dot{y} = 2$]
  - Nonlinear Ordinary Diff. Eqns. – scalability problems

Outline

- ✓ Motivation & Challenges
- Powertrain Benchmark
- Specification
- Simulation Based Verification Technique
- Engineering
- Verification Results
- Conclusions and Future Work

Powertrain Systems Benchmark (previous work)

- Falsification techniques
  S-Taliro Annpureddy et.al.[TACAS'11], Breach Donze et.al.[CAV'10].
- Requirement mining (also found bugs) Jin et.al.[HSCC'13].
- Simulation guided Lyapunov analysis Balkan et.al.[ICC'15], and more …

Model I

Delay Differential Equations
\+
Lookup Tables
\+
Hierarchical Components

→

Model II

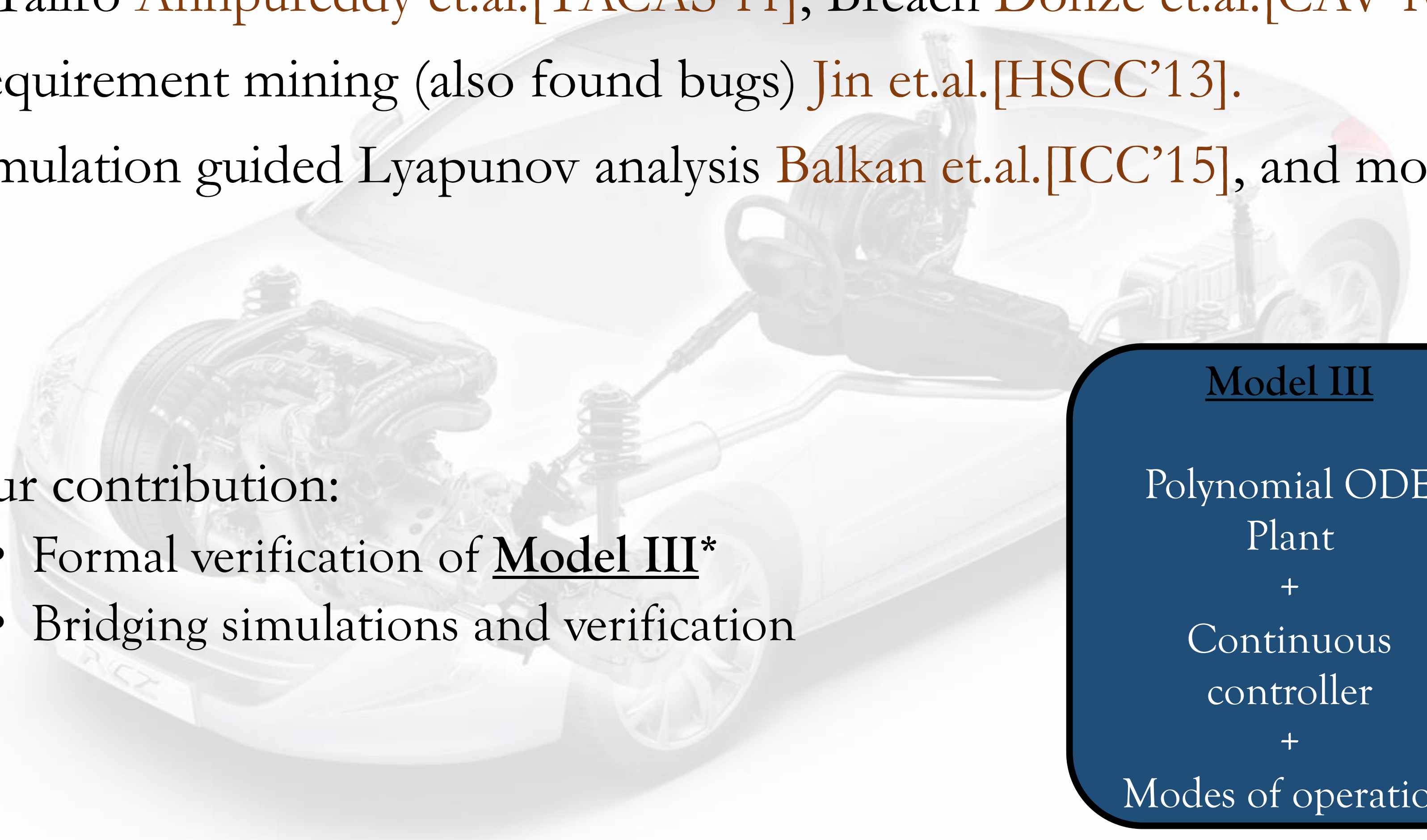
Nonlinear ODE Plant
(Non – polynomial)
\+
Discrete update control software

→

Model III

Polynomial ODE Plant
\+
Continuous controller
\+
Modes of operation

Powertrain Systems Benchmark (previous work)



- Falsification techniques
  S-Taliro Annpureddy et.al.[TACAS'11], Breach Donze et.al.[CAV'10].
- Requirement mining (also found bugs) Jin et.al.[HSCC'13].
- Simulation guided Lyapunov analysis Balkan et.al.[ICC'15], and more …

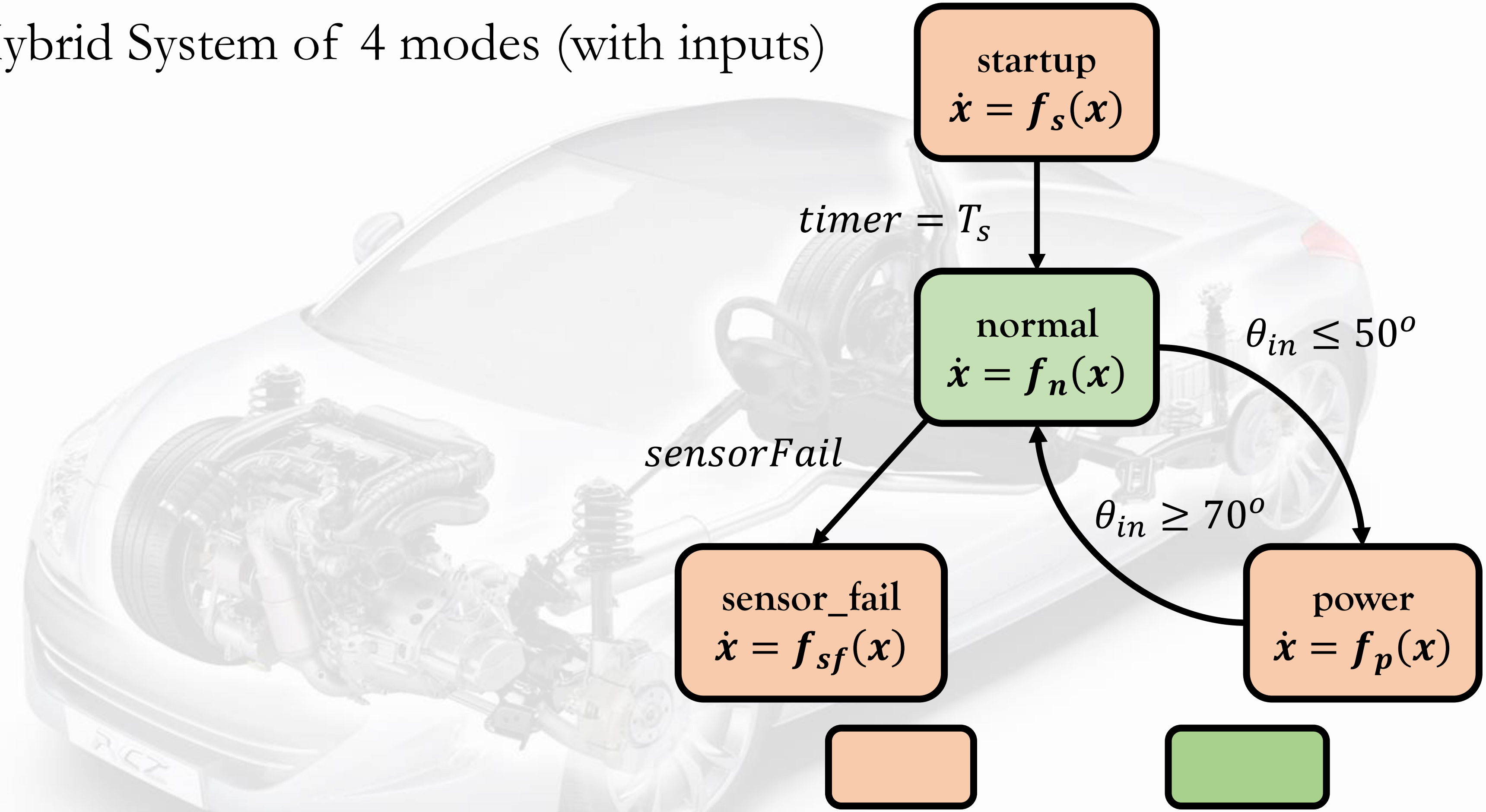
- Our contribution:
  - Formal verification of **Model III***
  - Bridging simulations and verification

Model III

Polynomial ODE Plant
+
Continuous controller
+
Modes of operation

Powertrain Model
(Model III)

- Hybrid System of 4 modes (with inputs)



- ❑ No Feedback Control
- ❑ Open Loop mode, feedforward estimator

- ❑ Feedback Control
- ❑ Closed-loop mode, feedback PI control + feedforward estimator

Powertrain Model

(Model III)

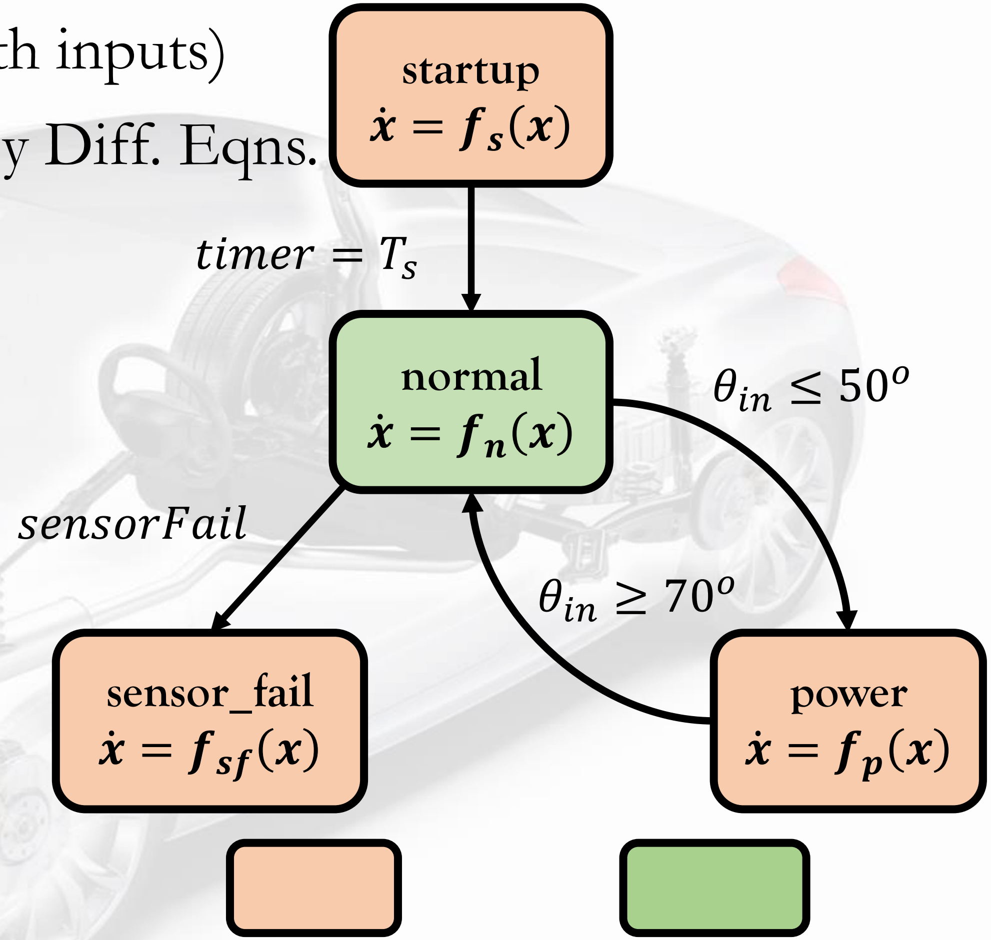
- Hybrid System of 4 modes (with inputs)
- Real valued variables – Ordinary Diff. Eqns.

λ – Air/fuel ratio

p – Intake manifold pressure

p_e – Estimate of p

i – PI control variable



- ❑ No Feedback Control
- ❑ Open Loop mode, feedforward estimator

- ❑ Feedback Control
- ❑ Closed-loop mode, feedback PI control + feedforward estimator

Powertrain Model
(Model III)

- Hybrid System of 4 modes (with inputs)
- Real valued variables – Ordinary Diff. Eqns.

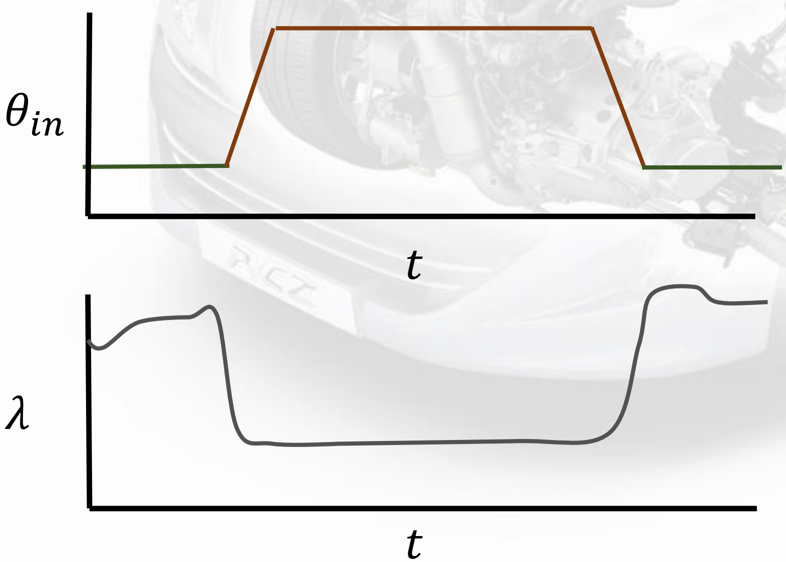
λ – Air/fuel ratio

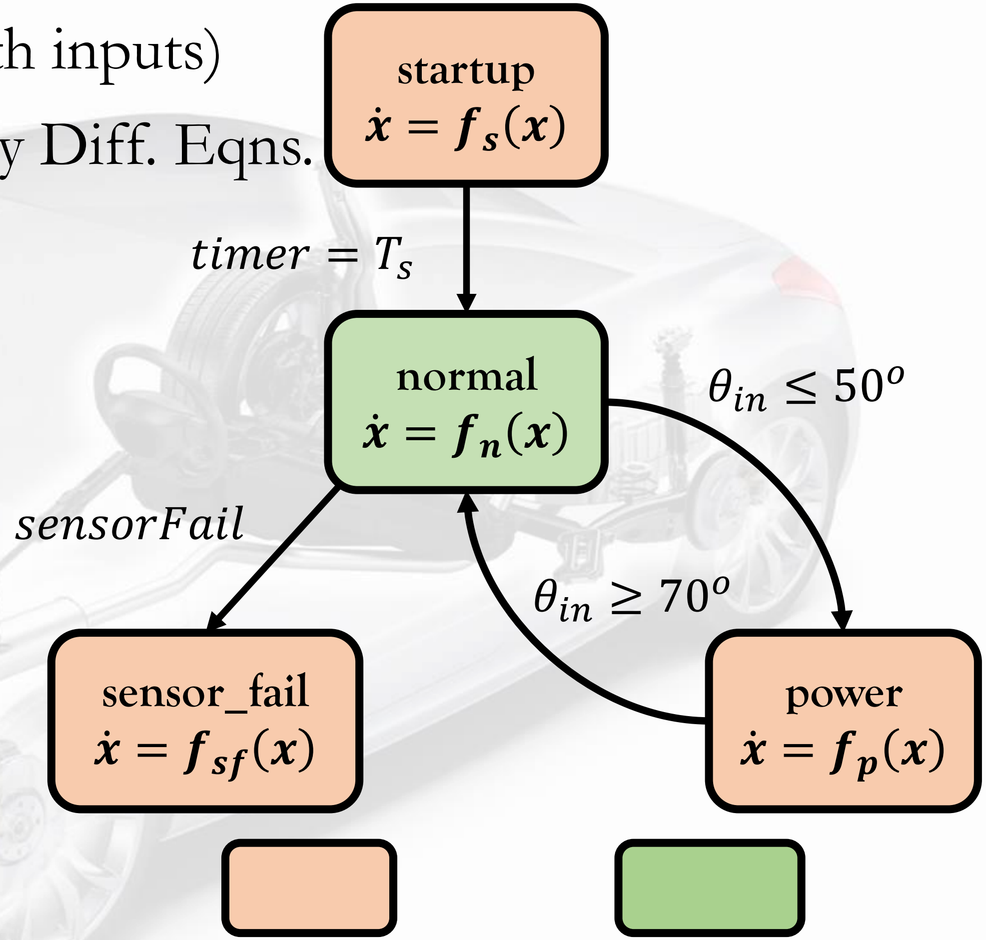
p – Intake manifold pressure

p_e – Estimate of p

i – PI control variable

- Transitions – input signal θ_{in}



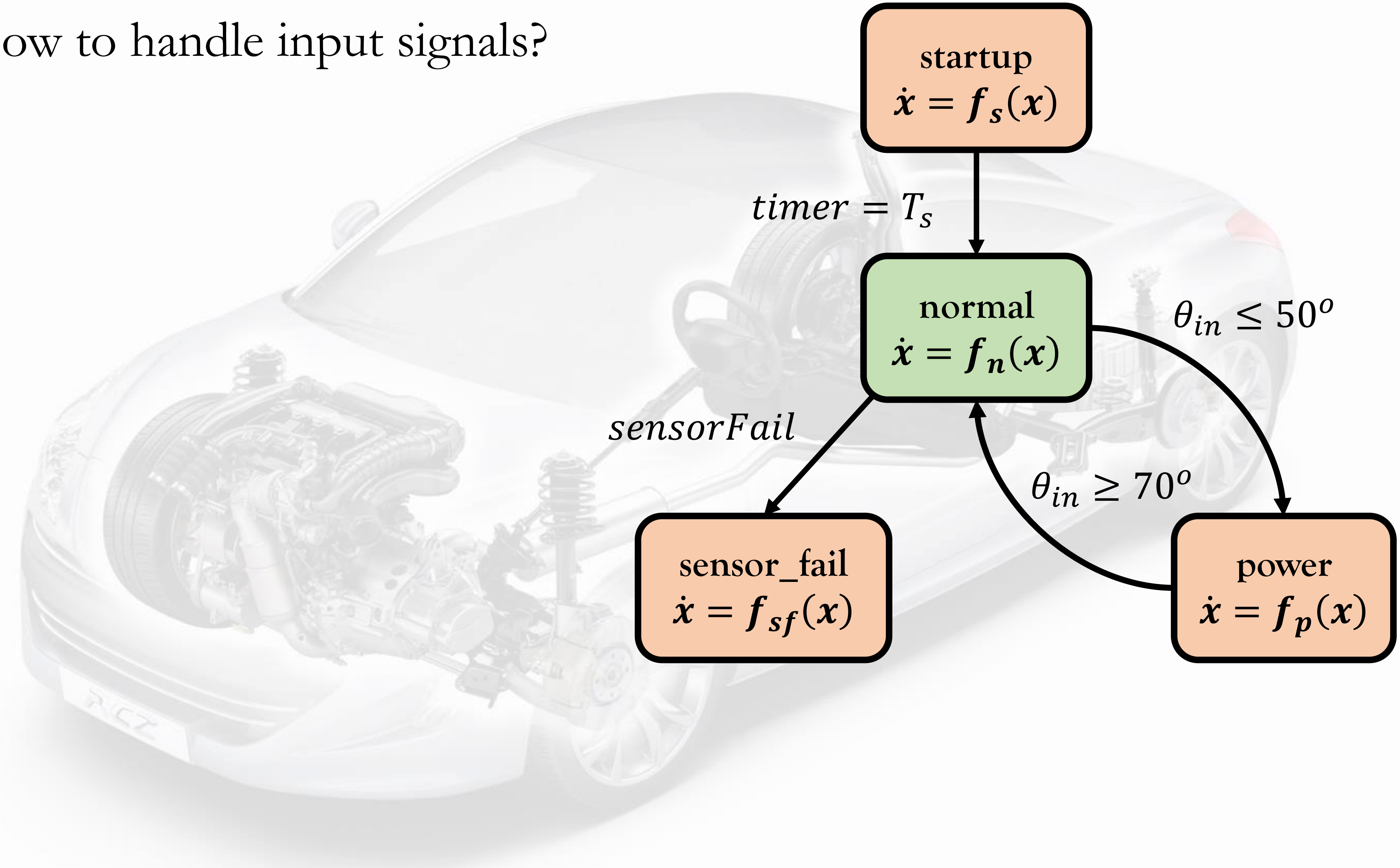


- ❑ No Feedback Control
- ❑ Open Loop mode, feedforward estimator

- ❑ Feedback Control
- ❑ Closed-loop mode, feedback PI control + feedforward estimator

Powertrain Model
(Challenges)

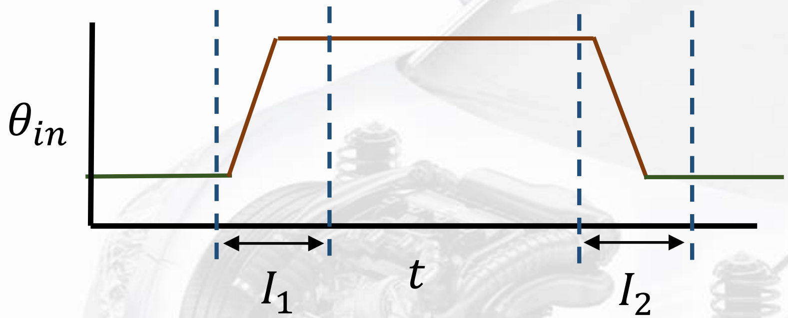
- How to handle input signals?

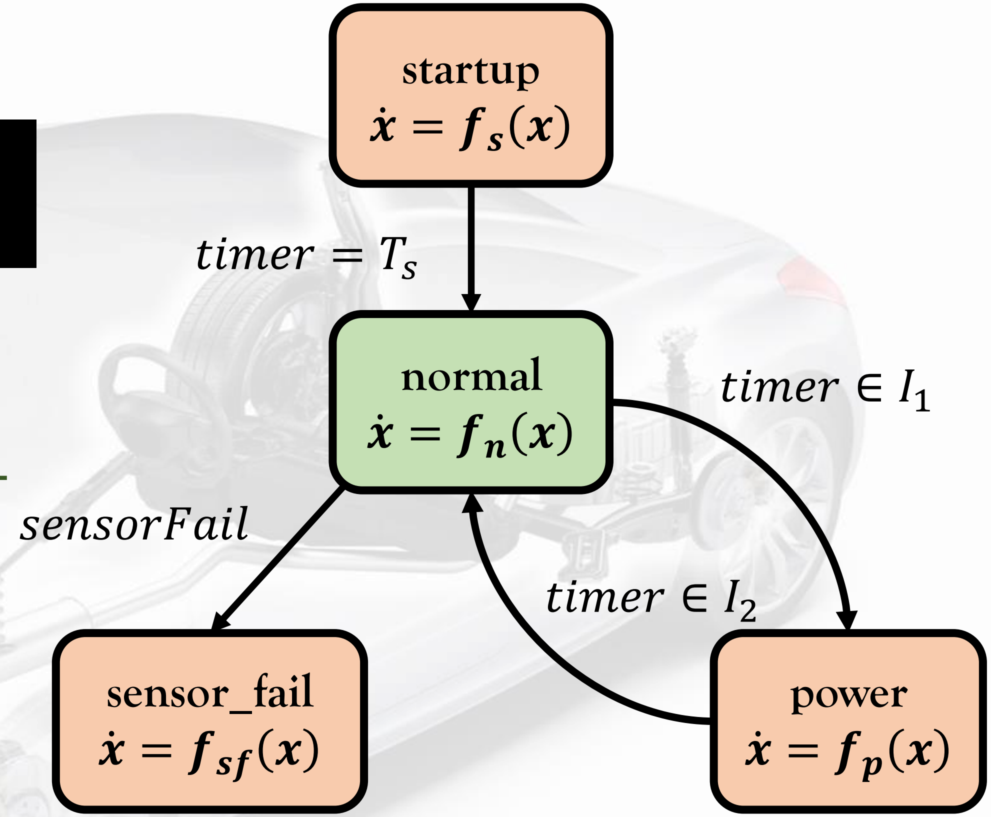


Powertrain Model
(Challenges)

- How to handle input signals?

Consider family of input signals θ_{in} and construct closed hybrid system

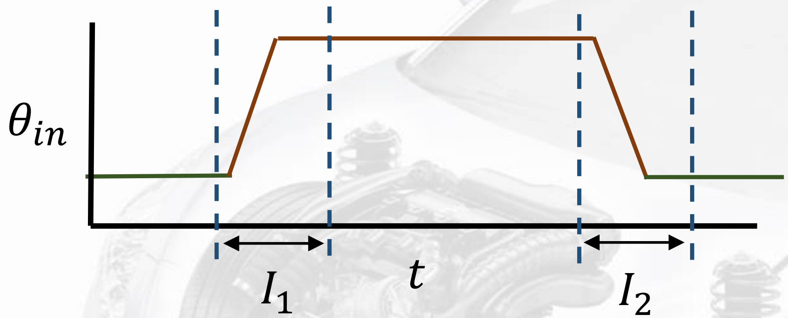


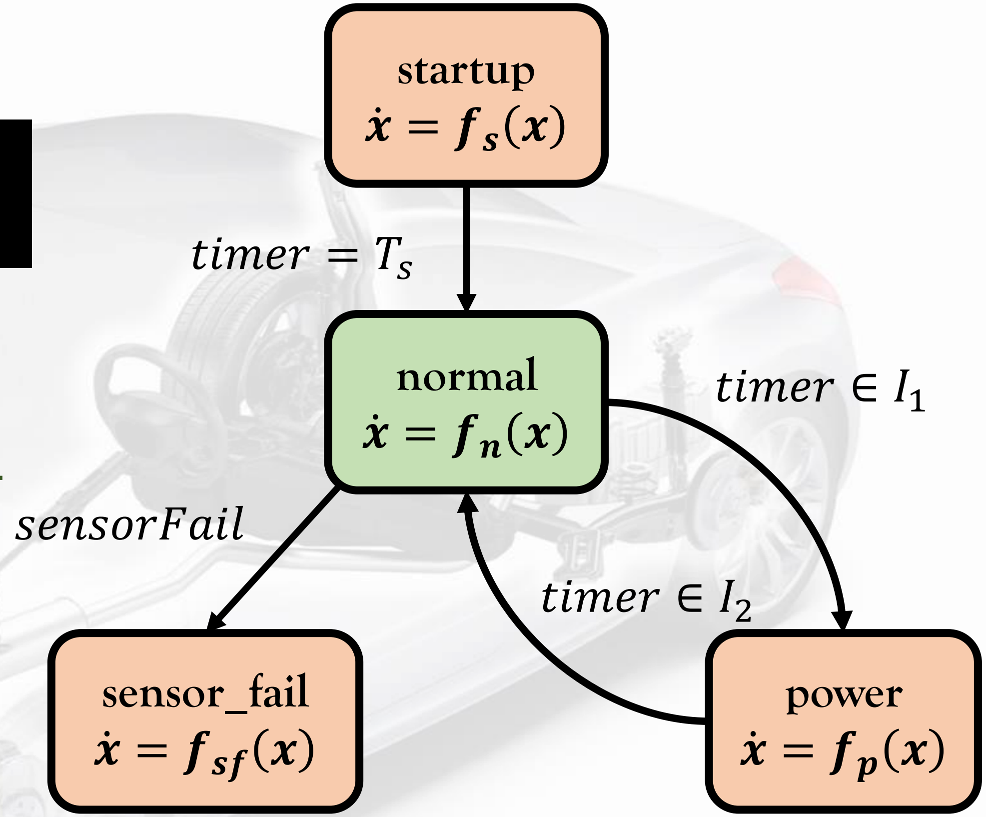


Powertrain Model
(Challenges)

- How to handle input signals?

Consider family of input signals θ_{in} and construct closed hybrid system





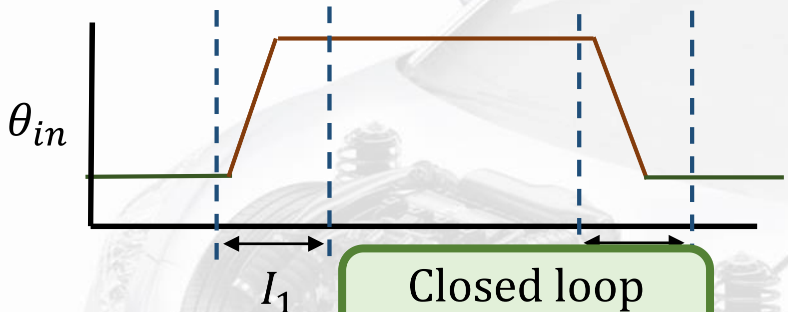
- Nonlinearity of ODE

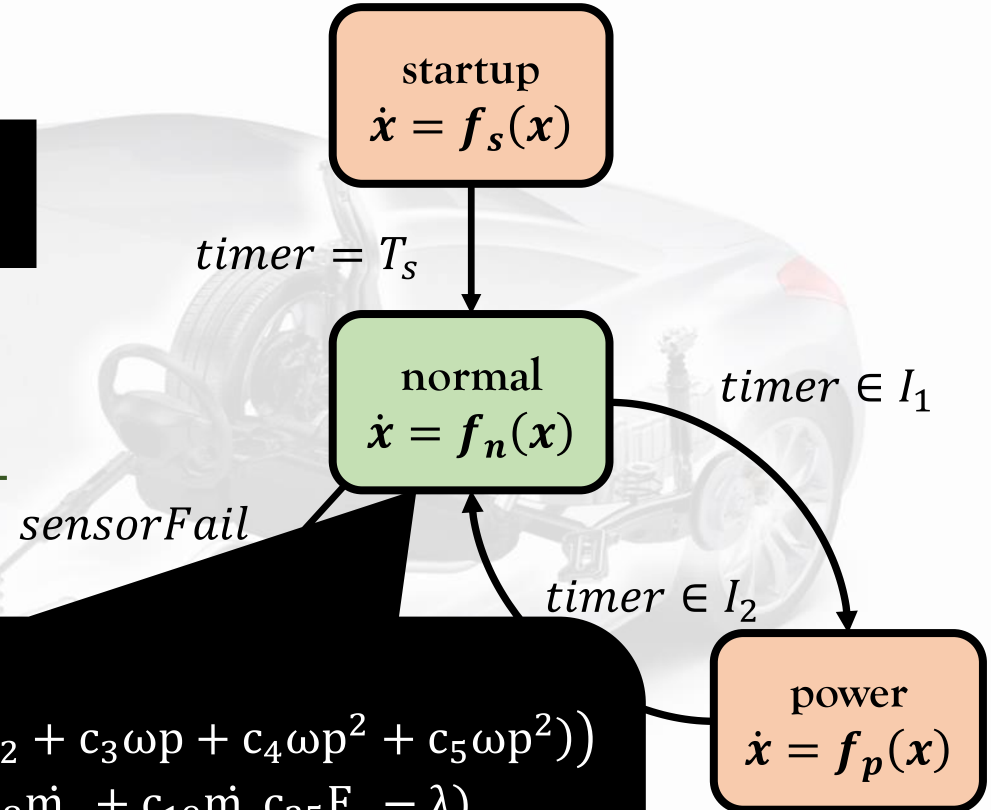
Powertrain Model

(Challenges)

- How to handle input signals?

Consider family of input signals θ_{in} and construct closed hybrid system





Closed loop Dynamics

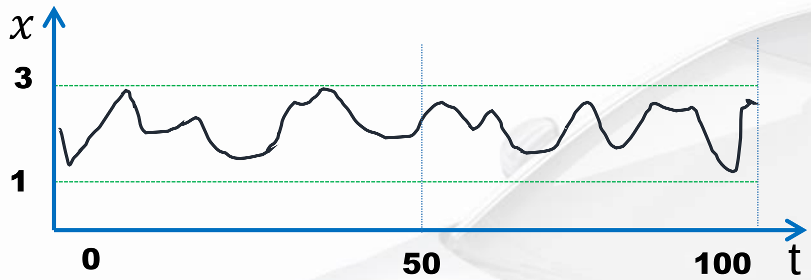
$$\dot{p} = c_1\left(2\theta(c_{20}p^2 + c_{21}p + c_{22}) - c_{12}(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2)\right)$$
$$\dot{\lambda} = c_{26}(c_{15} + c_{16}c_{25}F_c + c_{17}c_{25}^2F_c^2 + c_{18}\dot{m}_c + c_{19}\dot{m}_c c_{25}F_c - \lambda)$$
$$\dot{p_e} = c_1\left(2c_{23}\theta(c_{20}p^2 + c_{21}p + c_{22}) - (c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2)\right)$$
$$i = c_{14}(c_{24}\lambda - c_{11})$$

where

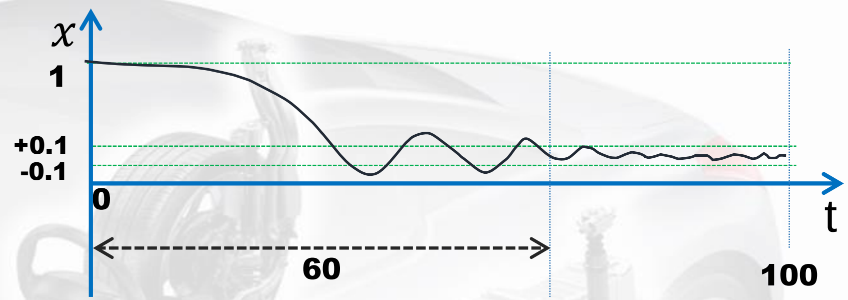
$$F_c = \frac{1}{c_{11}}(1 + i + c_{13}(c_{24}\lambda - c_{11}))(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2)$$
$$\dot{m}_c = c_{12}(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2)$$

Powertrain Specification

- Signal Temporal Logic: temporal specification for signals



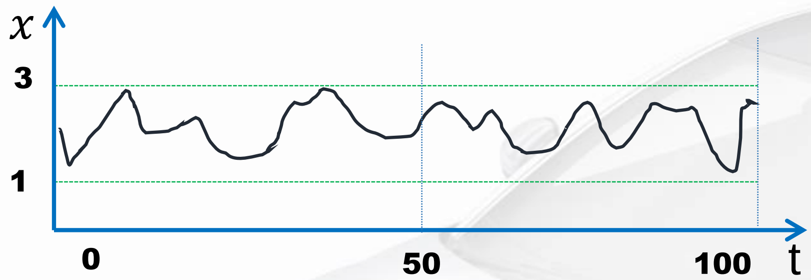
$\Box_{[0,100]}\, x \in [1,3]$

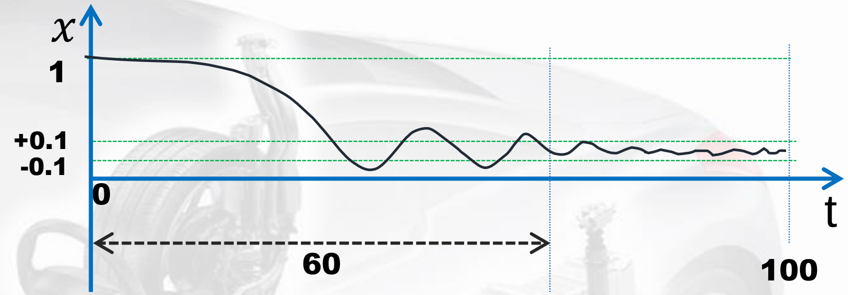


$\Box_{[60,100]}\, |x| < 0.1$

Powertrain Specification

- Signal Temporal Logic: temporal specification for signals





$\square_{[0,100]}\, x \in [1,3]$

$\square_{[60,100]} |x| < 0.1$

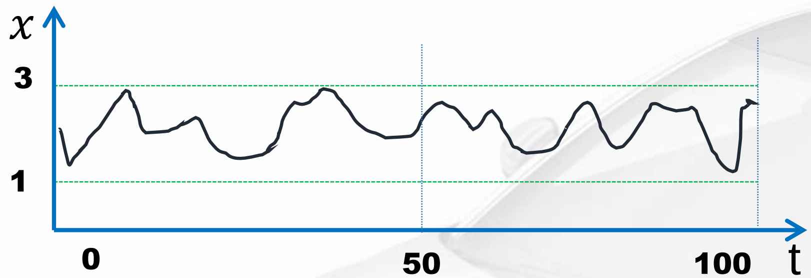
$\mathbf{U} \triangleq (\boldsymbol{x} < \mathbf{1} \vee \boldsymbol{x} > \mathbf{3}) \wedge (\boldsymbol{t} \leq \mathbf{100})$

$\mathbf{U} \triangleq (\boldsymbol{x} < -\mathbf{0.1} \vee \boldsymbol{x} > \mathbf{0.1}) \wedge (\boldsymbol{t} \geq \mathbf{60} \wedge \boldsymbol{t} \leq \mathbf{100})$

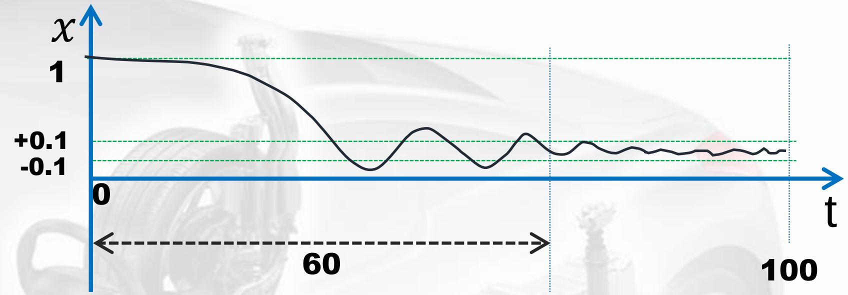
- Encoded as safety properties

Powertrain Specification

- Signal Temporal Logic: temporal specification for signals



$\Box_{[0,100]}\, x \in [1,3]$



$\Box_{[60,100]} |x| < 0.1$

$\mathbf{U} \triangleq (x < 1 \vee x > 3) \wedge (t \leq 100)$

$\mathbf{U} \triangleq (x < -0.1 \vee x > 0.1) \wedge (t \geq 60 \wedge t \leq 100)$

- Encoded as safety properties

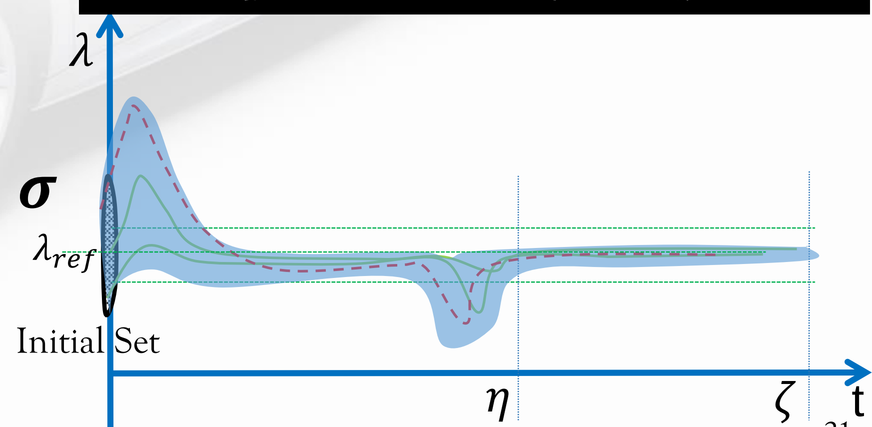
Verification goal:

Given initial set $\boldsymbol{\Phi}$ and switching signals $\boldsymbol{\sigma}$

Prove that

$$\textbf{rise} \Rightarrow \Box_{[\eta,\zeta]}[0.98\,\lambda_{ref}, 1.02\lambda_{ref}]$$

Technique: Reachability Computation

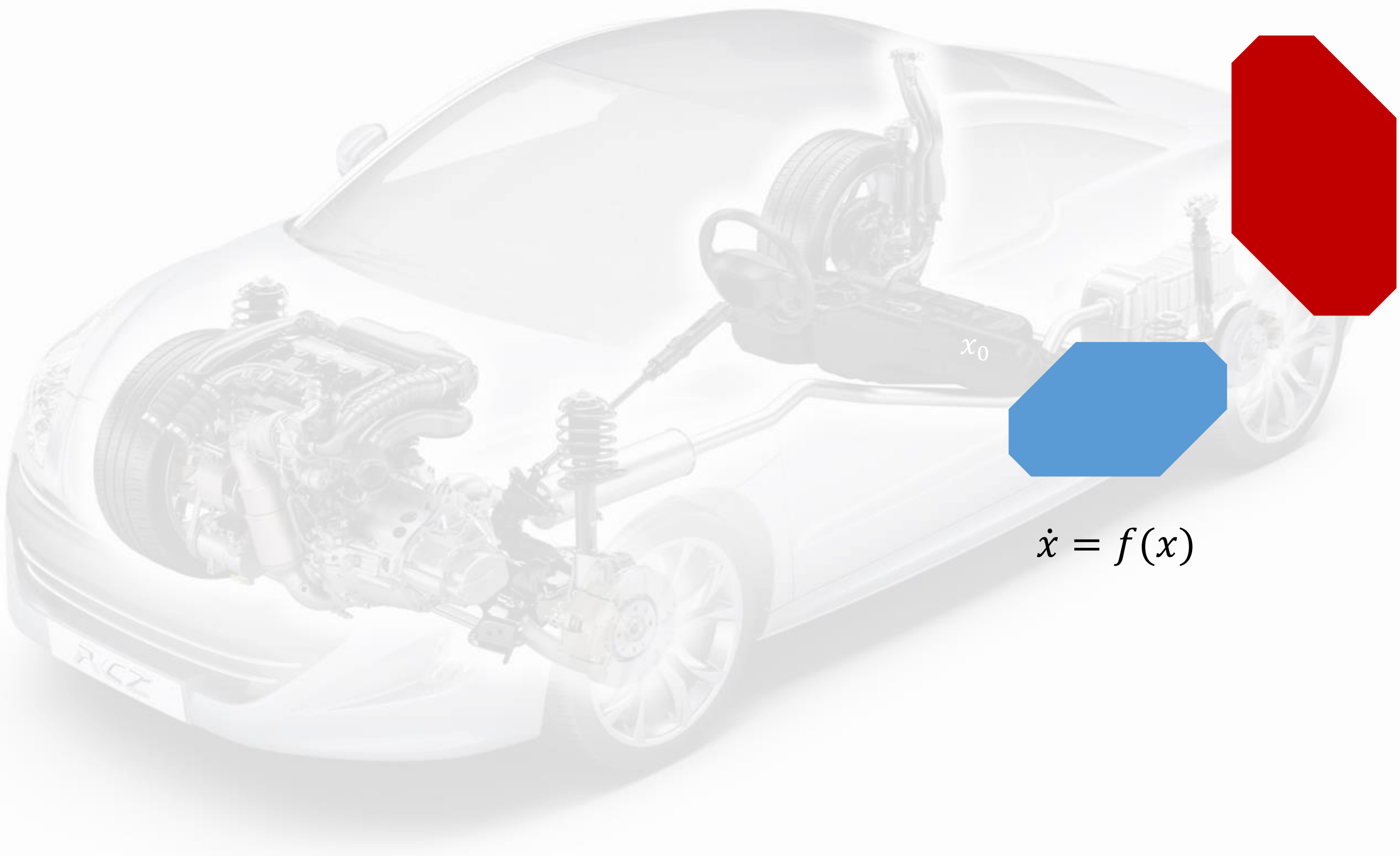


Outline

- ✓ Motivation & Challenges
- ✓ Powertrain Benchmark
- ✓ Specification
- Simulation Based Verification Technique
- Engineering
- Verification Results
- Conclusions and Future Work

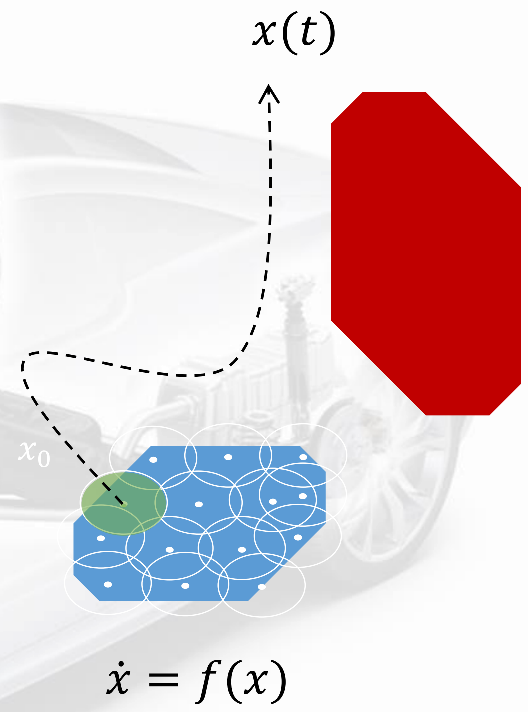
A Simple (Often The Only) Strategy

- Given start Θ and unsafe U



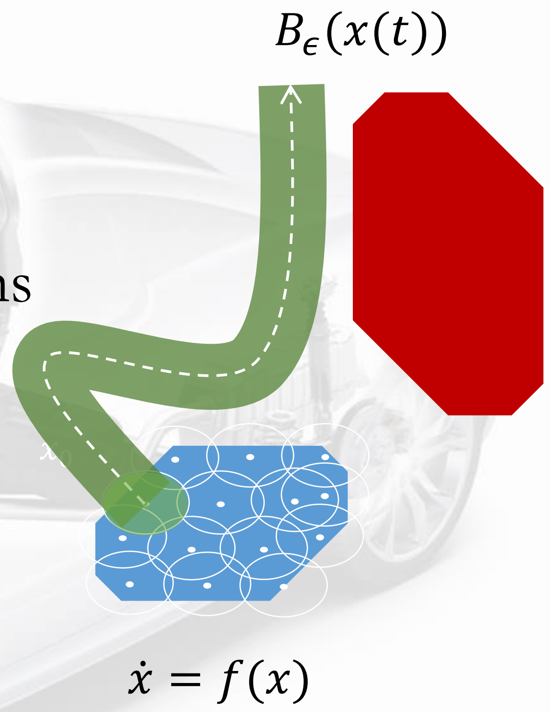
A Simple (Often The Only) Strategy

- Given start Θ and unsafe U
- Compute finite cover of initial set
- Simulate from the center x of each cover



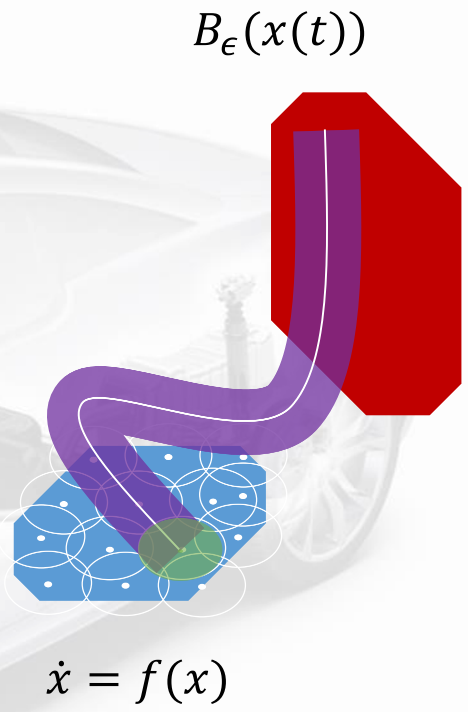
A Simple (Often The Only) Strategy

- Given start Θ and unsafe U
- Compute finite cover of initial set
- Simulate from the center x of each cover
- **Bloat** simulation so that bloated tube contains all trajectories from the cover
- Union = over-approximation of reach set



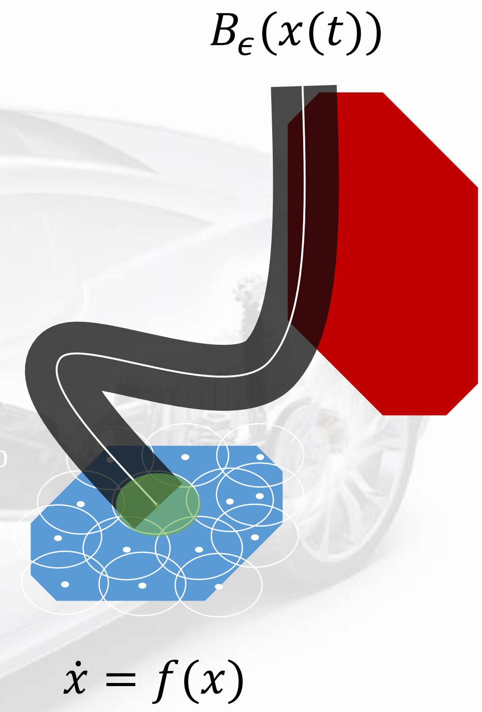
A Simple (Often The Only) Strategy

- Given start Θ and unsafe U
- Compute finite cover of initial set
- Simulate from the center x of each cover
- **Bloat** simulation so that bloated tube contains all trajectories from the cover
- Union = over-approximation of reach set
- Check intersection/containment with U
- Refine



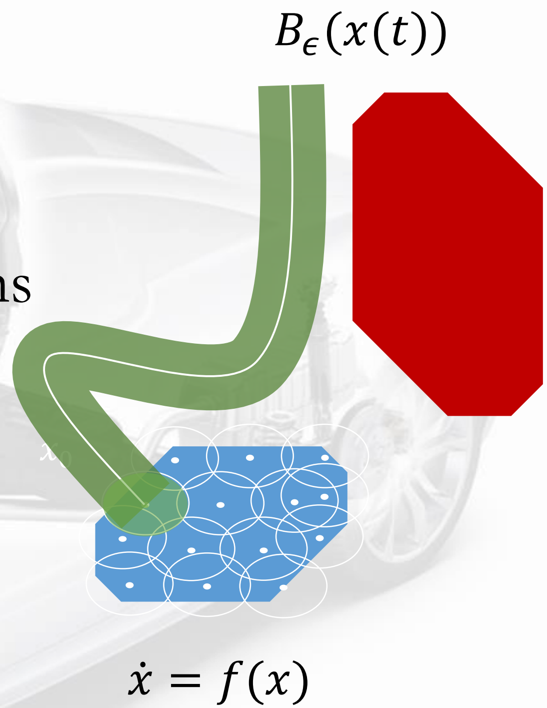
A Simple (Often The Only) Strategy

- Given start Θ and unsafe U
- Compute finite cover of initial set
- Simulate from the center x of each cover
- **Bloat** simulation so that bloated tube contains all trajectories from the cover
- Union = over-approximation of reach set
- Check intersection/containment with U
- Refine



A Simple (Often The Only) Strategy

- Given start Θ and unsafe U
- Compute finite cover of initial set
- Simulate from the center x of each cover
- **Bloat** simulation so that bloated tube contains all trajectories from the cover
- Union = over-approximation of reach set
- Check intersection/containment with U
- Refine



How much to bloat the sample simulation?

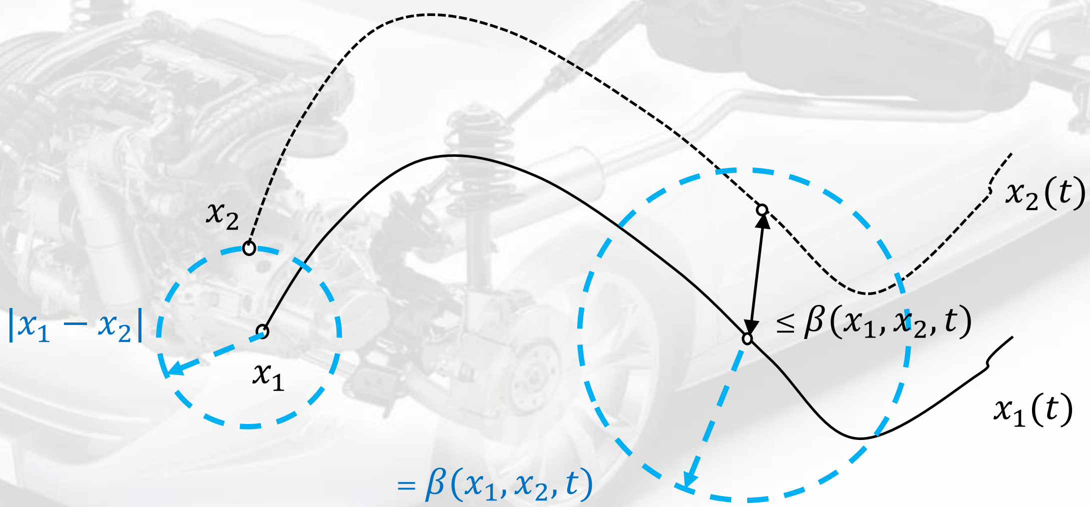
Discrepancy Function

<u>*Discrepancy Function*</u>: capturing the continuity of ODE solutions

executions that start close, stay close

β is called a discrepancy function of the system if for any two states x_1 and x_2,

$$|x_1(t) - x_2(t)| \leq \beta(x_1, x_2, t)$$



Discrepancy Function

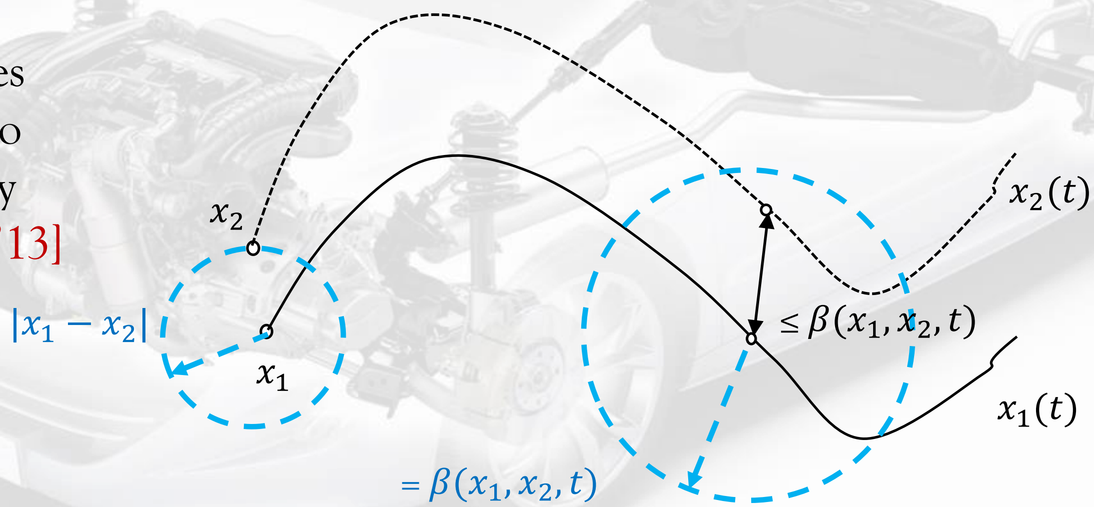
<u>Discrepancy Function</u>: capturing the continuity of ODE solutions

executions that start close, stay close

β is called a discrepancy function of the system if for any two states x_1 and x_2,

$$|x_1(t) - x_2(t)| \leq \beta(x_1, x_2, t)$$

Use proof techniques in Control Theory to compute discrepancy function [EMSOFT'13]



<u>Discrepancy functions are given as model annotations, i.e. β is given by the user</u>

Discrepancy Function

$$\dot{p} = c_1(2\theta(c_{20}p^2 + c_{21}p + c_{22}) - c_{12}(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2))$$
$$\dot{\lambda} = c_{26}(c_{15} + c_{16}c_{25}F_c + c_{17}c_{25}^2F_c^2 + c_{18}\dot{m}_c + c_{19}\dot{m}_c c_{25}F_c - \lambda)$$
$$\dot{p_e} = c_1(2c_{23}\theta(c_{20}p^2 + c_{21}p + c_{22}) - (c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2))$$
$$i = c_{14}(c_{24}\lambda - c_{11})$$

where

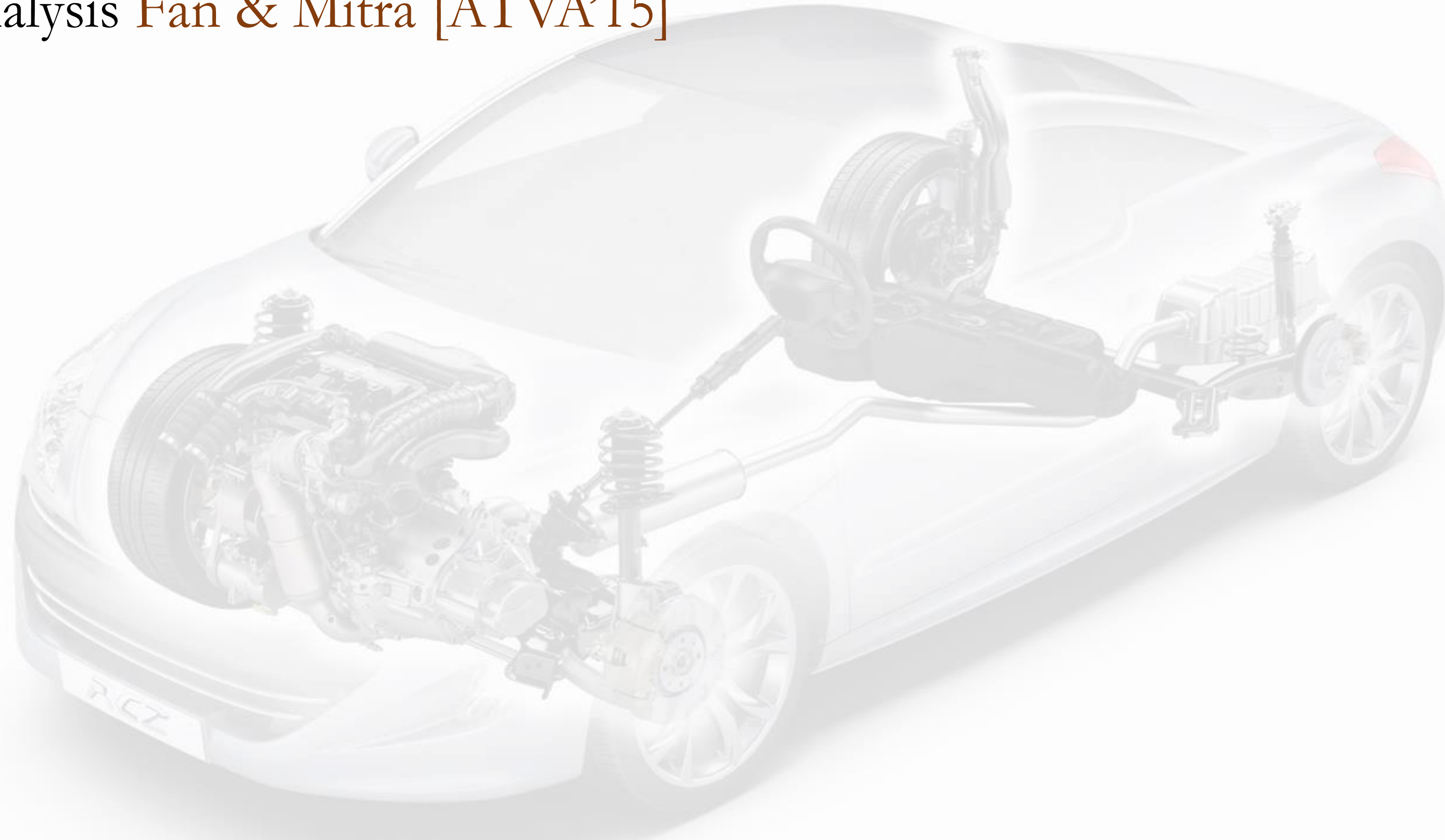
$$F_c = \frac{1}{c_{11}}(1 + i + c_{13}(c_{24}\lambda - c_{11}))(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2)$$
$$\dot{m}_c = c_{12}(c_2 + c_3\omega p + c_4\omega p^2 + c_5\omega p^2)$$

All known tools failed to find any discrepancy functions

Discrepancy functions are given as model annotations, i.e. β is given by the user

On-The-Fly Discrepancy

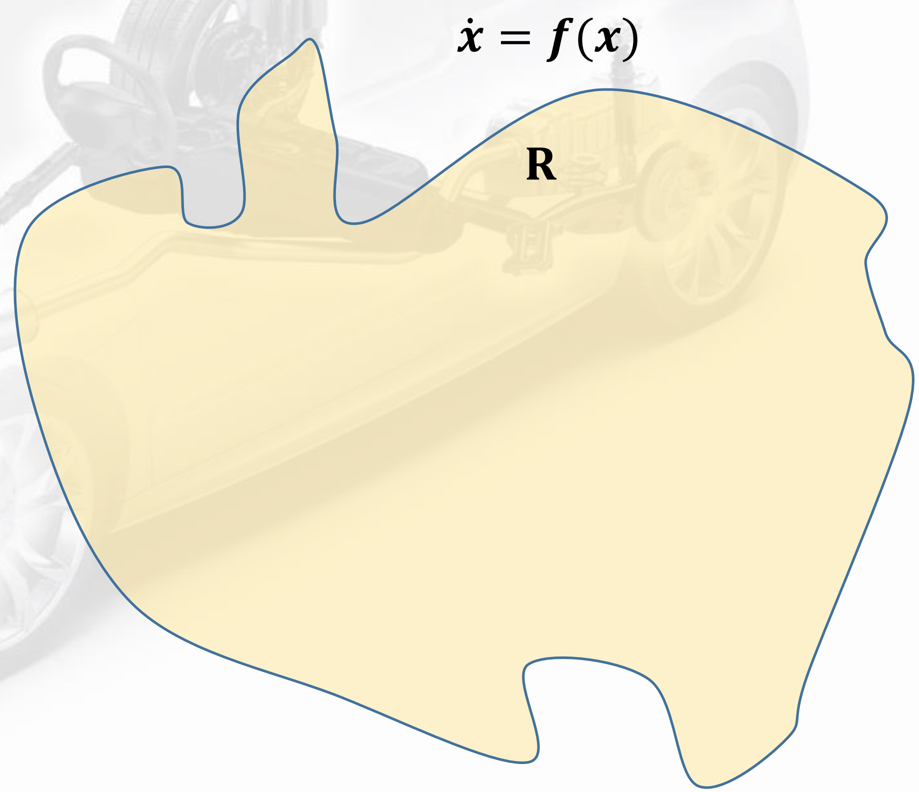
- Computing discrepancy function from simulations and static analysis Fan & Mitra [ATVA'15]



On-The-Fly Discrepancy

- Computing discrepancy function from simulations and static analysis Fan & Mitra [ATVA'15]

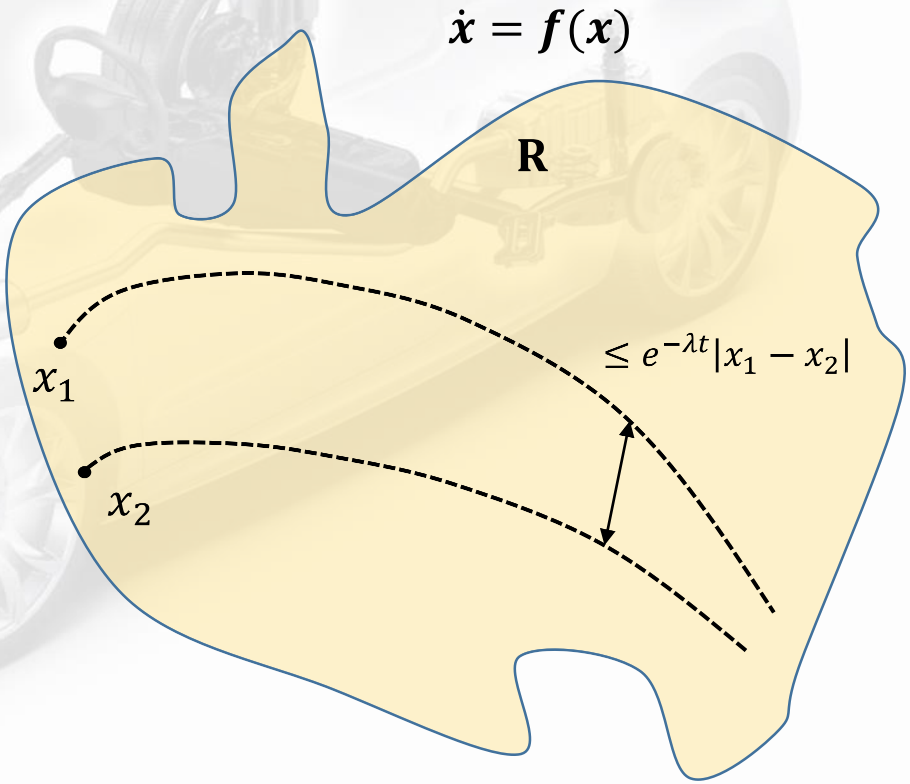
- Key principle
  - $J = \frac{\partial f}{\partial x}$
  - If $\mathrm{eig}(J + J^T) < 0$ in **R** then trajectories converge in **R**



On-The-Fly Discrepancy

- Computing discrepancy function from simulations and static analysis Fan & Mitra [ATVA'15]

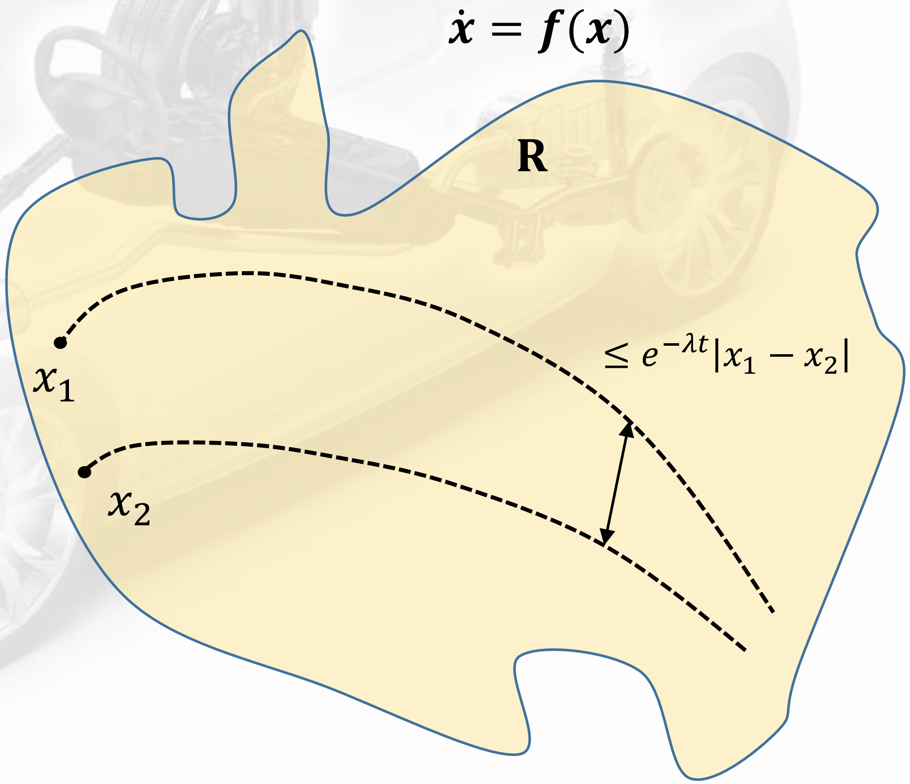
- Key principle
  - $J = \frac{\partial f}{\partial x}$
  - If $\text{eig}(J + J^T) < 0$ in **R** then trajectories converge in **R**



On-The-Fly Discrepancy

- Computing discrepancy function from simulations and static analysis Fan & Mitra [ATVA'15]

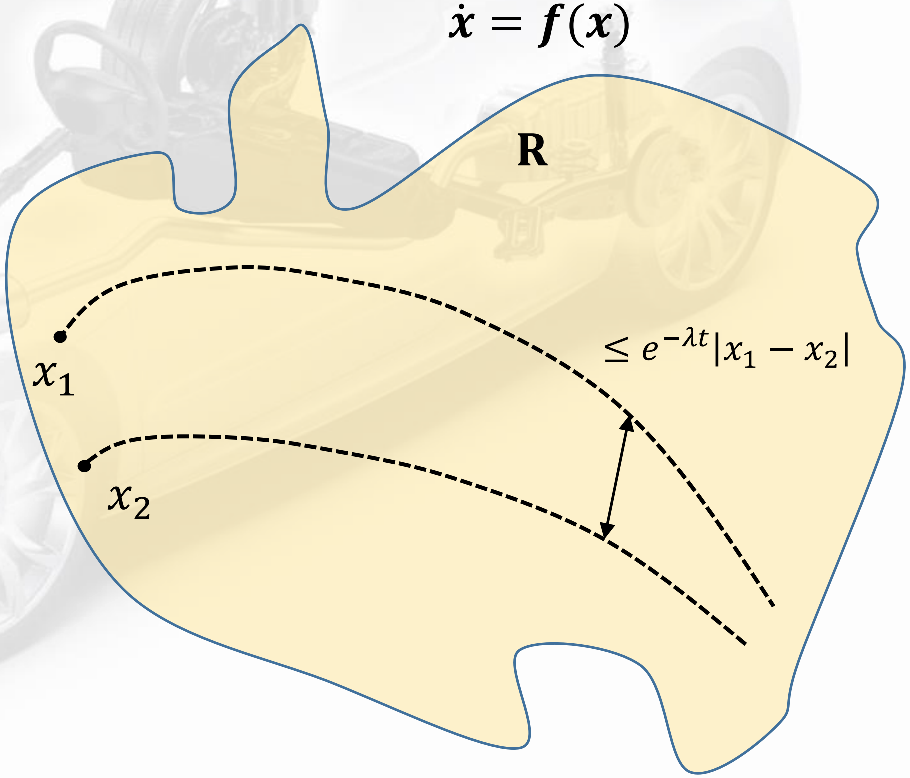
- Key principle
  - $J = \frac{\partial f}{\partial x}$
  - If $\mathrm{eig}(J + J^T) < 0$ in **R** then trajectories converge in **R**
  - Compute $\max \mathrm{eig}(J + J^T)$ in **R**
  - Gives a local discrepancy function in region **R**



On-The-Fly Discrepancy

- Computing discrepancy function from simulations and static analysis Fan & Mitra [ATVA'15]

- Key principle
  - $J = \frac{\partial f}{\partial x}$
  - If $\mathrm{eig}(J + J^T) < 0$ in **R** then trajectories converge in **R**
  - Compute $\max \mathrm{eig}(J + J^T)$ in **R**
  - Gives a local discrepancy function in region **R**



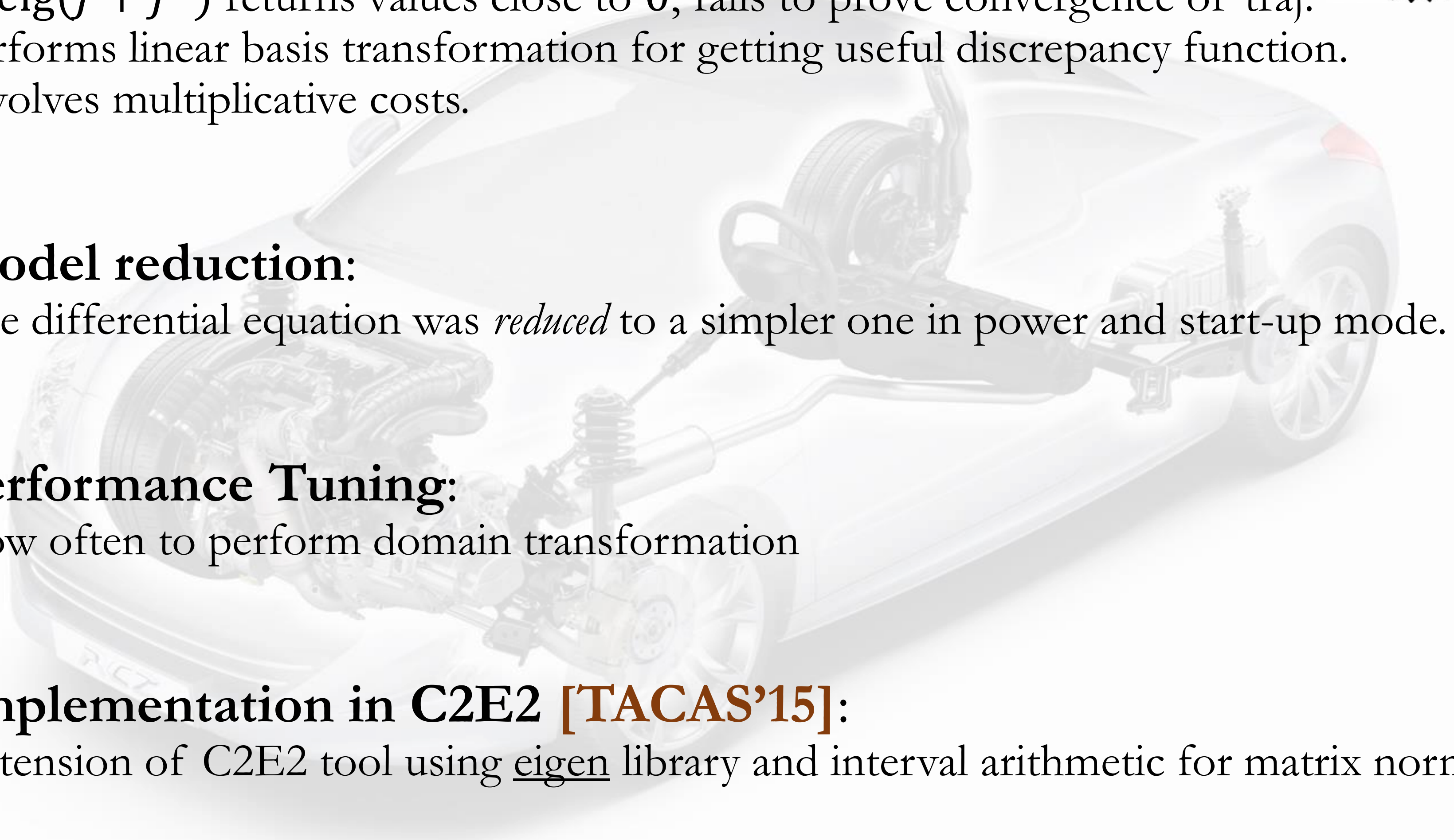
We apply on-the-fly discrepancy function for verifying powertrain control system

Engineering

- **Domain Transformation**:
  If $\mathrm{eig}(J + J^T)$ returns values close to 0, fails to prove convergence of traj.
  Performs linear basis transformation for getting useful discrepancy function.
  Involves multiplicative costs.

- **Model reduction**:
  The differential equation was *reduced* to a simpler one in power and start-up mode.

Engineering



- **Domain Transformation**:
  If $\text{eig}(J + J^T)$ returns values close to 0, fails to prove convergence of traj.
  Performs linear basis transformation for getting useful discrepancy function.
  Involves multiplicative costs.

- **Model reduction**:
  The differential equation was *reduced* to a simpler one in power and start-up mode.

- **Performance Tuning**:
  How often to perform domain transformation

- **Implementation in C2E2 [TACAS'15]**:
  Extension of C2E2 tool using eigen library and interval arithmetic for matrix norms.

Powertrain Verification Results

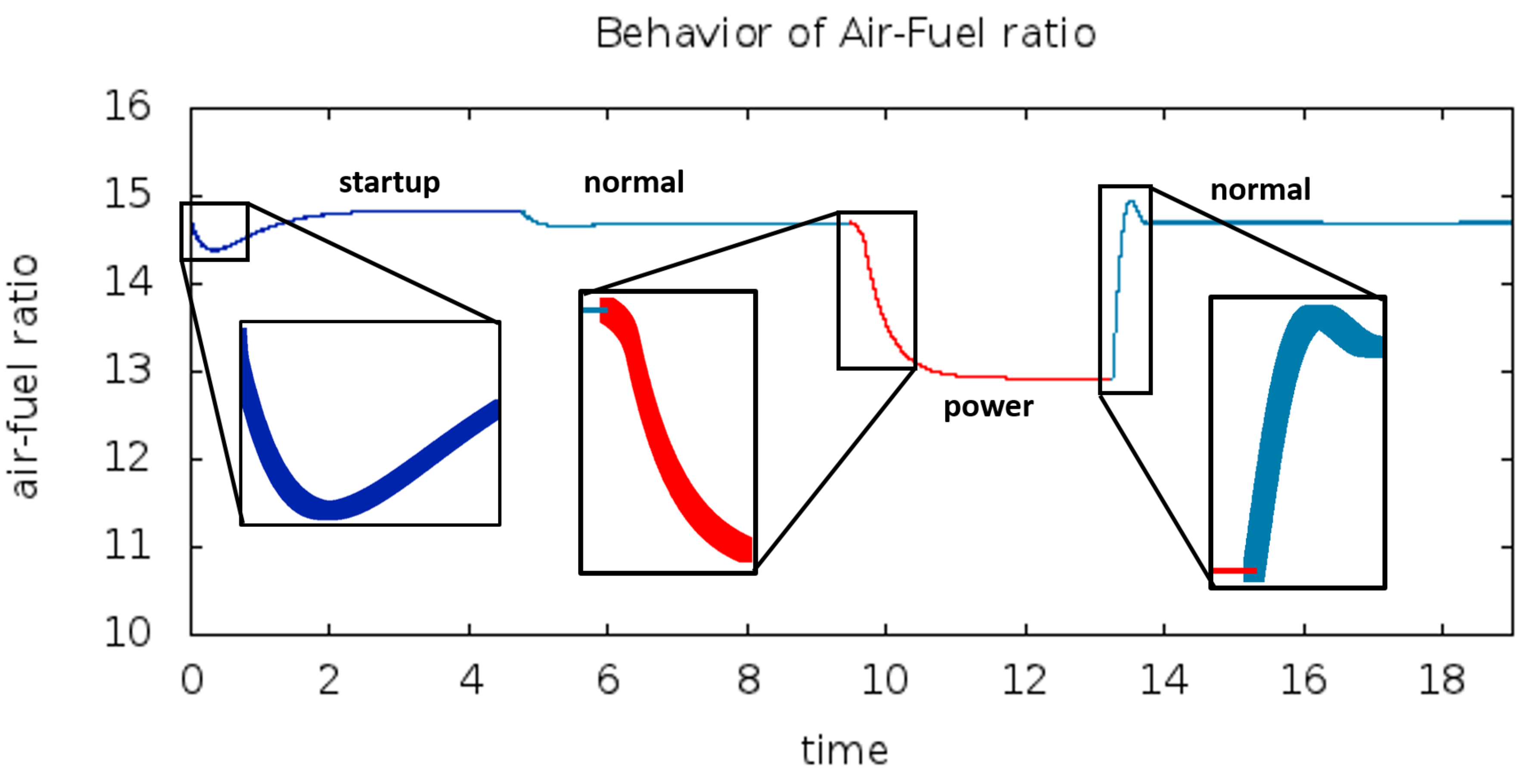
Verified many key specification for a given set of driver behaviors

Property	Mode	Sat	Sim.	Time
$\square\, \lambda \in [0.8\lambda_{ref}, 1.2\lambda_{ref}]$	all modes	Yes	53	11m58s
$\square\, \lambda \in [0.8\lambda_{ref}, 1.2\lambda_{ref}]$	startup	Yes	50	10m21s
$\square\, \lambda \in [0.8\lambda_{ref}, 1.2\lambda_{ref}]$	normal	Yes	50	10m21s
$\square\, \lambda \in [0.8\lambda_{ref}^{pwr}, 1.2\lambda_{ref}^{pwr}]$	power	Yes	53	11m12s
$\square\, \lambda \in [0.8\lambda'_{ref}, 1.2\lambda'_{ref}]$	power	No	4	0m43s
$rise \Rightarrow \square_{(\eta,\xi)}\lambda \in [0.98\ \lambda_{ref}, 1.02\lambda_{ref}]$	normal	Yes	50	10m15s
$(l = pwr) \Rightarrow \square_{(\eta,\xi)}\lambda \in [0.95\ \lambda_{ref}, 1.05\lambda_{ref}]$	power	Yes	53	11m35s
$(l = pwr) \Rightarrow \square_{(\eta/2,\xi)}\lambda \in [0.95\ \lambda_{ref}, 1.05\lambda_{ref}]$	power	No	4	0m45s

Safety properties (rows 1–5)

Performance properties (rows 6–8)

Reachable Set



Conclusions and Future Work

- Verified the polynomial hybrid system model in the Powertrain Control Benchmark

- Scalability of dynamic analysis tool C2E2 to handle systems of industrial complexity

Future Work:

- Handling properties with path integrals
- New algorithms for handling other models in the benchmark

Thank You

- Xiaoqing Jin
- Jyotirmoy Deshmukh
- Jim Kapinski
- Koichi Ueda
- Ken Butts

Questions?