Conformance Testing for the Design of Connected Vehicle Functions

Mohammad Reza Mousavi

UNIVERSITY OF LEICESTER

Based on joint work with:

Arend Aerts (TU Eindhoven, Netherlands),
Hugo Araujo (F.U. Pernambuco, Brazil),
Gustavo Carvalho (F.U. Pernambuco, Brazil),
Maciej Gazda (Leicester, UK),
Ties Hoenselaar (TU Eindhoven, Netherlands),
Narges Khakpour (Linneaus U., Sweden),
Morteza Mohaqeqi (Uppsala, Sweden),
Michel Reniers (TU Eindhoven, Netherlands),
Augusto Sampaio (F.U. Pernambuco, Brazil),
Masoumeh Taromi Rad (Halmstad, Sweden)

Results

- **Test case generation** algorithms for sound **conformance testing** of **cyber-physical systems**

- **Matlab-based tool prototype** to implement the algorithms:
  - soundness bound calculation,
  - test case execution, and
  - conformance analysis.

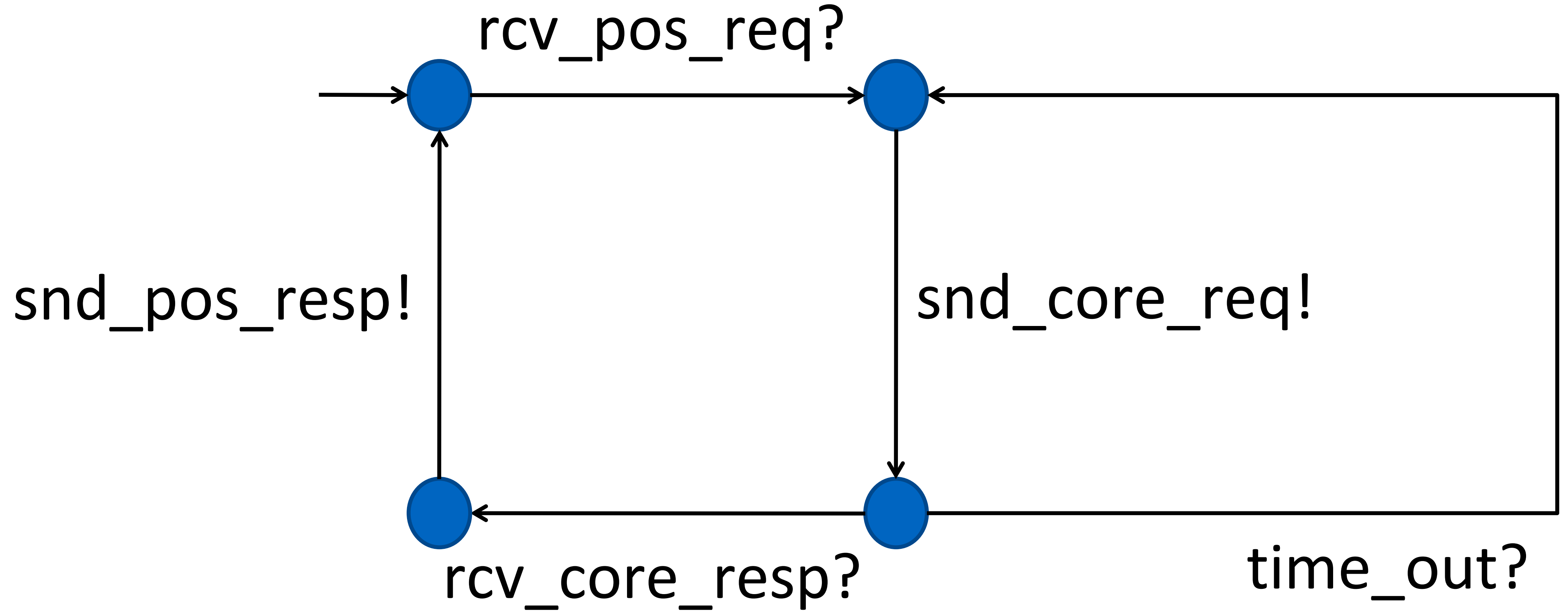
- Applied to a number **of case studies** from the automotive domain, including **connected platoons**

Model-Based Testing

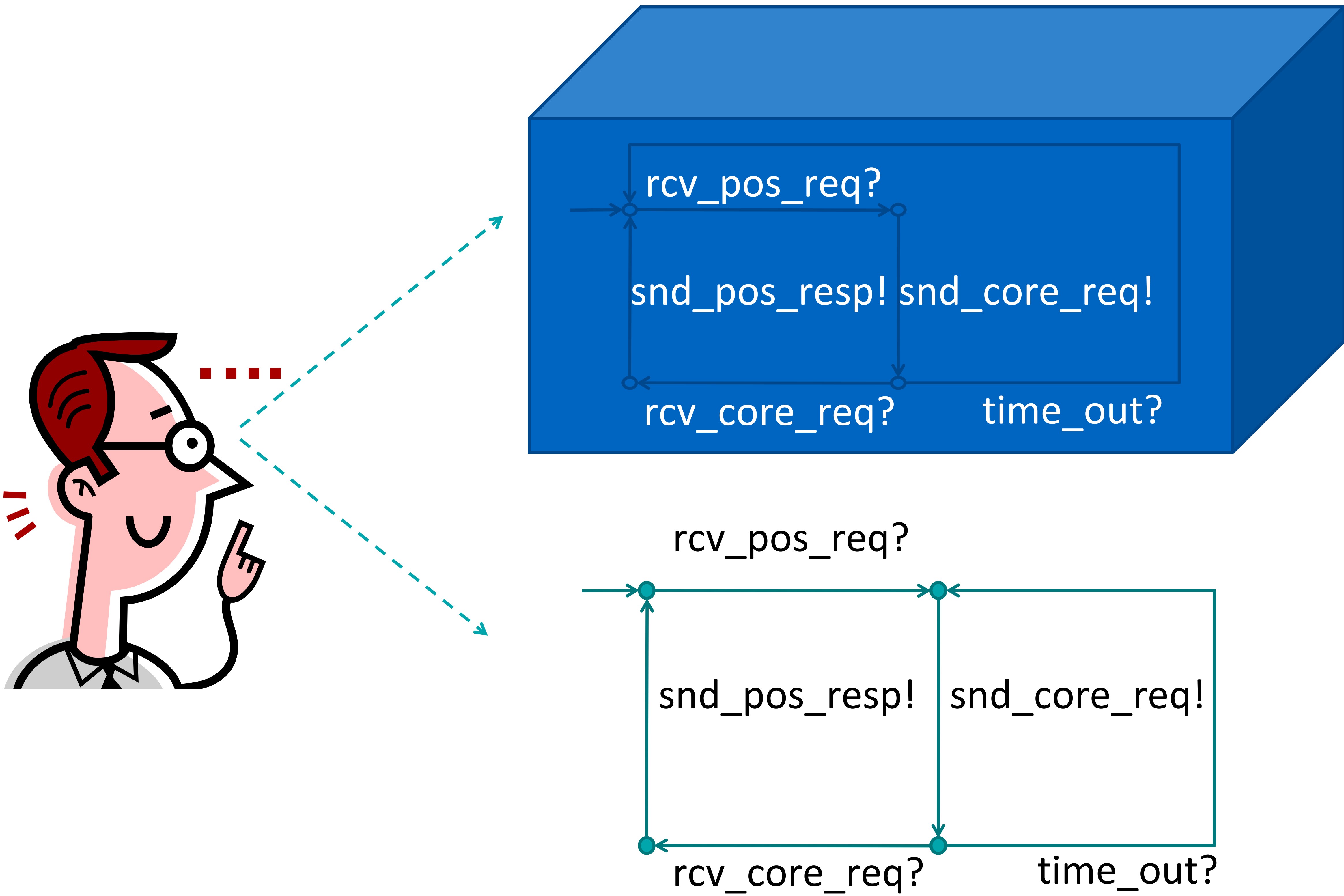
- **Abstractions** from reality
- Separating different **concerns**
- Approximating system behavior and / or its **environment**
  - Restricting environment interactions
  - Simpler than actual system
  - Easier to verify

Model-Based Testing

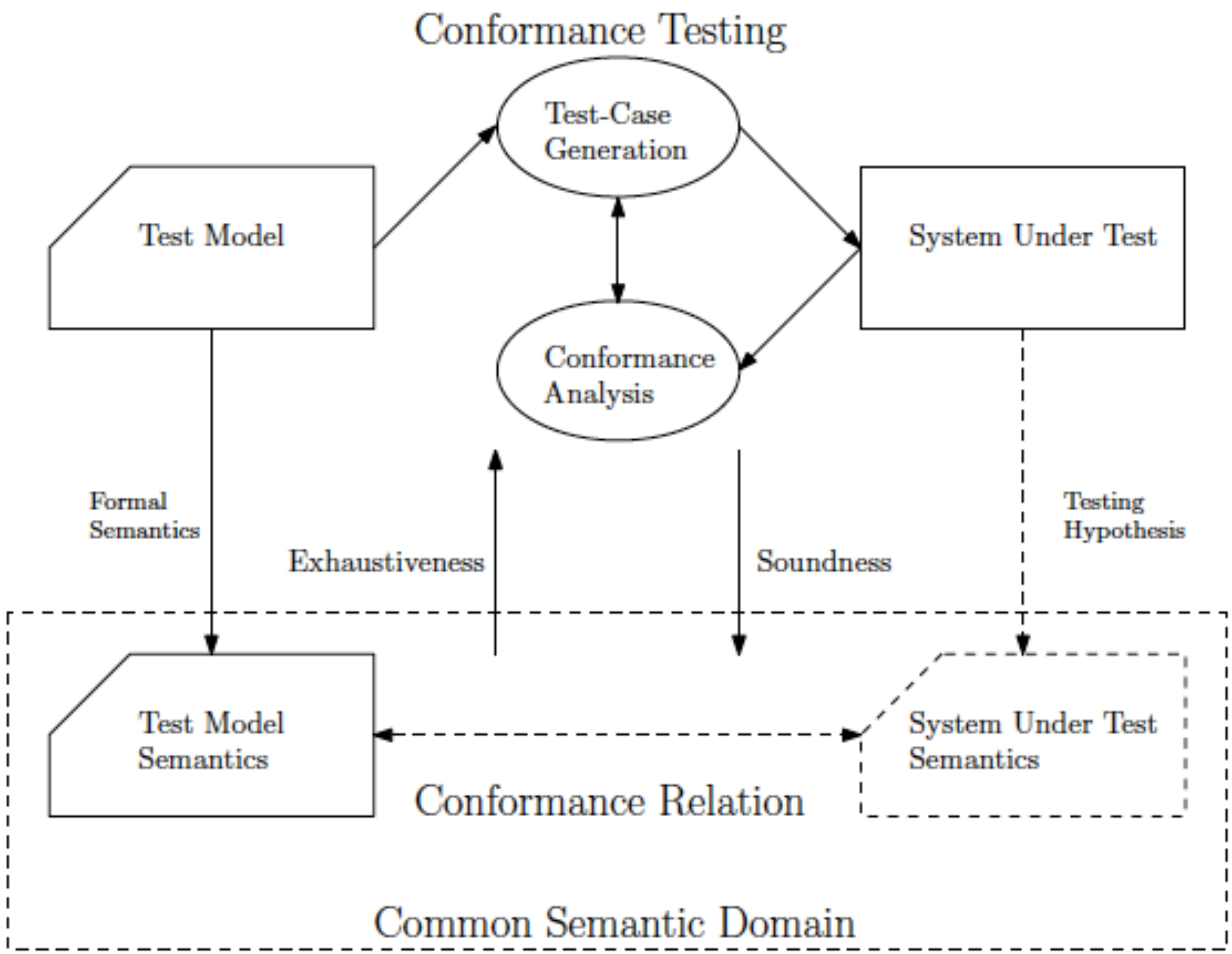
- Modeling the desired behavior (system) / possible interactions (environment)



Model-Based Testing

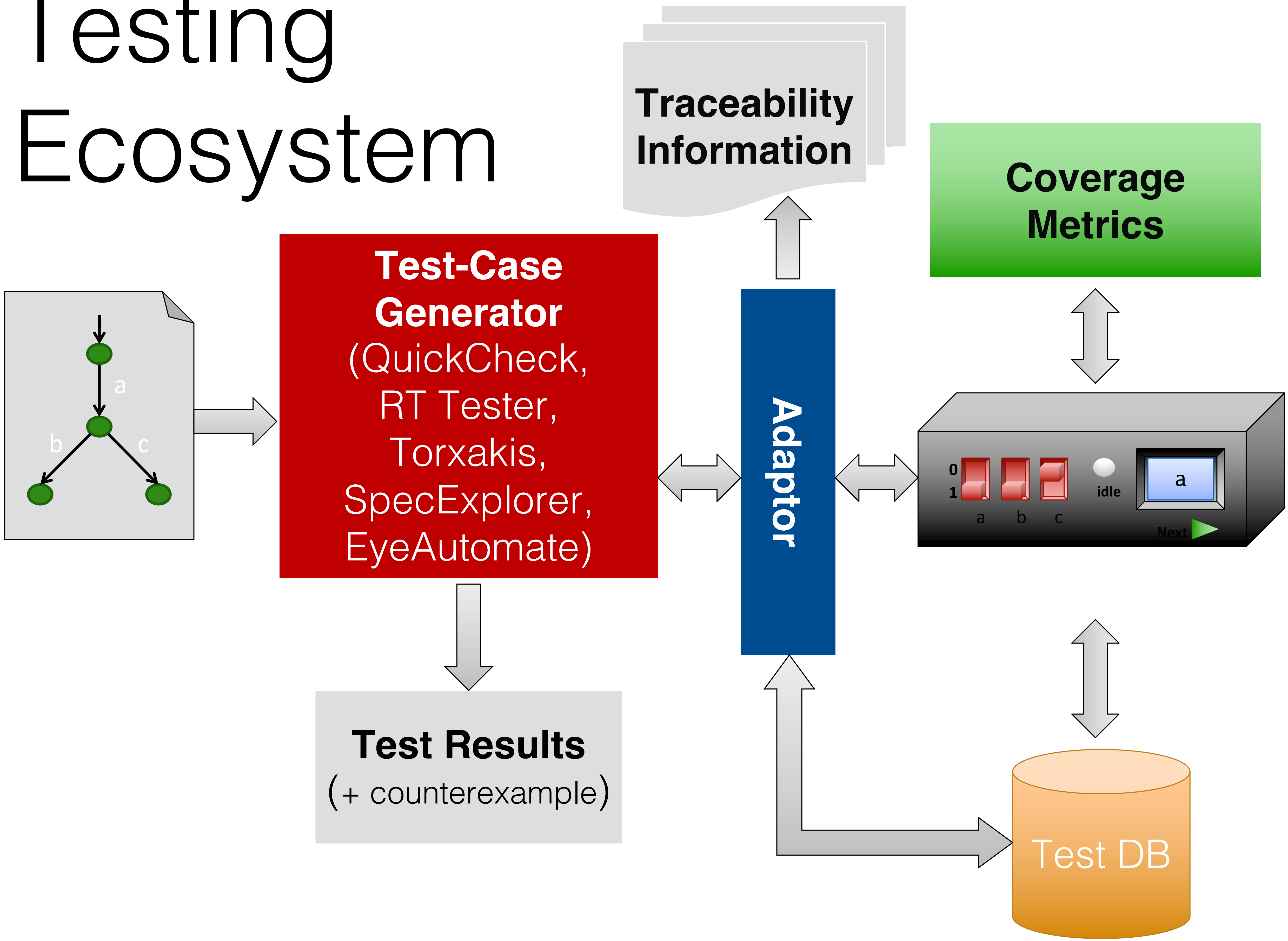


Conformance Testing



[Aerts, MRM, and Reniers. Model-Based Testing Cyber-Physical Systems, Handbook of CPS 2017]

Testing Ecosystem



Some Success Stories

- Asaadi, Khosravi, MRM, and Noroozi. **Towards Model-Based Testing of Electronic Funds Transfer Systems**. Proc. of FSEN 2011. Models publicly available on Assembla.

- Vishal, Kovacioglu, Kherazi, and MRM. **Integrating Model-Based and Constraint-Based Testing Using SpecExplorer**. Proc. of MoTiP 2012. (X-Ray Machines at Philips Healthcare)

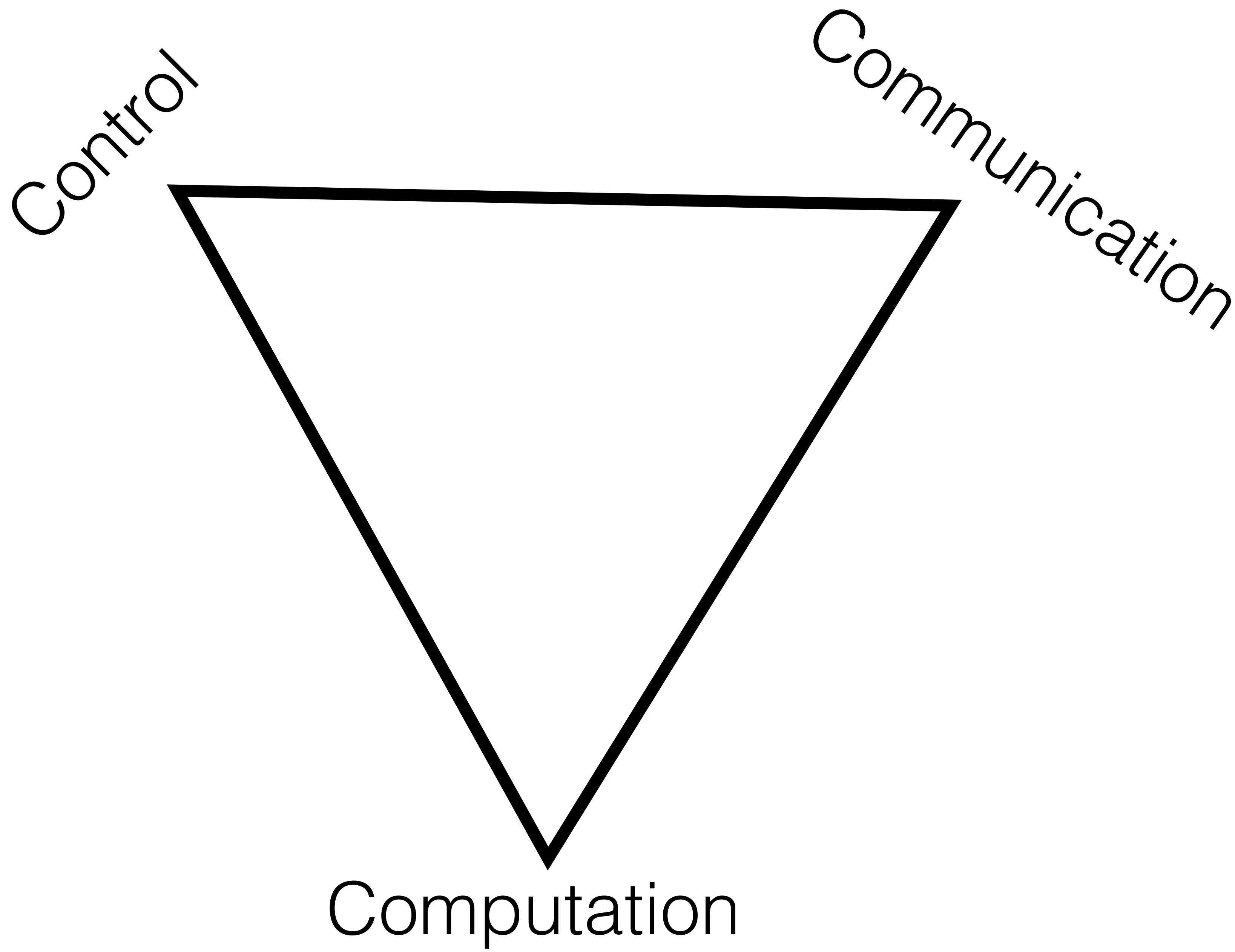
Conformance Testing

- Test case **generation**: sampling specification behaviour
- Test case **execution**: running tests on system under test
- Conformance **analysis**: reaching a verdict by comparing the test cases with the observed behaviour

Cyber-Physical Systems

Copyright SAP.com

Cyber-Physical Systems



Automotive CPS

"if you bought a premium-class automobile recently,
it probably contains close to
100 million lines of software code.

All that software executes on **70 to 100** microprocessor-based electronic control units (**ECUs**) **networked** throughout the body of your car."

-- Manfred Broy,
IEEE Spectrum, 2009

Copyright SAP.com

Automotive CPS

"By 2025, the share of **software** in the car industry will increase to **25%** of the total value;
the share of **software and hardware** will increase to **65%** of the total value."

--Roemer and Kramer
The Intelligent Car, 2010

BMW's 100th Birthday

"Our task is to preserve our business model without surrendering it to an internet player.

Otherwise we will end up … delivering only the **metal bodies** for them."

http://bit.ly/bmw_100

Automotive CPS

- 90% of the **innovation** in Sw.
- **1GB** downloadable Sw.
- live updates every **2 days**
- Service scope include vehicle, app and **cloud**

Continuous deployment of mission critical software…

Automotive CPS

Company	Autonomous miles	Disengagements	Rate per 1000 miles
Google	635868	124	0.20
Cruise	10015	284	28.36
Nissan	4099	28	6.83
Delphi	3125	178	56.95
Bosch	983	1442	1466.94
Mercedes	673	336	498.95
BMW	638	1	1.57
Ford	590	3	5.08
Tesla	550	182	330.91

Disengagement Rates for
Major Autonomous Vehicles

(source: IEEE Spectrum, February 2017)

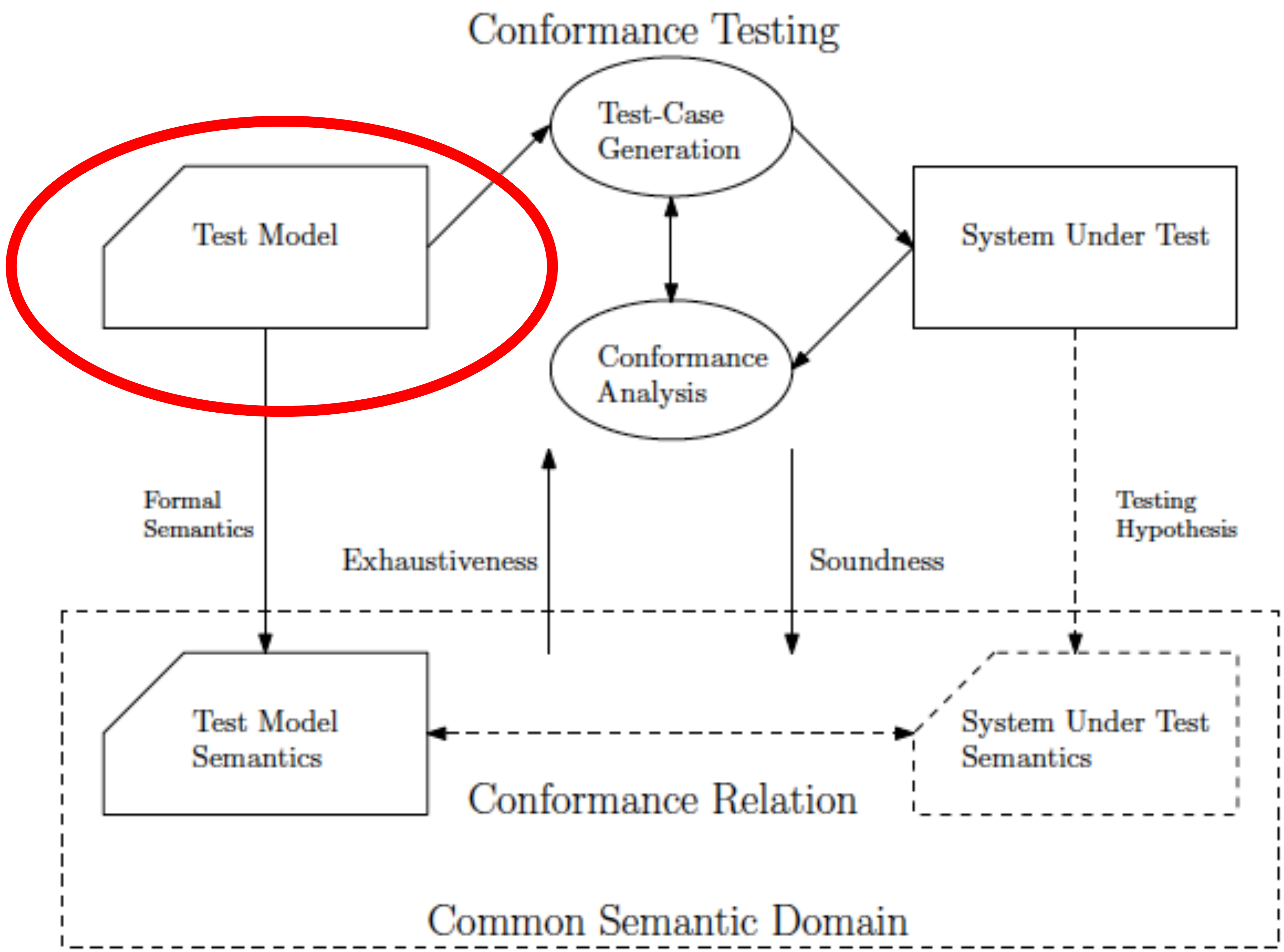
HOUSES OF PARLIAMENT
PARLIAMENTARY OFFICE OF SCIENCE & TECHNOLOGY

"Vehicles capable of driving without human intervention are rapidly moving up the policy agenda.

The main policy challenges are **verifying the safety and reliability** of autonomous road vehicles …"

www.parliament.uk/briefing-papers/post-pn-443.pdf

Model-Based Testing



CPS Dynamics and Control

To analyze a cyber-physical system,
such as a pacemaker,
we need to consider
the **discrete software controller**
interacting with the **physical world**,
which is typically modeled by
differential equations.

-- Rajeev Alur,
CACM 10/2013

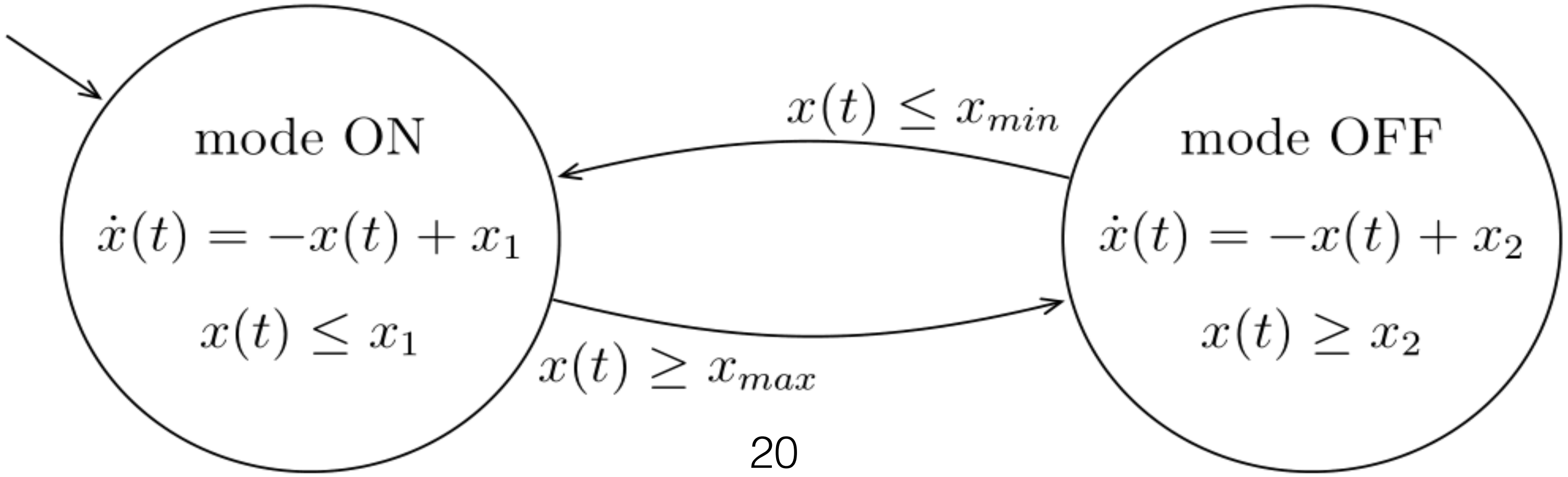
Models for CPS

Control theory:

- piecewise linear/affine systems,
- jump-flow systems

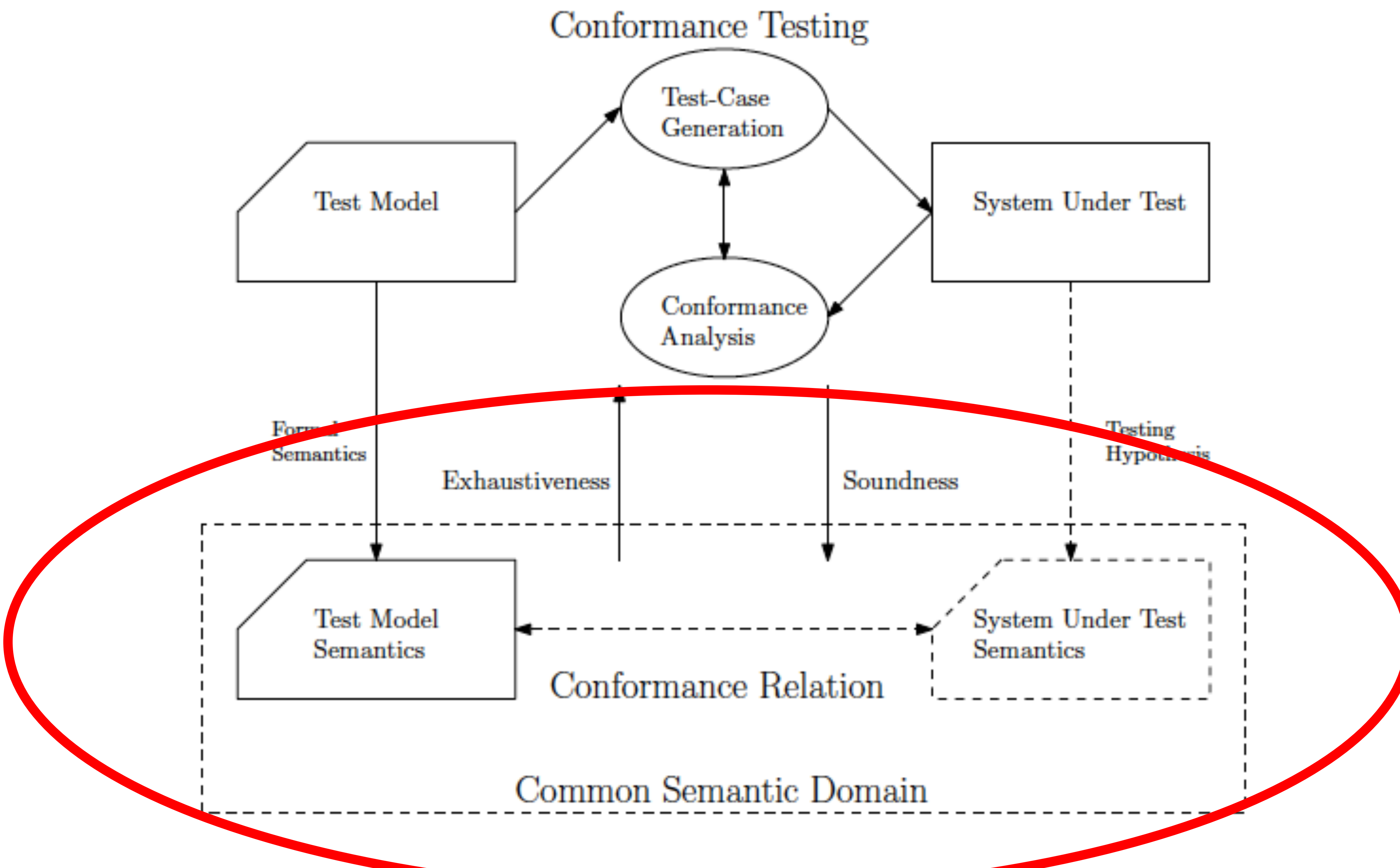
Computer science:

- finite state machines,
- labeled transition systems



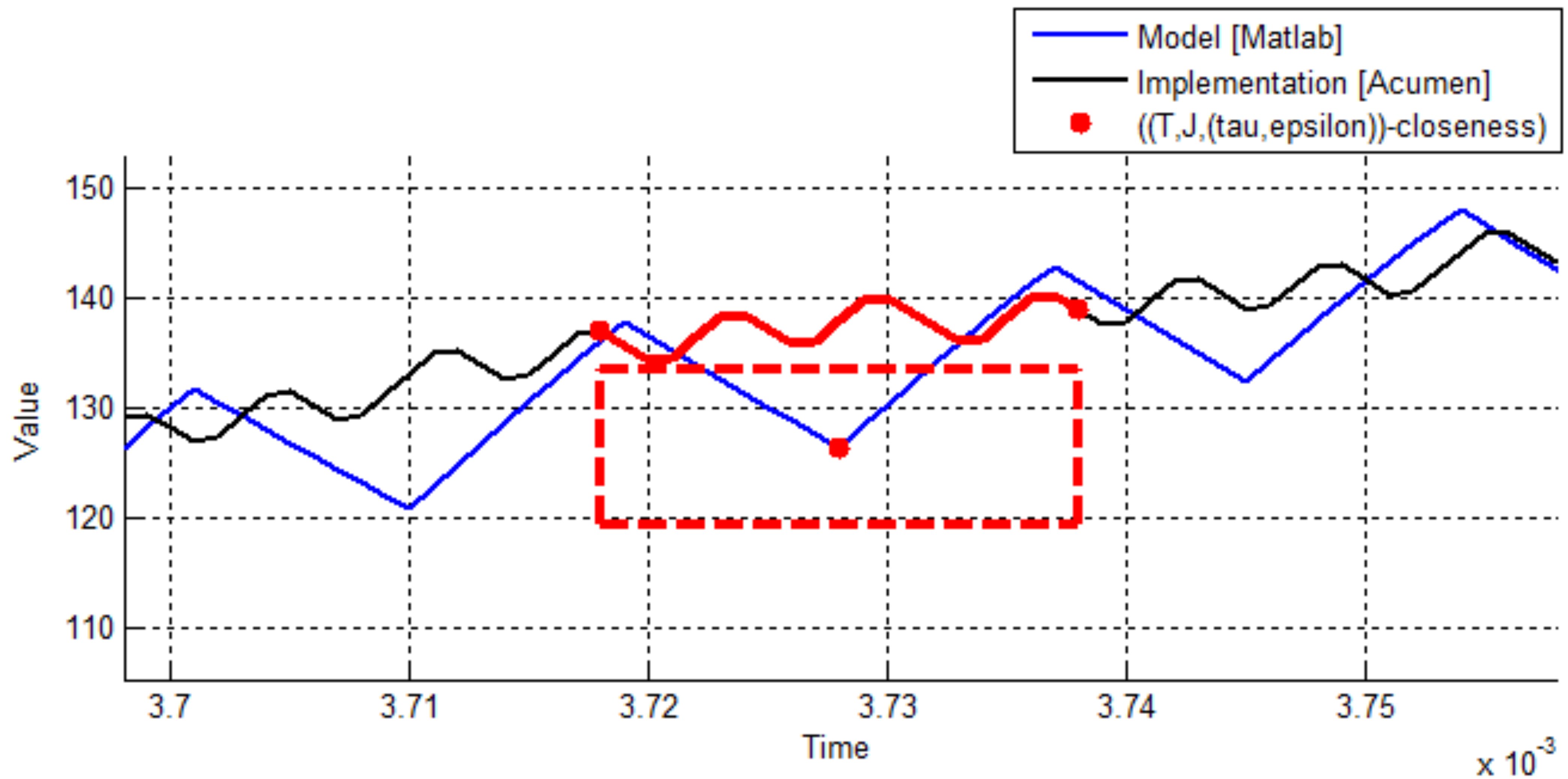
[Khakpour and MRM. CONCUR 2015]

Model-Based Testing



Conformance for CPS

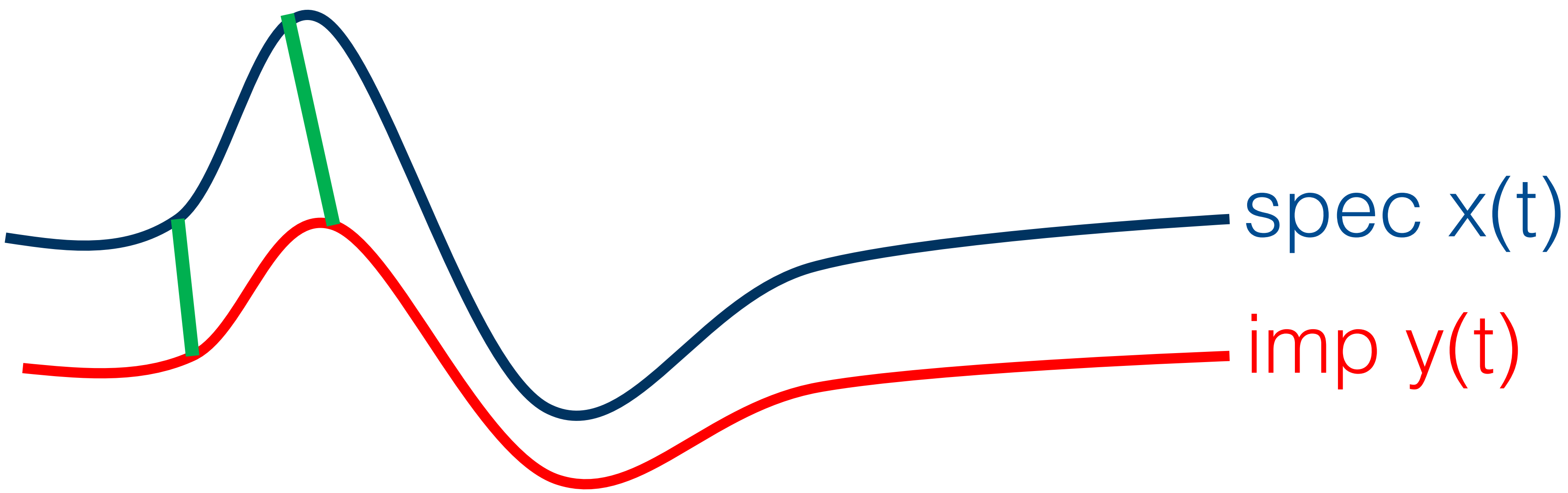
(τ, ε) -Conformance



[Abbas, Mittelmann and Fainekos. MEMOCODE 2014]

[Khakpour and MRM. CONCUR 2015]

Skorokhod-Conformance



$$\max\left(\sup_{t\in[0,T]}|\mathsf{r}(t)-t|,\ \sup_{t\in[0,T]}\mathcal{D}_{\mathcal{O}}\left(x\left(\mathsf{r}(t)\right),y(t)\right)\right)$$

[Deshmukh, Majumdar and Prabhu, FMSD 2017]

Logical Characterisation of Conformance

A logic $\mathcal{L}$ characterises a conformance relation $\preccurlyeq$, when

$$\mathrm{p} \preccurlyeq \mathrm{q} \iff \forall \varphi \in \mathcal{L} \,.\, (\, p \vDash \varphi \implies q \vDash \varphi)$$

[Fainekos and Pappas, TCS, 2009]
[Deshmukh, Majumdar and Prabhu, FMSD 2017]

Logical Characterisation of Conformance

A logic $\mathcal{L}$ characterises
an approximate conformance relation $\preccurlyeq_{\tau,\varepsilon}$, when

$$\mathrm{p} \preccurlyeq_{\tau,\varepsilon} \mathrm{q} \quad \Leftrightarrow$$
$$\forall \varphi \in \mathcal{L}\,.\, \left(p \vDash \varphi \implies q \vDash rel(\varphi)_{\tau,\varepsilon}\right)$$

[Fainekos and Pappas, TCS, 2009]
[Deshmukh, Majumdar and Prabhu, FMSD 2017]

Metric Temporal Logical

$$\varphi ::= true \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi U_I \varphi \mid \varphi R_I \varphi$$

$$F_I = true\ U_I \varphi$$

[Alur, Feder, and Henzinger, JACM, 96]

Relaxing Metric Temporal Logical

$$rel(true)_{\tau,\varepsilon} = true$$

$$rel(p)_{\tau,\varepsilon} = F_{[-\tau,+\tau]}(p_{\varepsilon}^{+})$$

$$rel(\neg p)_{\tau,\varepsilon} = F_{[-\tau,+\tau]}(p_{\varepsilon}^{-})$$

$$rel(\varphi \wedge \psi)_{\tau,\varepsilon} = rel(\varphi)_{\tau,\varepsilon} \wedge rel(\psi)_{\tau,\varepsilon}$$

$$rel(\varphi\, U_I\, \psi)_{\tau,\varepsilon} = rel(\varphi)_{\tau,\varepsilon}\, U_I\, rel(\psi)_{\tau,\varepsilon}$$

$$rel(\varphi\, R_I\, \psi)_{\tau,\varepsilon} = rel(\varphi)_{\tau,\varepsilon}\, R_I\, rel(\psi)_{\tau,\varepsilon}$$

[Gazda and MRM, Submitted] (Draft available upon request)

Logical Characterisation of (τ, ε)-Conformance

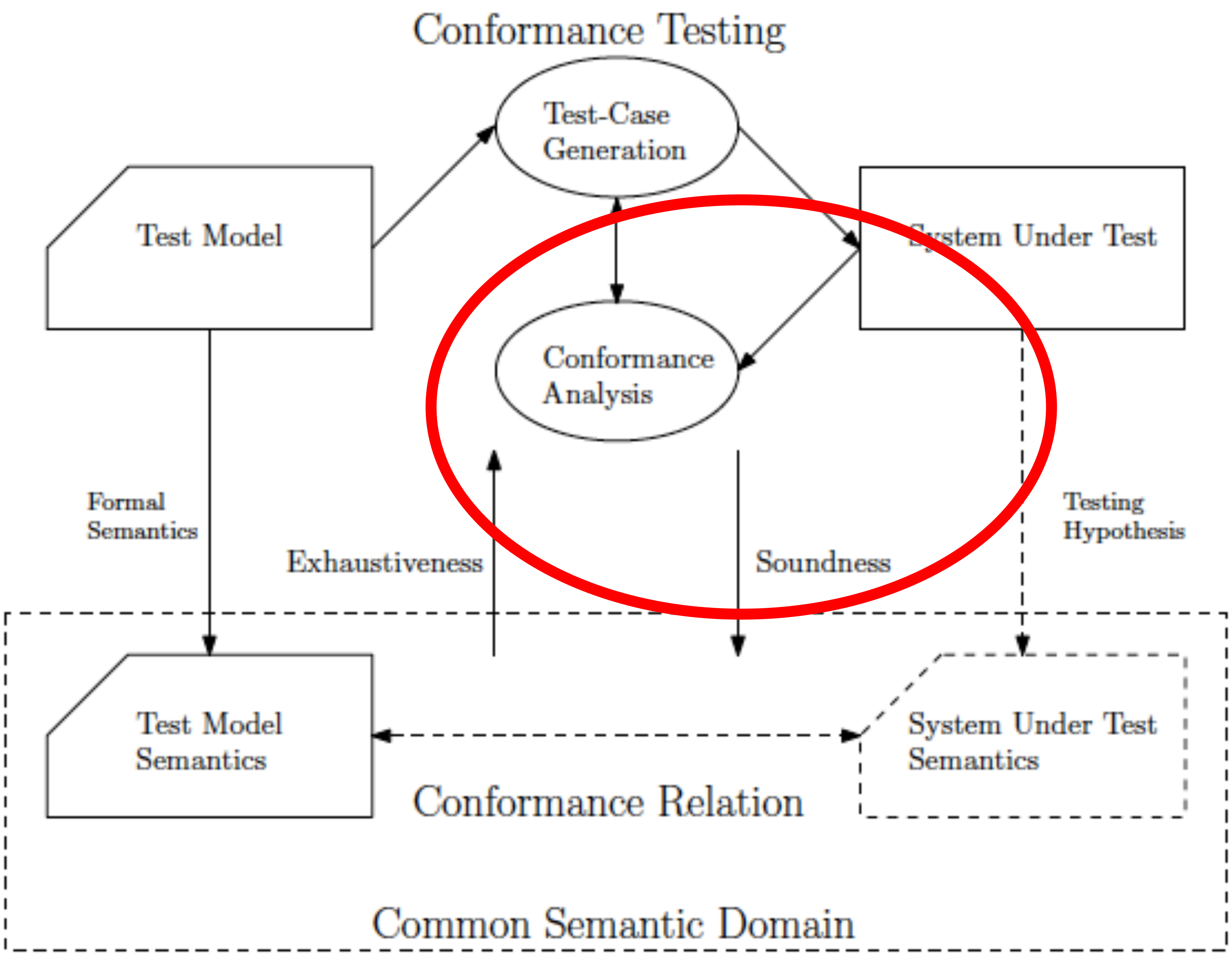
$$p \sim_{\tau,\varepsilon} q \quad \Leftrightarrow$$

$$\forall \varphi \in \mathcal{L} \,.\, \left(p \vDash \varphi \Longrightarrow q \vDash rel(\varphi)_{\tau,\varepsilon}\right) \wedge$$

$$\forall \varphi \in \mathcal{L} \,.\, \left(q \vDash \varphi \Longrightarrow p \vDash rel(\varphi)_{\tau,\varepsilon}\right)$$

[Gazda and MRM, Submitted] (Draft available upon request)

Model-Based Testing

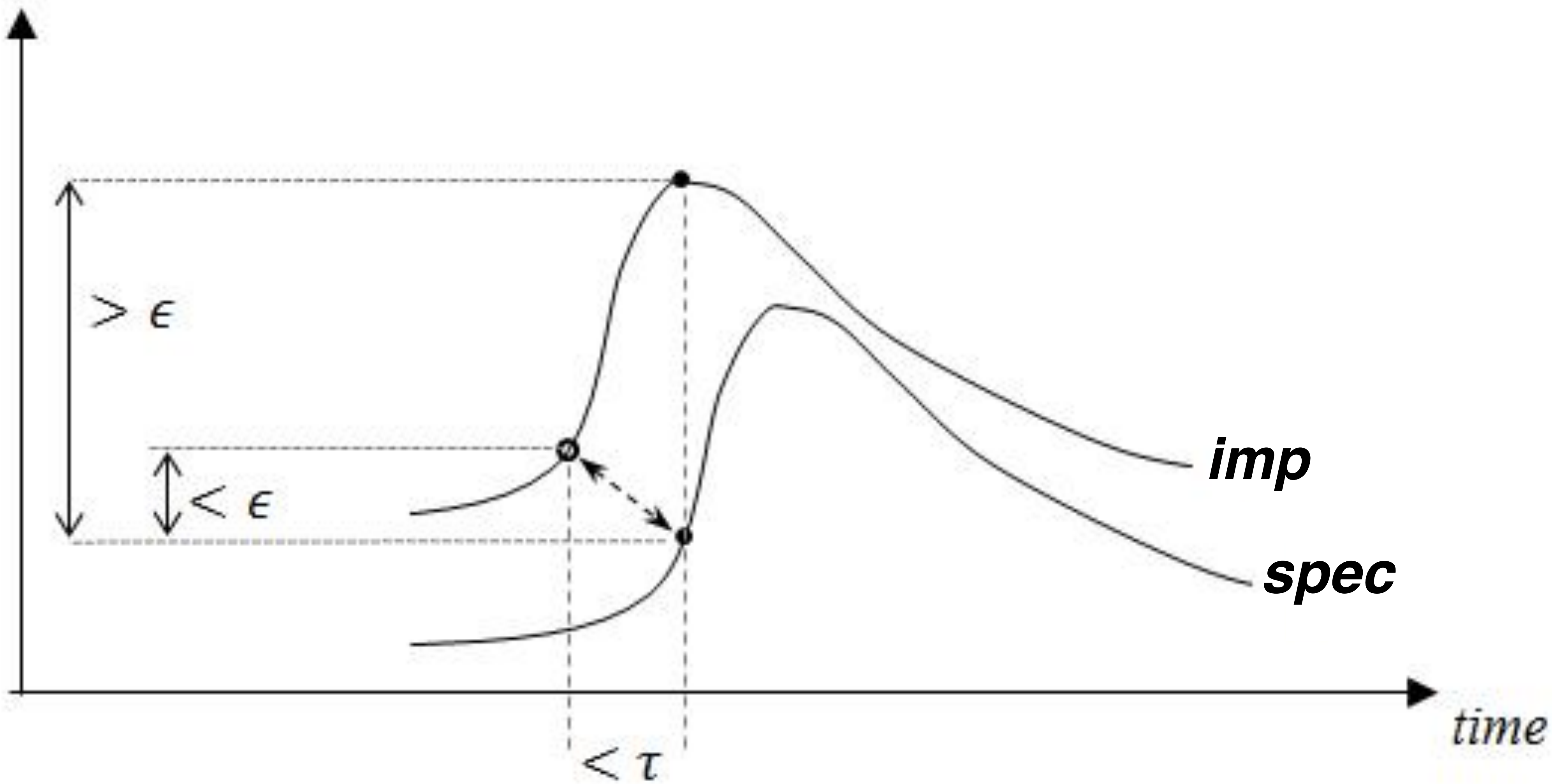


Conformance Analysis: Sampling

Connecting the Two Worlds

- **Soundness**: **only reject** non-conforming systems
- **Completeness**: **reject all** non-conforming systems

(Un)Soundness



The Theory

- Proven that testing with exact (τ, ε) conformance bounds leads to **unsound verdicts**

- Reinstating **soundness** requires **adjusting bounds** for conformance analysis and/or **adjusting the sampling rate**

- A **process** is required to apply these adjustments efficiently and effectively

Summing Up the Theory

Bottom line:
sampling rate and/or
error margin
should be adjusted to guarantee **soundness**.

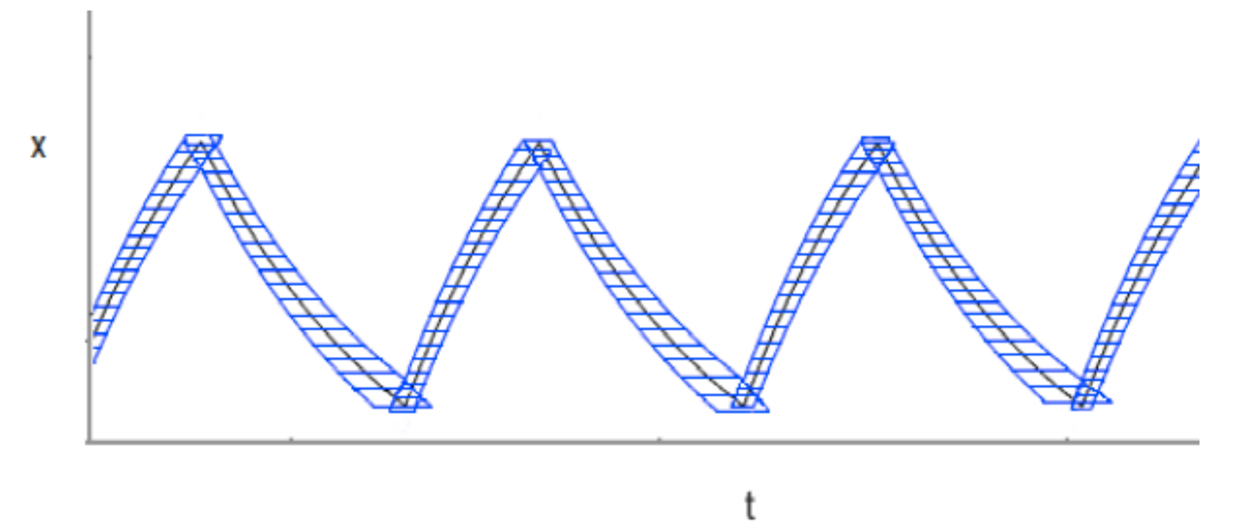
[Mohaqeqi and MRM. TASE 2016]

From Theory to Implementation

- Use **reachability analysis** to approximate the local changes in the dynamics

- Calculate **error margins**

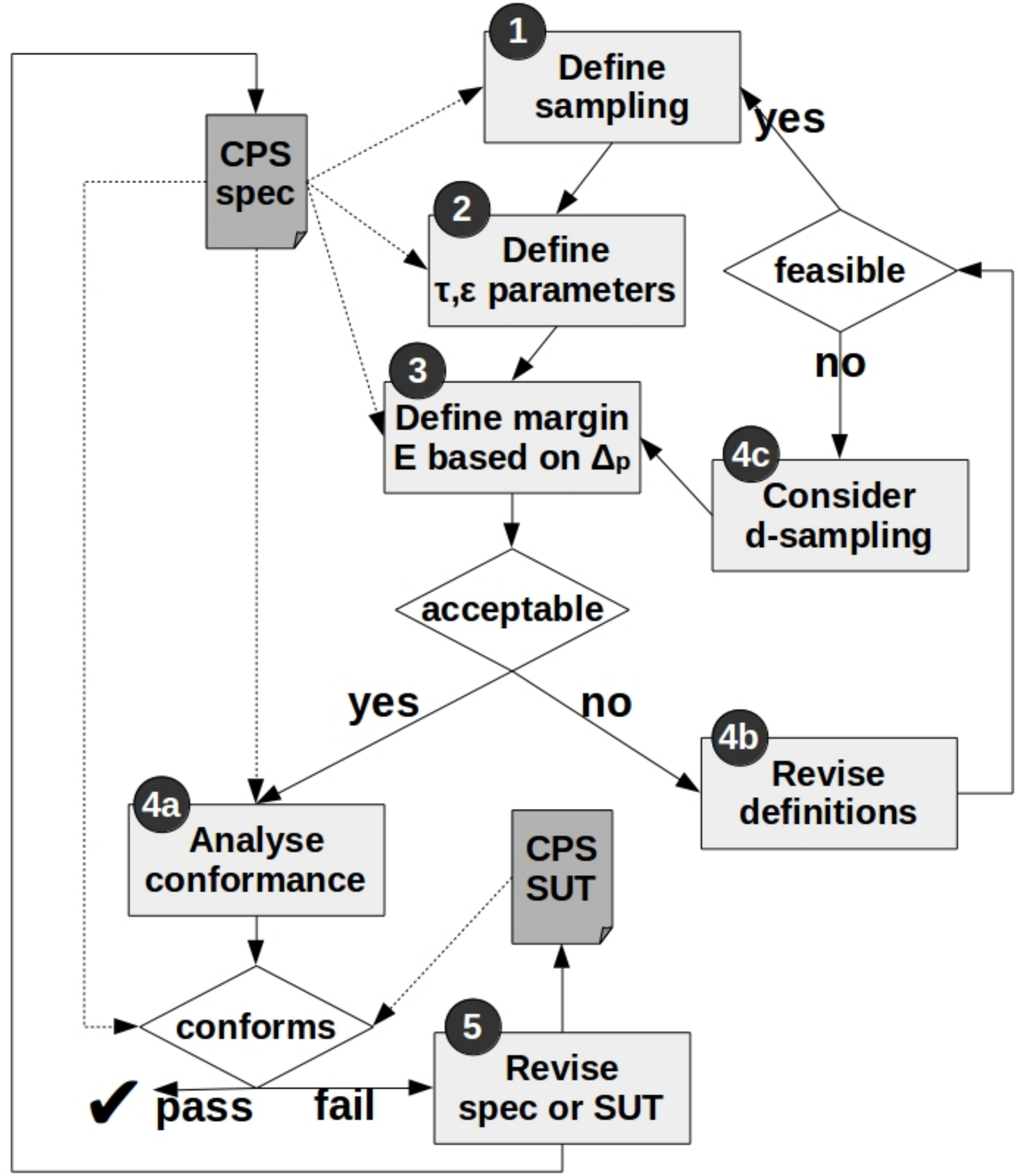
- Adapt the **sampling rate** if error margins are out of bounds, and iterate



[Althoff and Krogh, ICDC 2011]

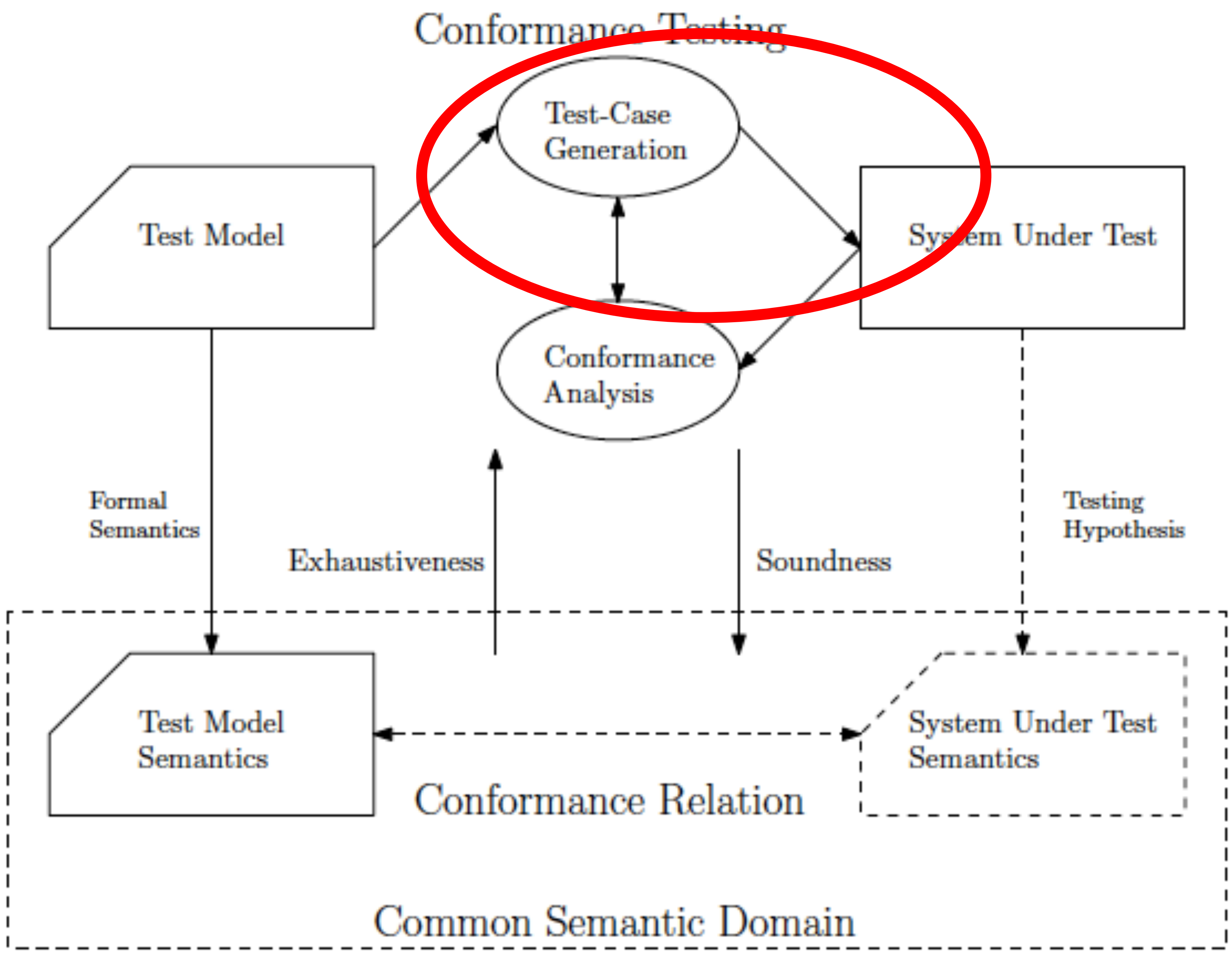
[Araujo, Carvalho, Mohaqeqi, MRM, and Sampaio, SCP 2018]

Process Sketch



[Araujo, Carvalho, MRM, Sampaio, and Taromirad, ICSTW 2017]

Model-Based Testing



Test-Case Generation: Test-Data Selection

First Objective: Maximising Critical Epsilon

Given two (target and control) signals
in the specification and
a fixed τ:

the **Critical Epsilon** is the **smallest ϵ** that makes
them (τ, ϵ)-conforming.

First Objective: Maximising Critical Epsilon

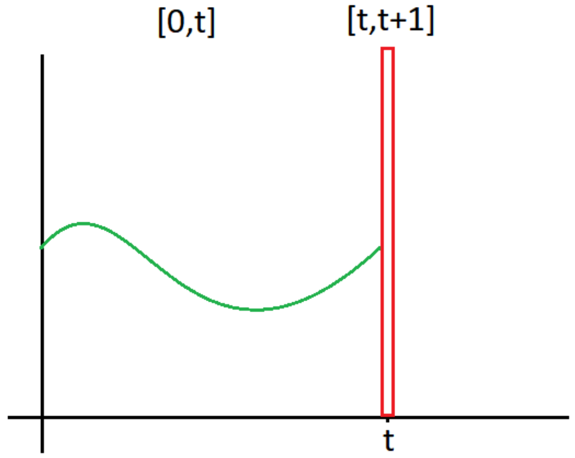
Idea: Search for inputs that maximise the spatial distance between reference and generated values.

Implementation: use Simulated Annealing to find the highest Critical Epsilon

Given an input from [0,t], we search for which input value at (t+1) generates the highest Critical Epsilon.

- Repeat this step until the end of the simulation.
- The initial input value (where t=0) must be given.

Drawback: algorithm might find unrealistic inputs.
Solution: Refine the model to disallow such inputs.



Multi-Objective Search: Coverage

- **Discrete state coverage**
  - SA guides the system towards a certain state.
  - Once in the state, switch the priority to find the highest Critical Epsilon.
  - Repeat this process for each discrete state.
- **Path coverage**
  - Prime paths coverage
  - Analogously, once the path is covered, switch the priority to find the highest CE.

Practical Evaluation

RQ 1:
Critical epsilon objective improves
fault detection capability significantly.

RQ 2:
Discrete state coverage also improves
fault detection capability, but
it is **less effective** than **critical epsilon**.

RQ 3:
Path coverage does not improve
fault detection capability (beyond state coverage).

Method: Mutation Analysis

Variable Negation

Variable Change

Constant Change

Constant Replacement

Statement Change

Delay Change

Relational Operator Replacement

Arithmetic Operator Replacement

Empirical Evaluation

Our prototype:

- Random test-data
- Search-based: single and multi-objective

https://github.com/hlsa/cps-conf-tool

S-Taliro:

- Simulated annealing
  (for minimising the robustness value)

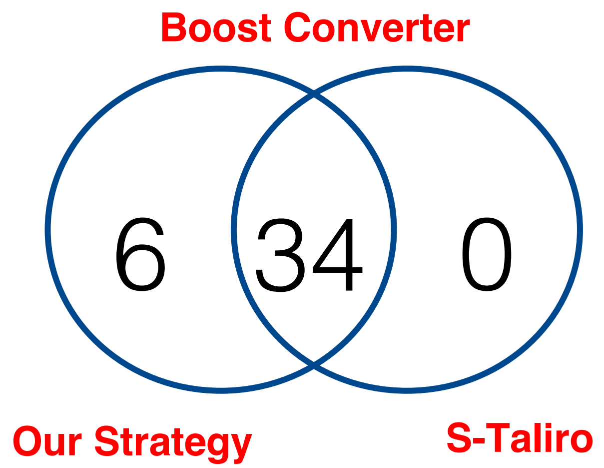
https://sites.google.com/a/asu.edu/s-taliro/

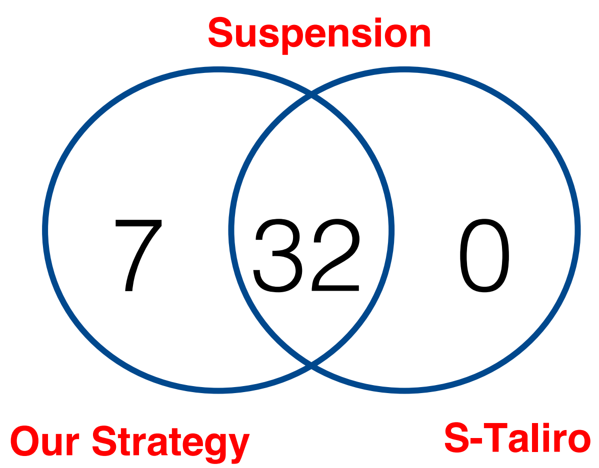
Mutation Analysis - Breakdown

Operator	Boost Converter	Suspension System
Variable Change	7/10	6/10
Constant Change	6/10	5/10
Variable Negation	5/5	5/5
Constant Replacement	5/5	5/5
Statement Change	4/5	4/5
Delay Change	3/5	4/5
Relational Operator Replacement	5/5	5/5
Arithmetic Operator Replacement	5/5	5/5

Mutation Analysis – Initial Results

Approach / Case Study	Boost Converter [1]	Suspension System [2]
Random Test Data	24/50	26/50
S-Taliro	34/50	32/50
Our Strategy	40/50	39/50





[1] - A Tool Prototype for Model-Based Testing of Cyber-Physical Systems, ICTAC 2015

[2] - Modelling and verification using linear hybrid automata: a case study, Müller, O., Stauner, T.

Mutation Analysis - Breakdown

Approach / Case Study	Boost Converter	Suspension
Critical Epsilon	34/50	32/50
Discrete State Coverage	40/50	39/50
Prime Paths Coverage	40/50	39/50
Total (Union)	40/50	39/50

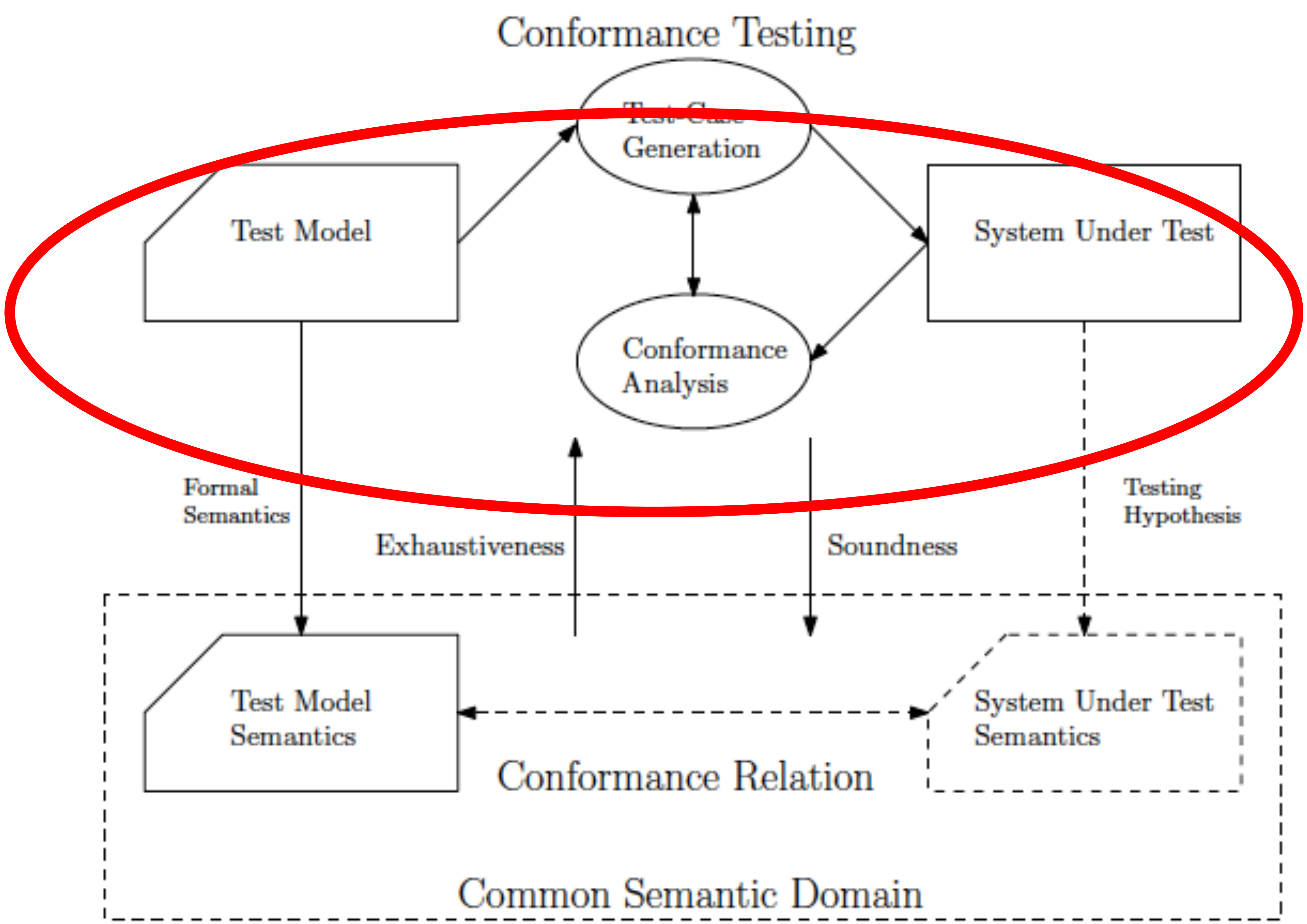
	Boost Converter	Suspension
Random Test Data	24/50	26/50
S-Taliro	34/50	32/50

Test-Date Selection: Efficiency

	Boost Converter		Suspension	
Critical Epsilon	1 tc	14 m	1 tc	17 m
Discrete State Coverage	4 tc	53 m	4 tc	70 m
Prime Path Coverage	11 tc	143 m	7 tc	188 m

	Boost Converter		Suspension	
Random Test Data	1 tc	1 s	1 tc	1 s
S-Taliro	1 tc	8 m	1 tc	11 m

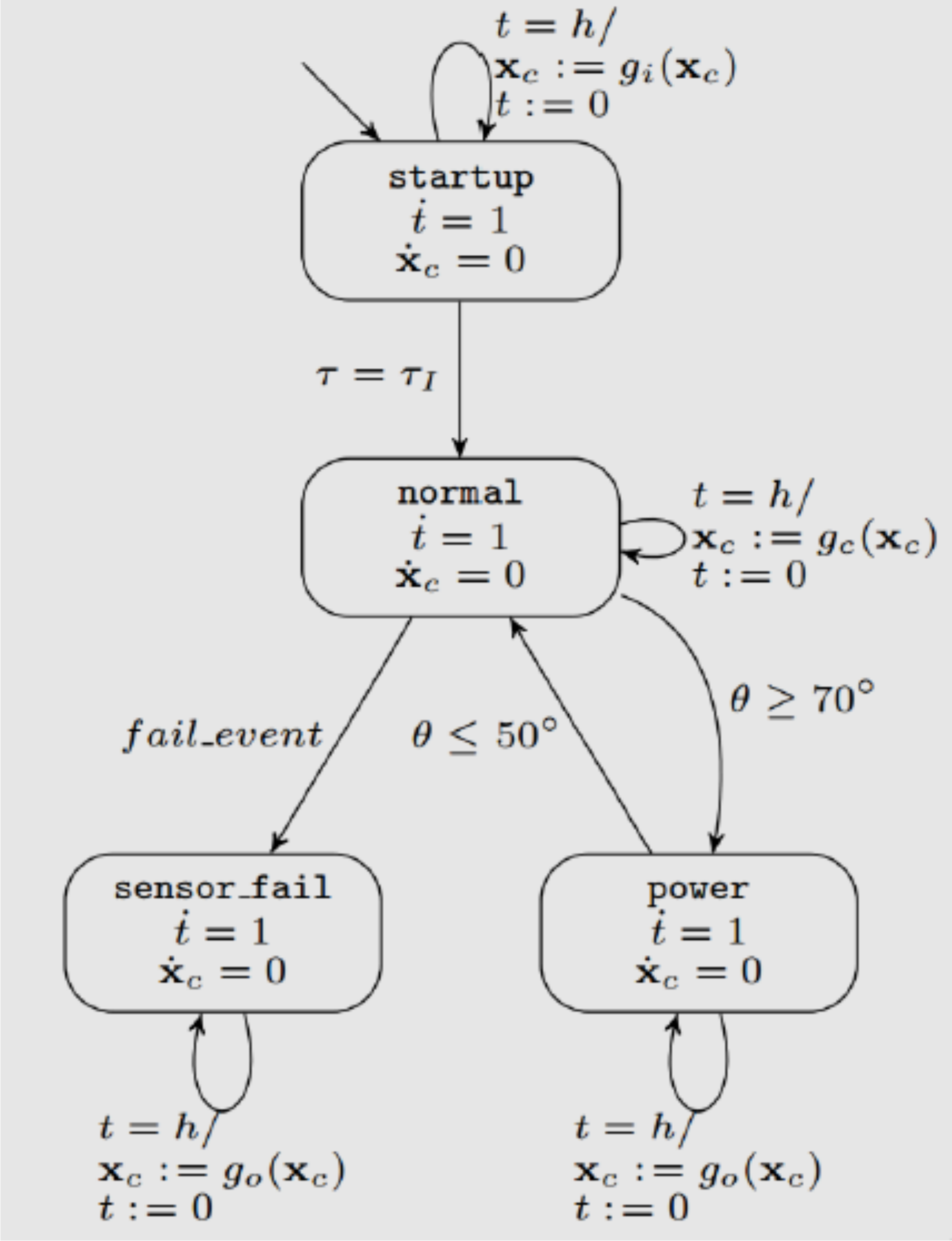
Model-Based Testing



Case Studies

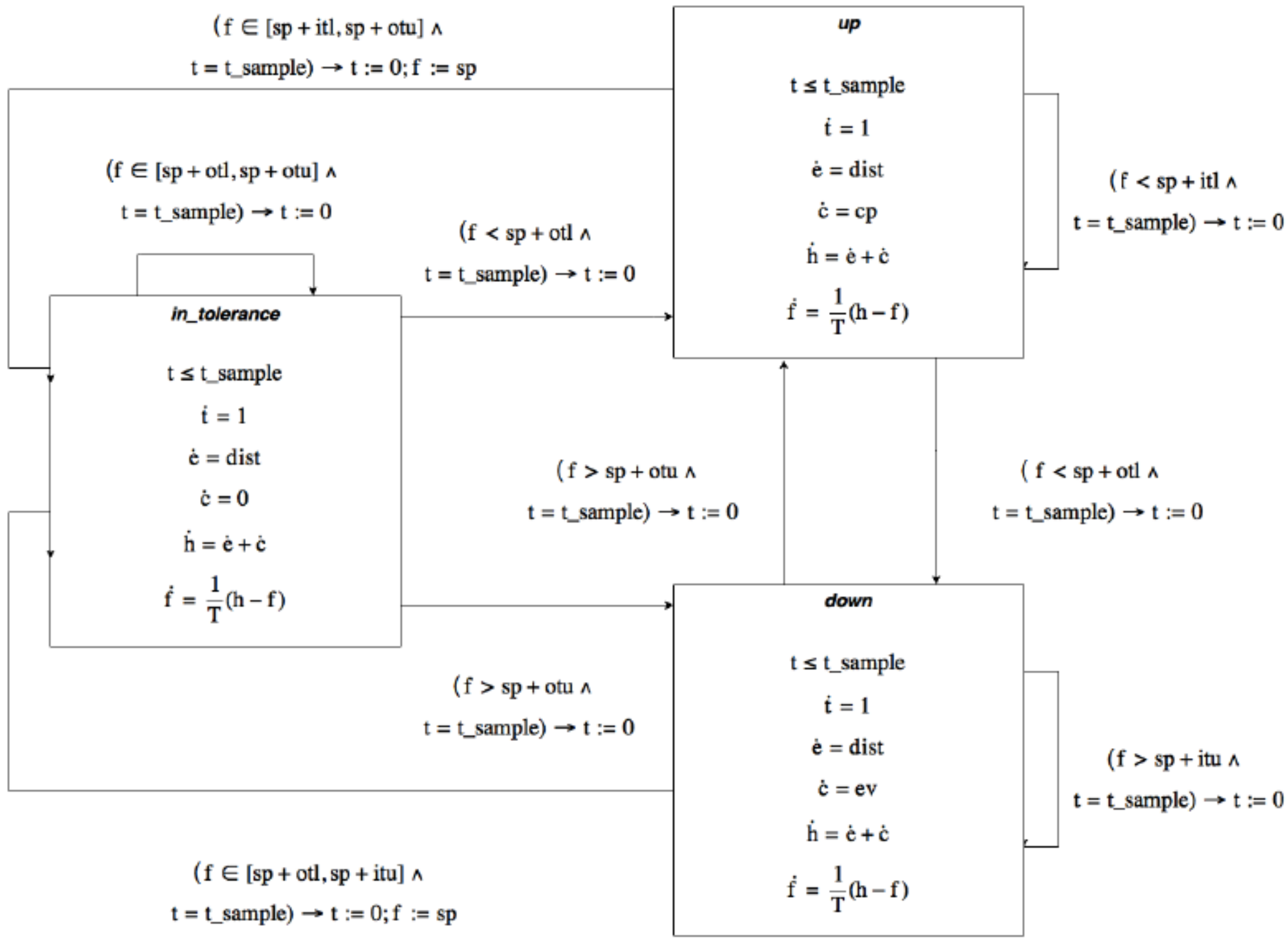
- Engine fuel controller [Jin et al. HSCC 2014]
- Pneumatic suspension system [Müller and Stauner, MCMD 2000]
- **Connected platoon controller**
- **NOx emission scandal and software doping**

Case Study 1: Engine Fuel Controller



[Jin et al. HSCC 2014]

Case Study 2: Pneumatic Suspension System

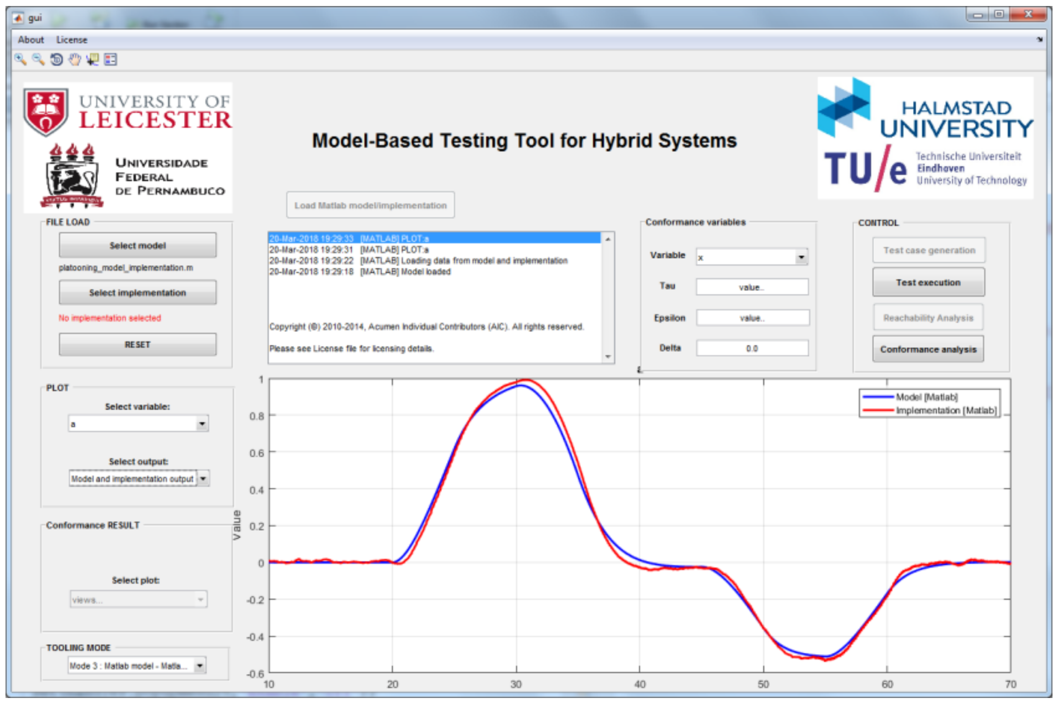


[Müller and Stauner, MCMD 2000]

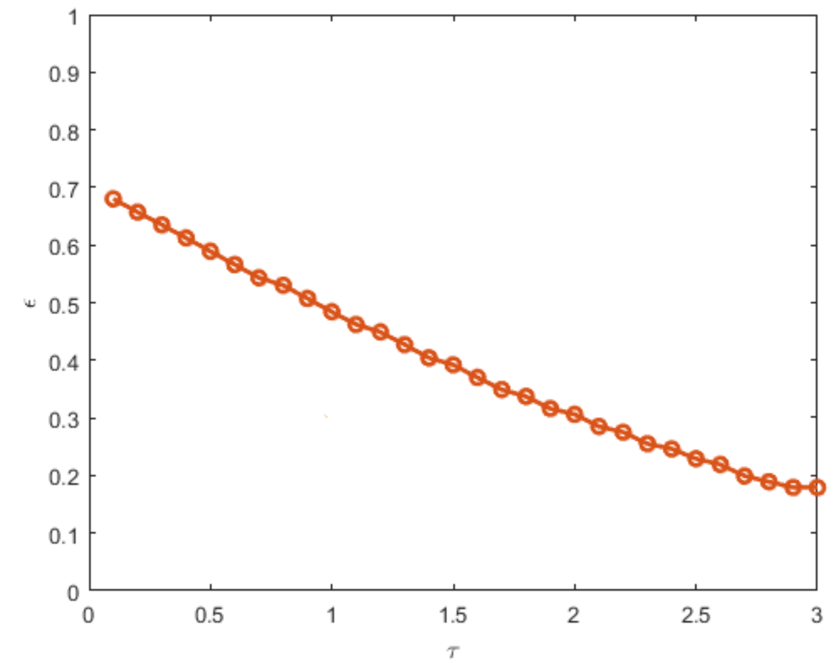
Case Study 3:
Connected Platoons

Conformance testing

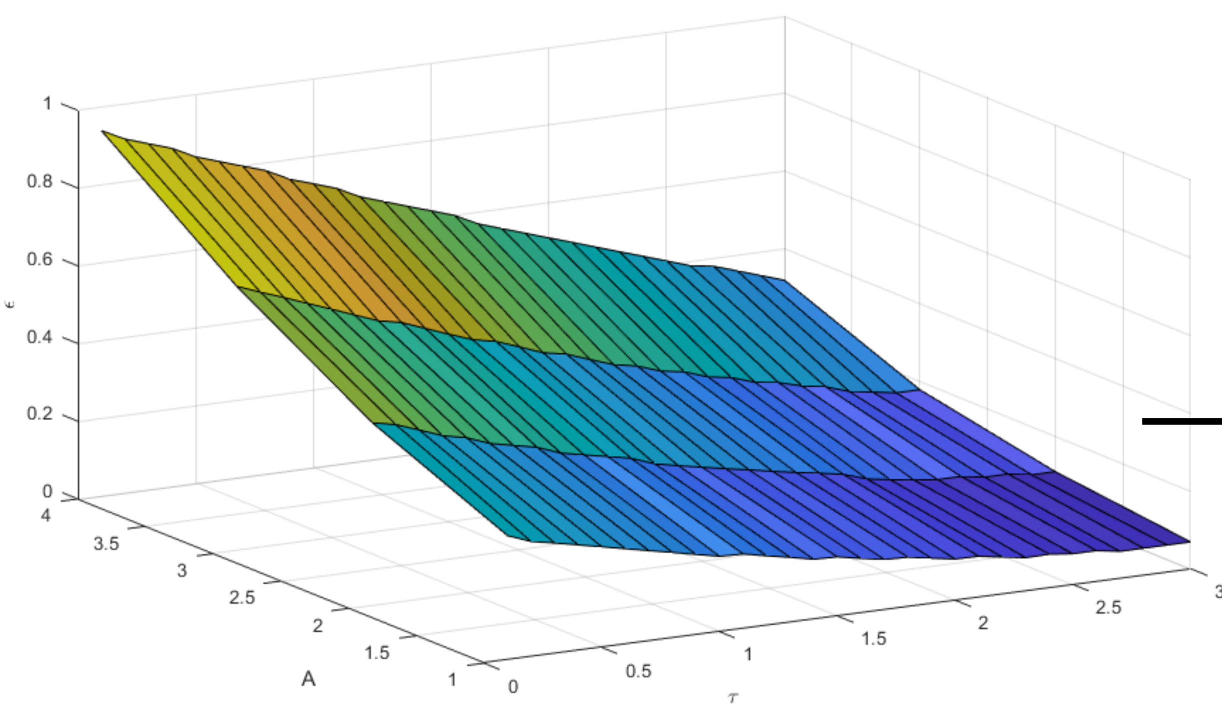
Single point



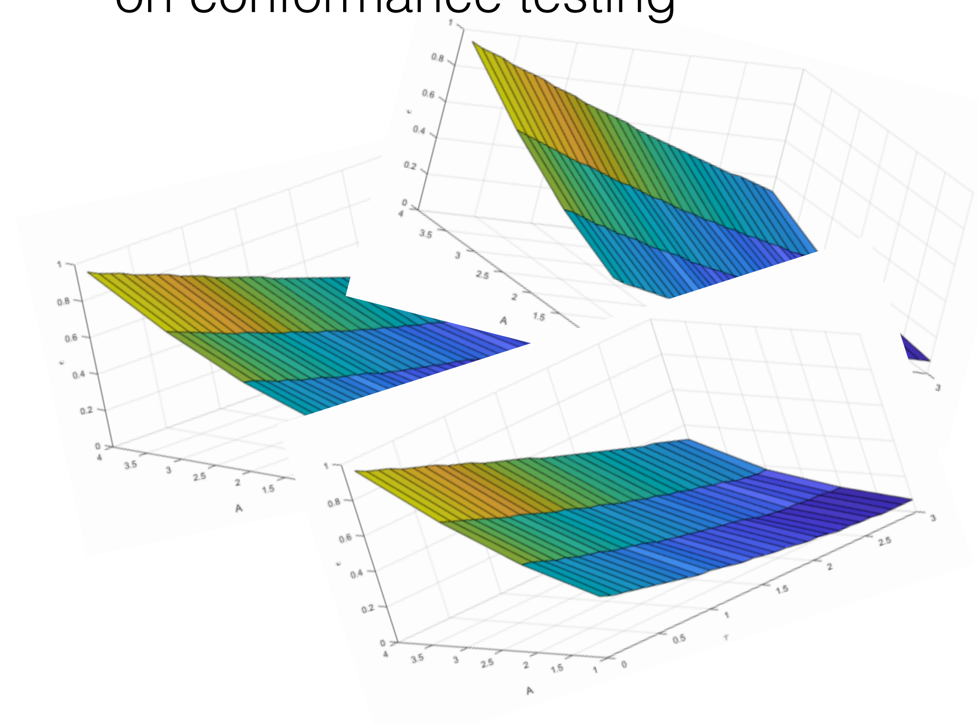
Multiple values



Multiple values with different inputs

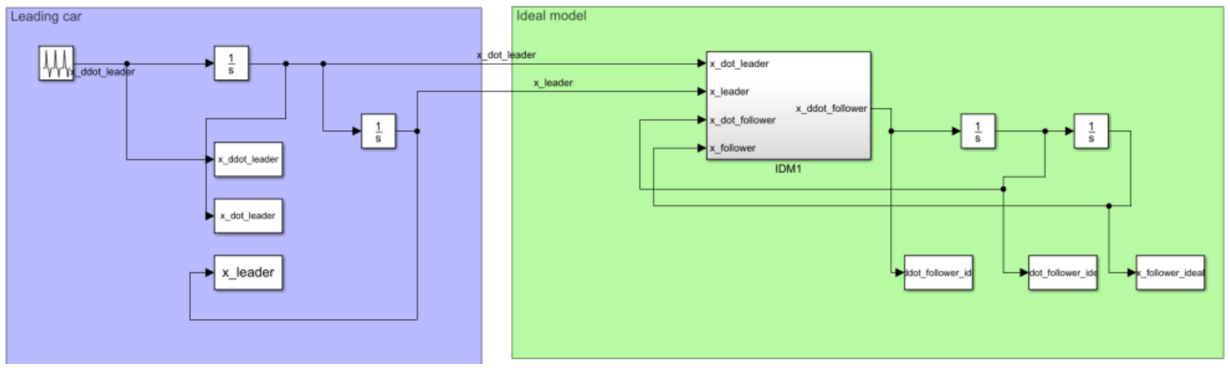


Classifying parameters based on conformance testing

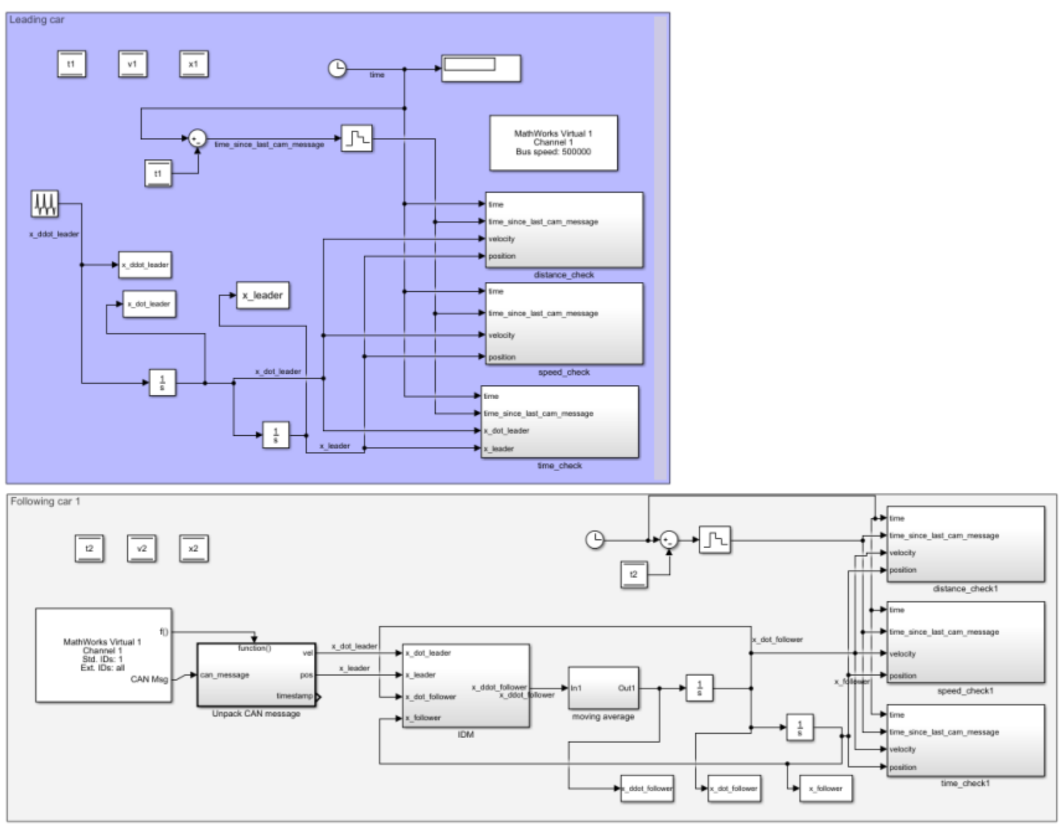


Models

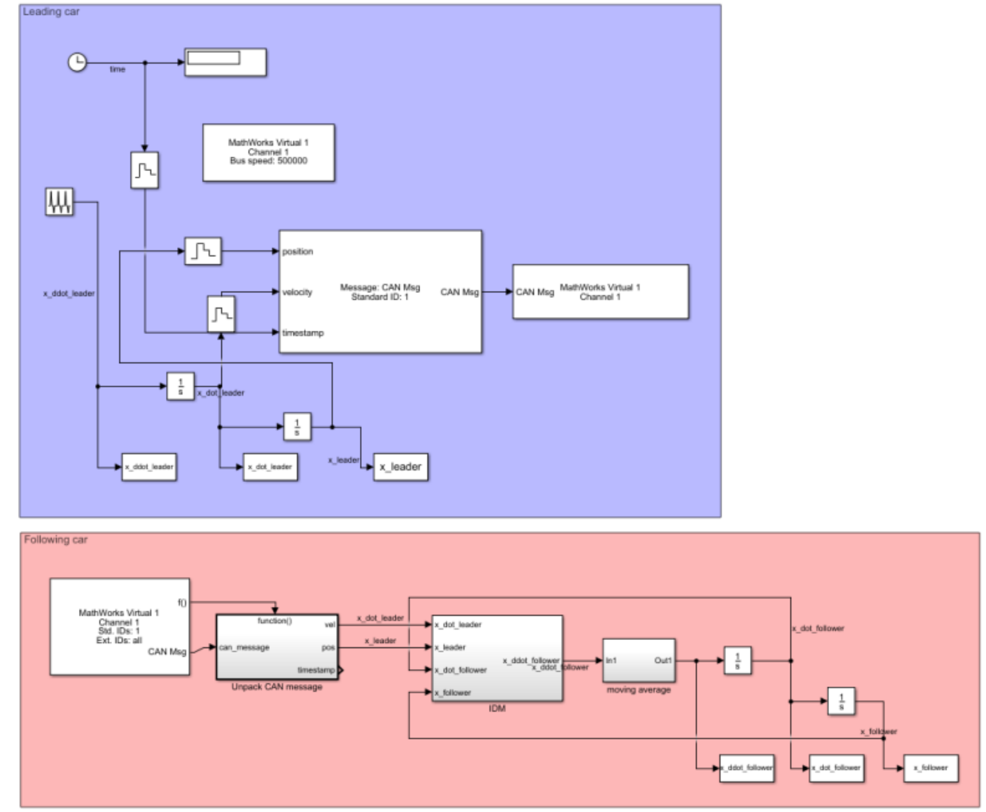
Ideal model



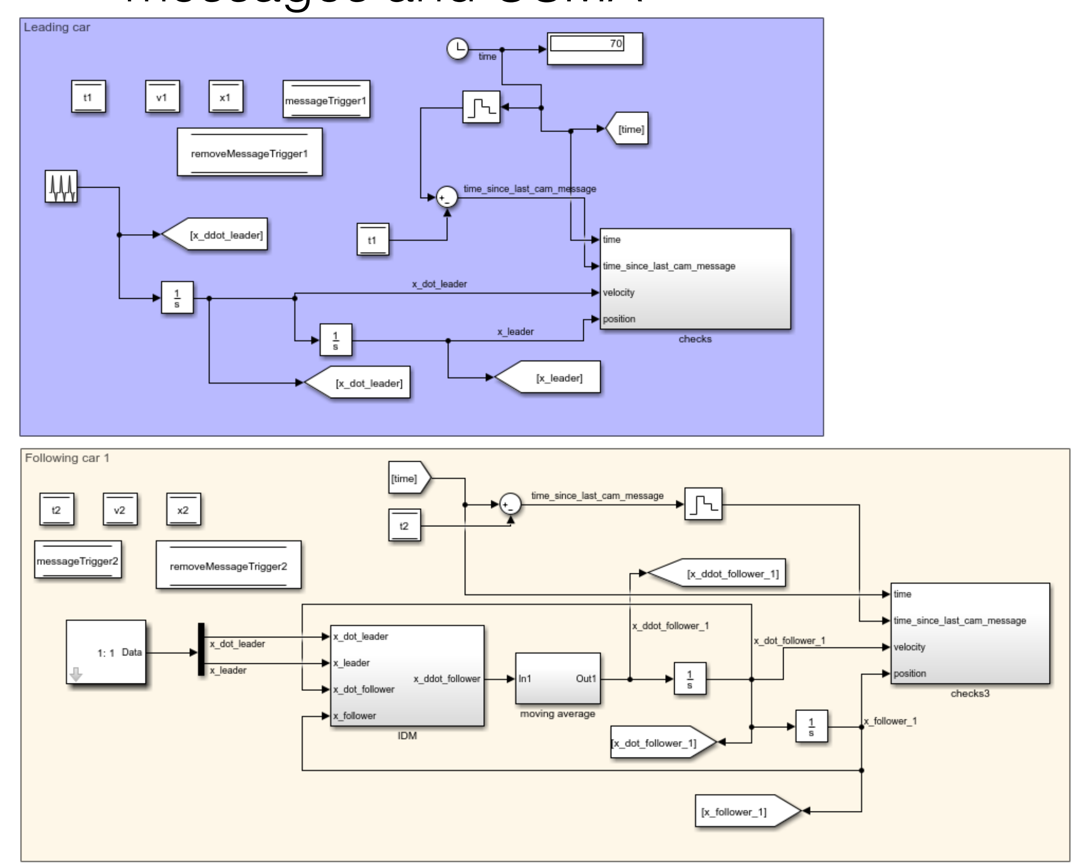
Model with triggered CAM messages



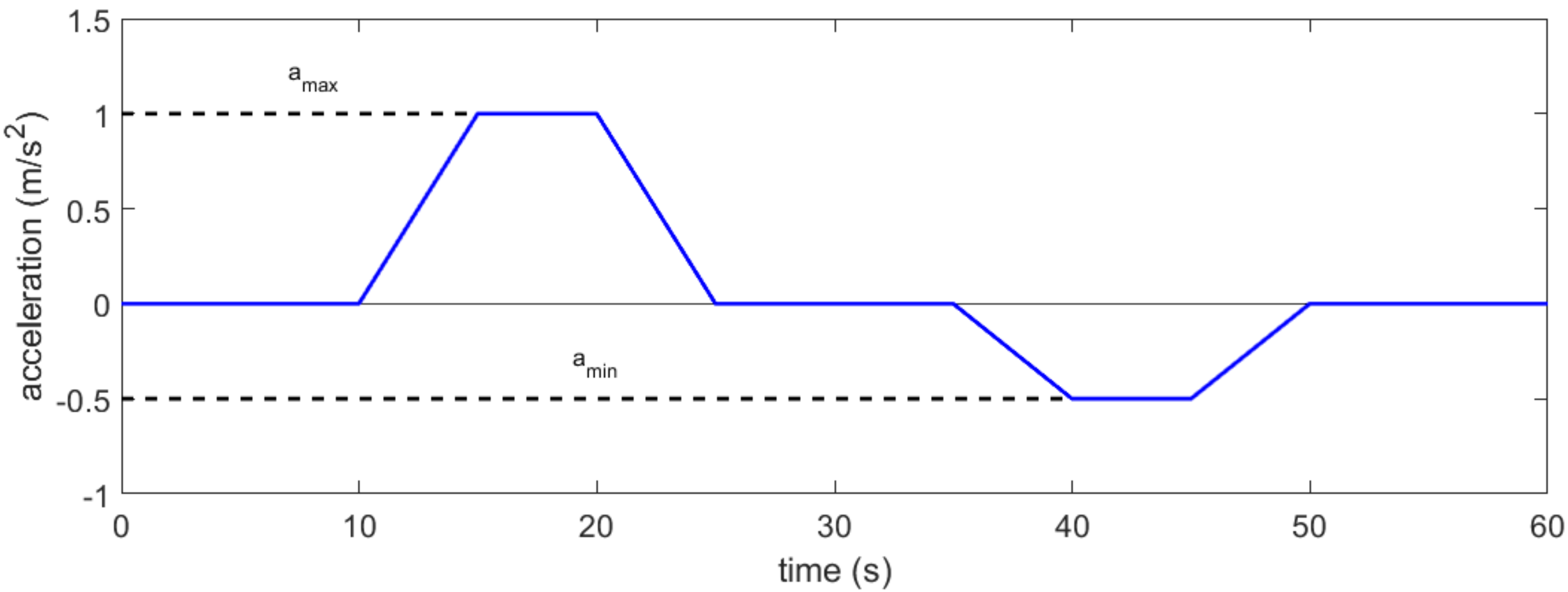
Model with CAM messages



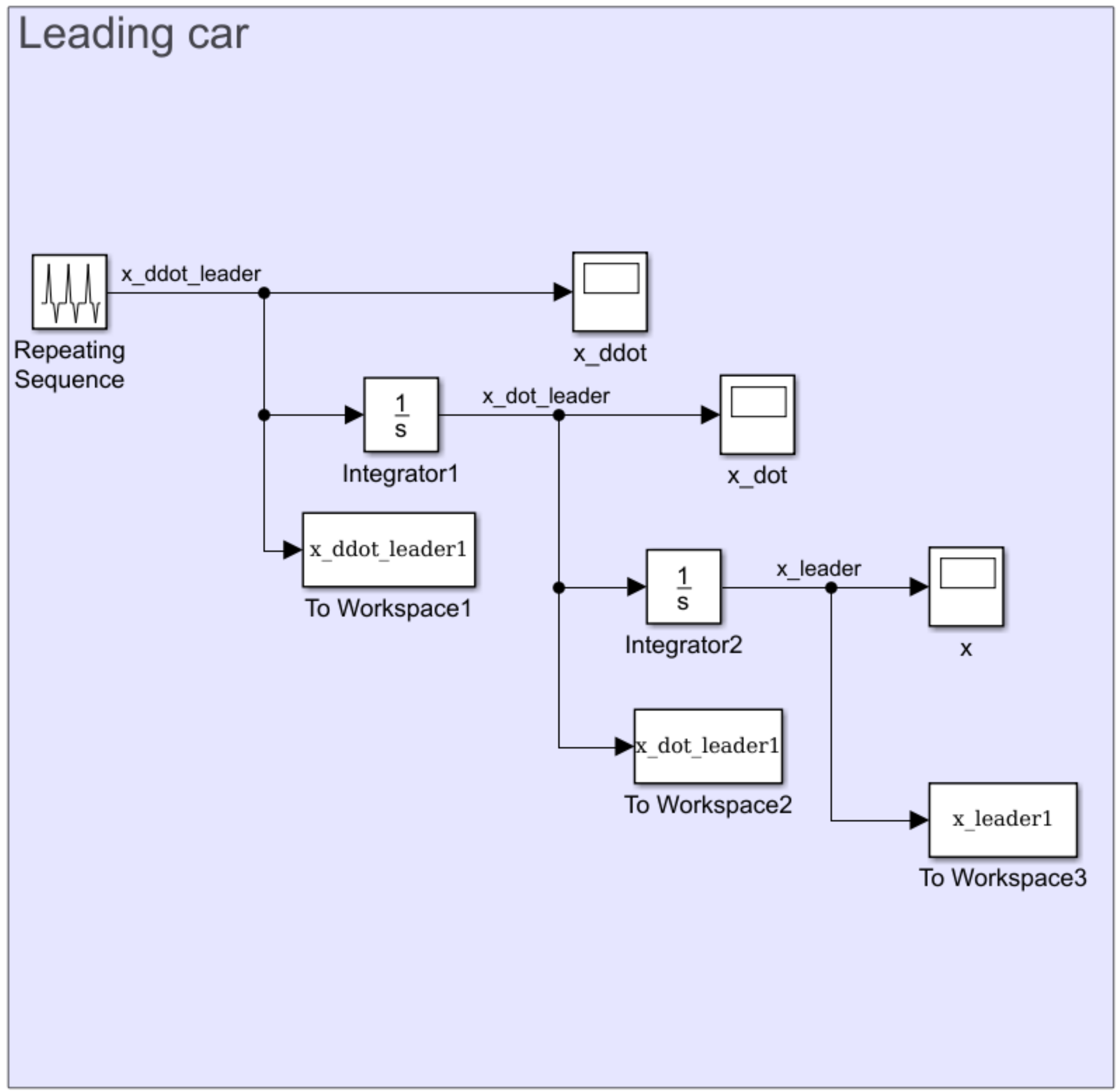
Model with triggered CAM messages and CSMA

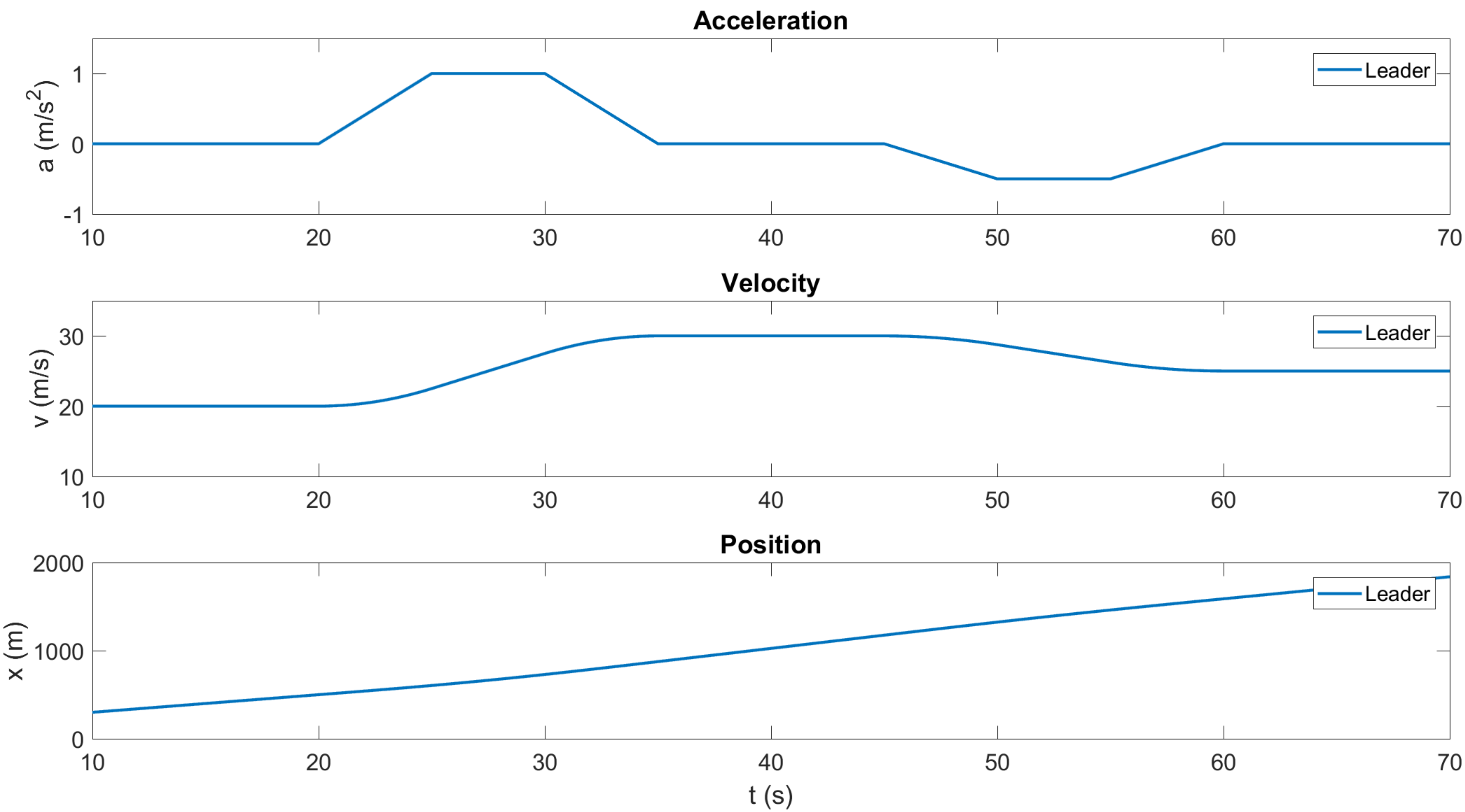


Parameterised acceleration pattern of the leading car



Simulink Model: Leading Car

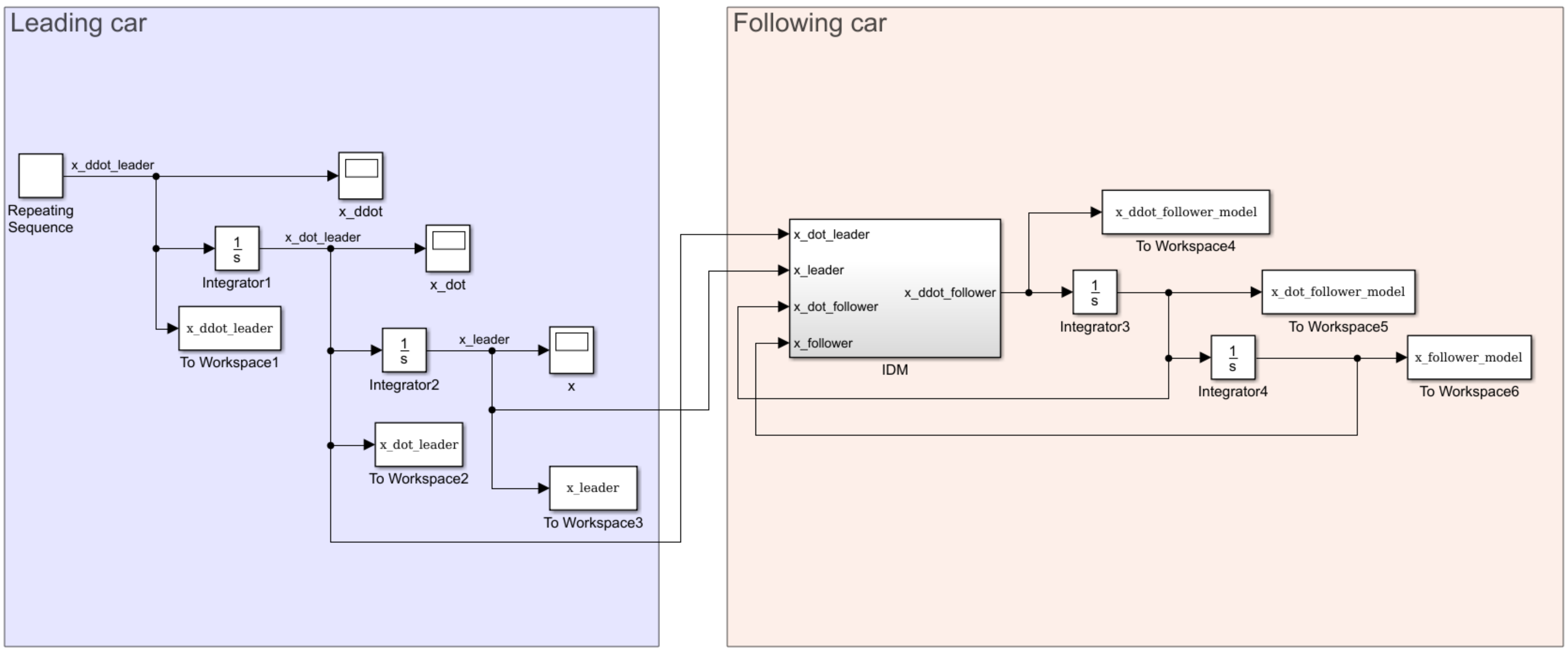




Acceleration
a (m/s^2)
1
0
-1
10
20
30
40
50
60
70
Leader
Velocity
v (m/s)
30
20
10
Position
x (m)
2000
1000
0
t (s)

Ideal car following model (not connected)

Leading and following cars model



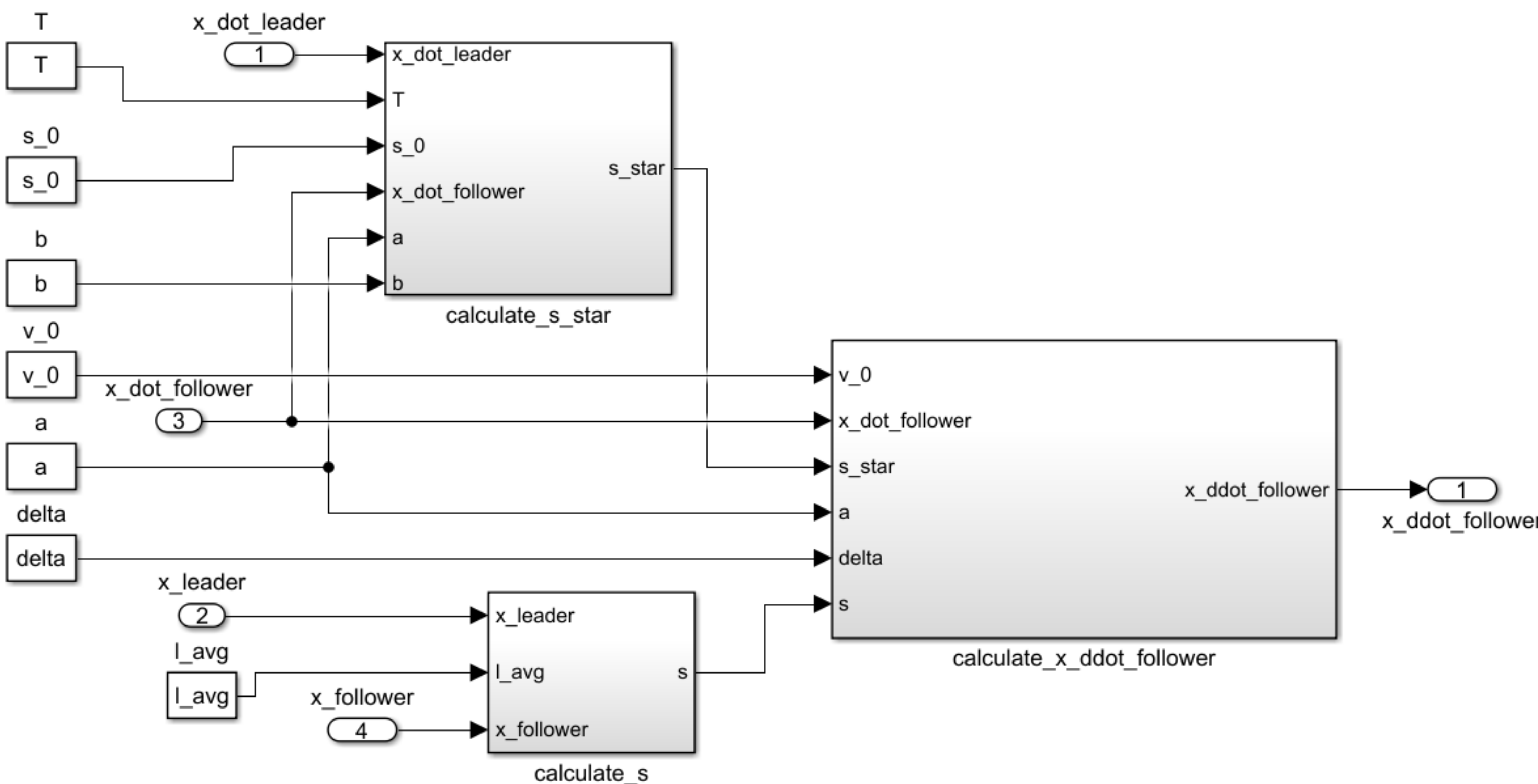
Intelligent Driver Model

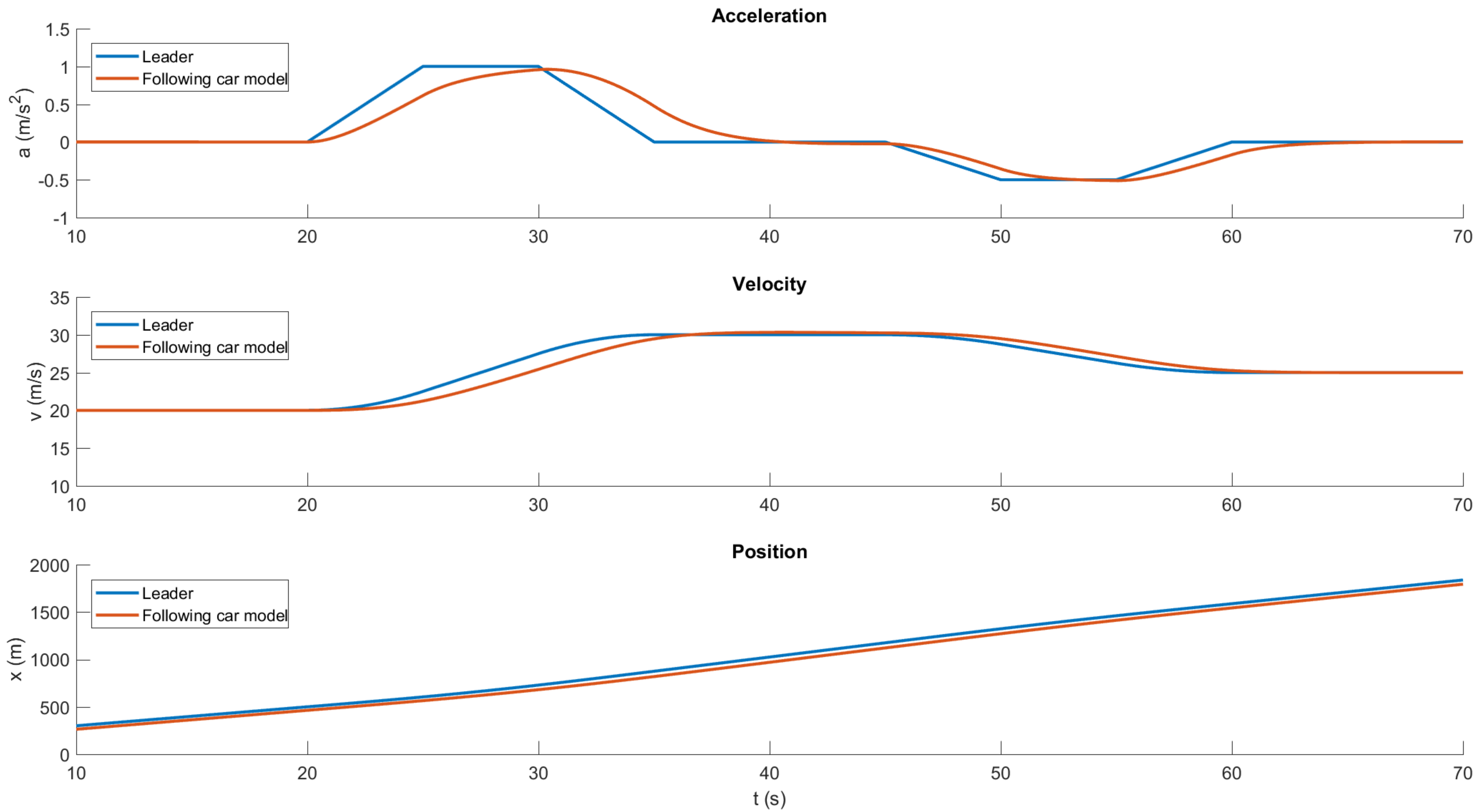
$$a_{IDM}(s,v,\Delta v) = \frac{dv}{dt} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*(v,\Delta v)}{s}\right)^2\right]$$

$$s^*(v,\Delta v) = s_0 + vT + \frac{v\Delta v}{2\sqrt{ab}}$$

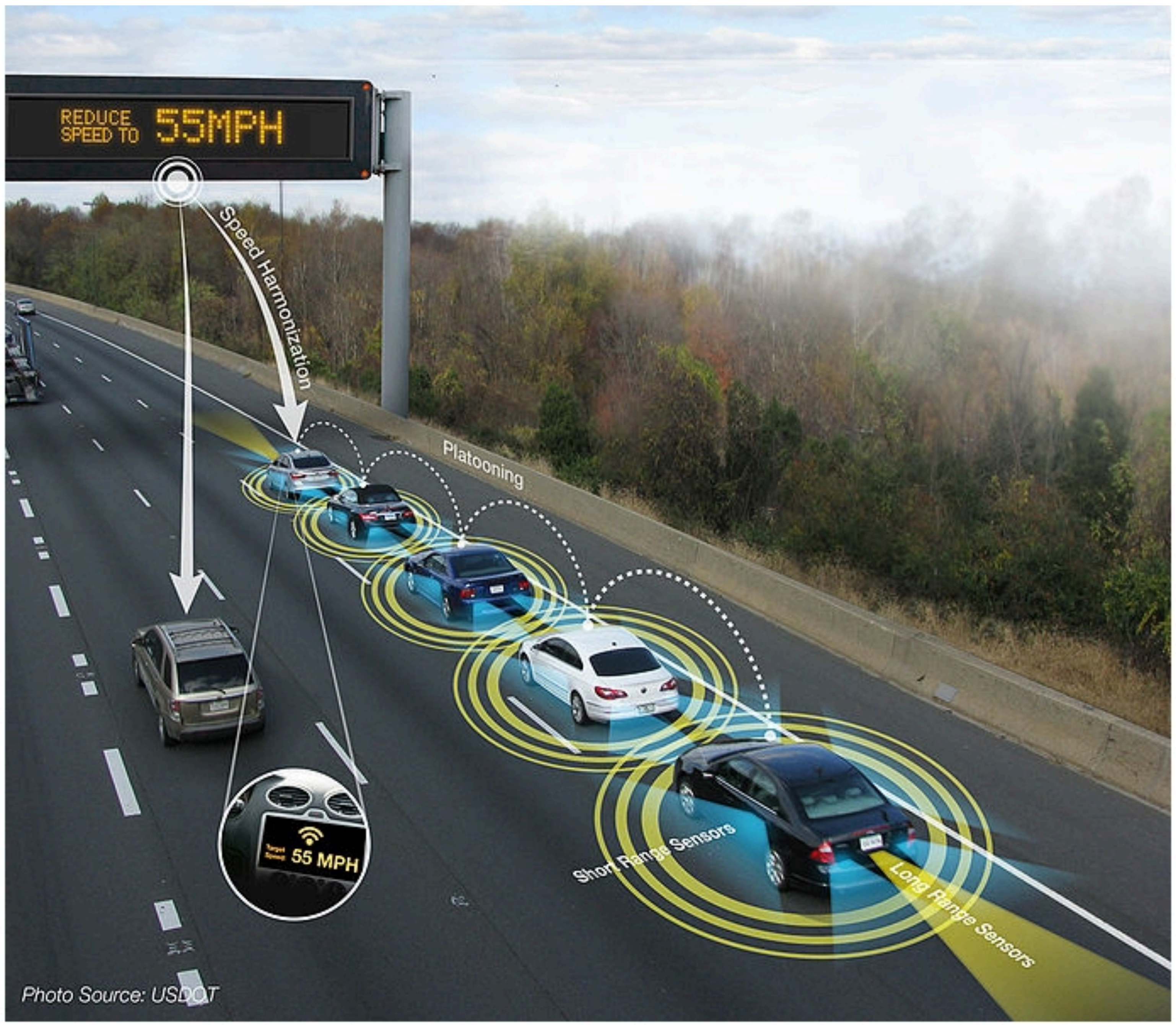
Parameter	**Description**	**Car**	**Truck**
v_0	Desired speed	120 km/h	85 km/h
δ	Free acceleration exponent	4	4
T	Desired time gap	1.5 s	2.0 s
s_0	Jam distance	2.0 m	4.0 m
a	Maximum acceleration	1.4 m/s^2	1.4 m/s^2
b	Desired deceleration	2.0 m/s^2	2.0 m/s^2

IDM model

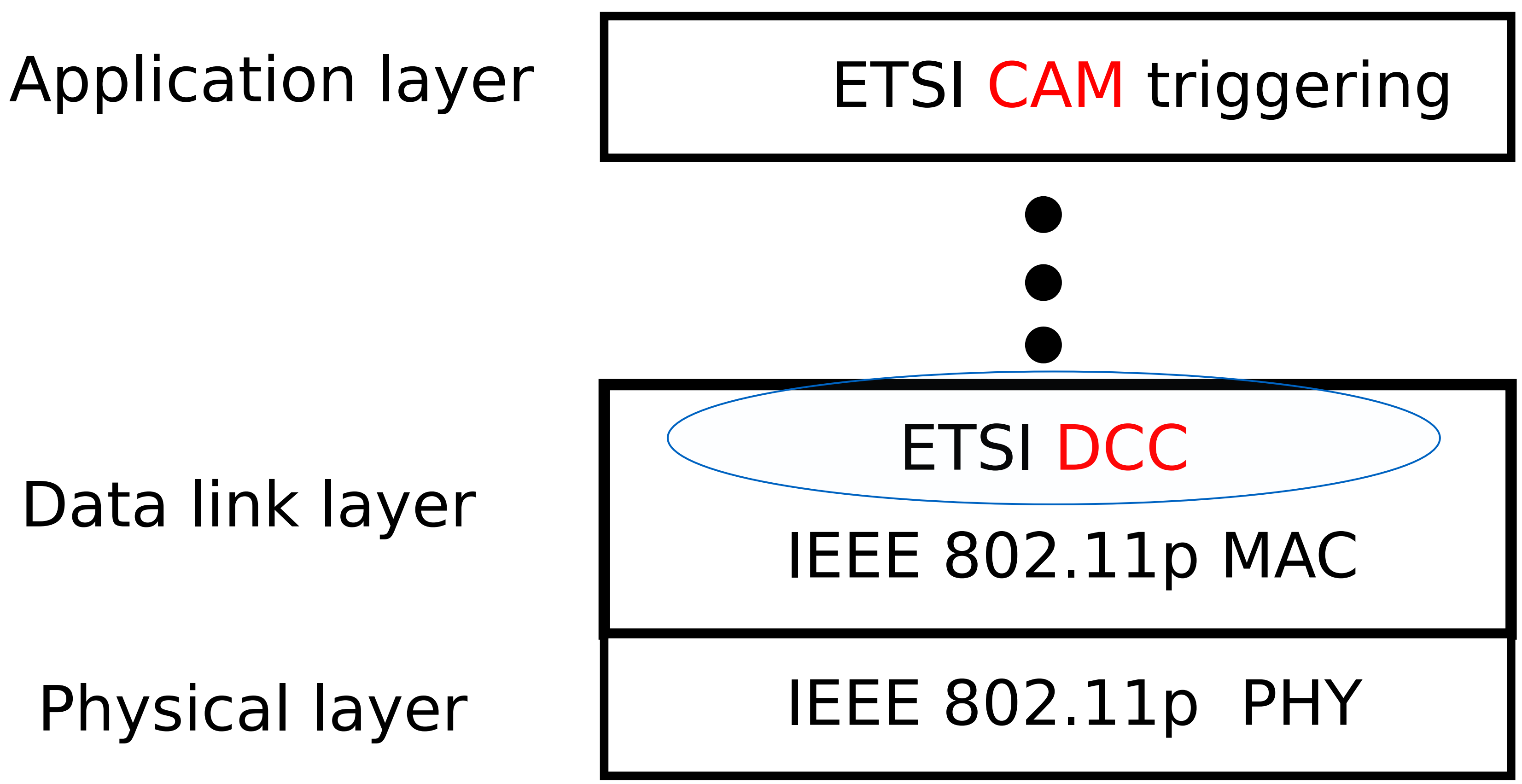

T
T
x_dot_leader
1
s_0
s_0
b
b
v_0
v_0
x_dot_follower
3
a
a
delta
delta
x_leader
2
l_avg
l_avg
x_follower
4
x_dot_leader
T
s_0
x_dot_follower
a
b
s_star
calculate_s_star
v_0
x_dot_follower
s_star
a
delta
s
x_ddot_follower
calculate_x_ddot_follower
1
x_ddot_follower
x_leader
l_avg
x_follower
s
calculate_s


Acceleration
Leader
Following car model
a (m/s^2)
1.5
1
0.5
0
-0.5
-1
10
20
30
40
50
60
70
Velocity
Leader
Following car model
v (m/s)
35
30
25
20
15
10
10
20
30
40
50
60
70
Position
Leader
Following car model
x (m)
2000
1500
1000
500
0
10
20
30
40
50
60
70
t (s)

Car following implementation (connected)



U.S. Department of Transportation http://www.its.dot.gov/image_gallery/image36.htm

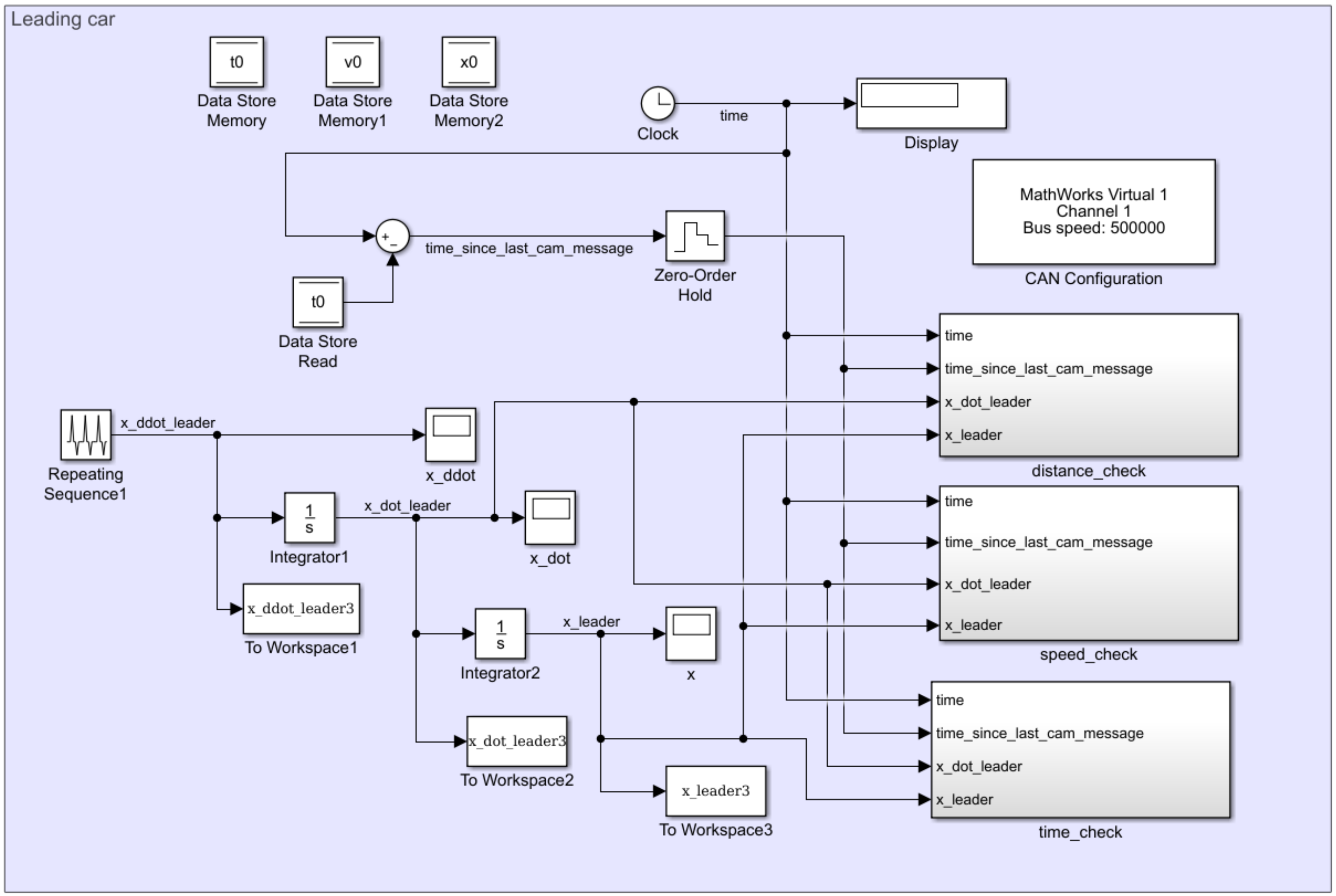

Application layer
ETSI CAM triggering
Data link layer
ETSI DCC
IEEE 802.11p MAC
Physical layer
IEEE 802.11p PHY

CAMs kinematic rules

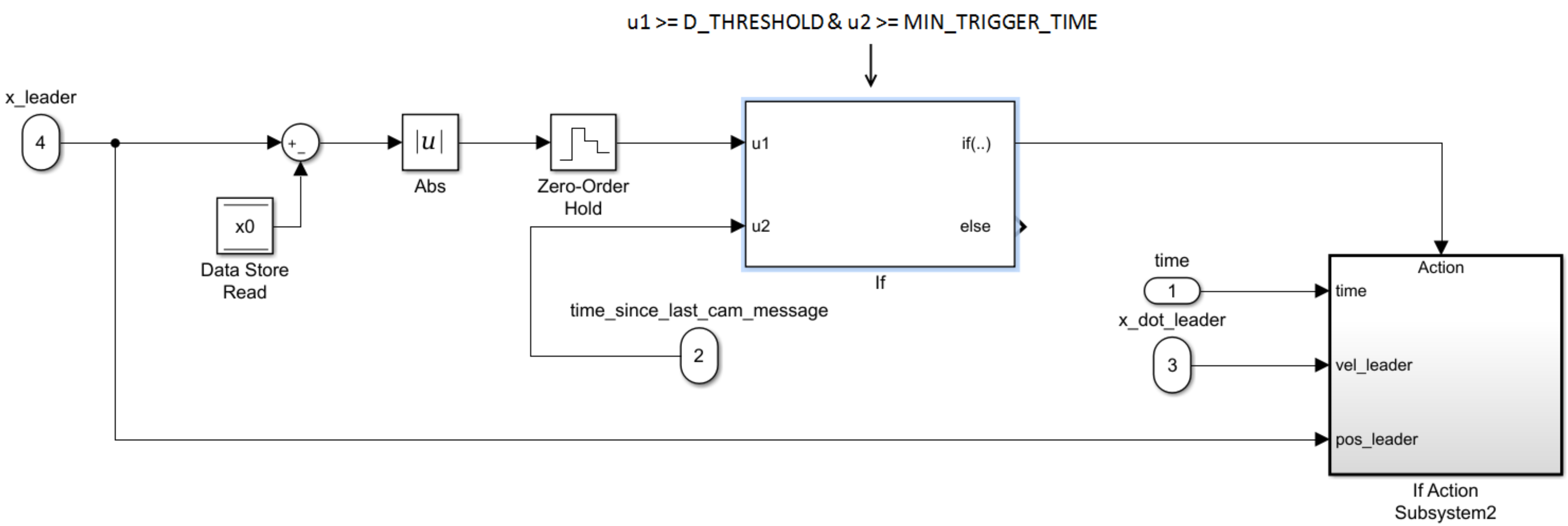
CAM shall be triggered in one of two cases:

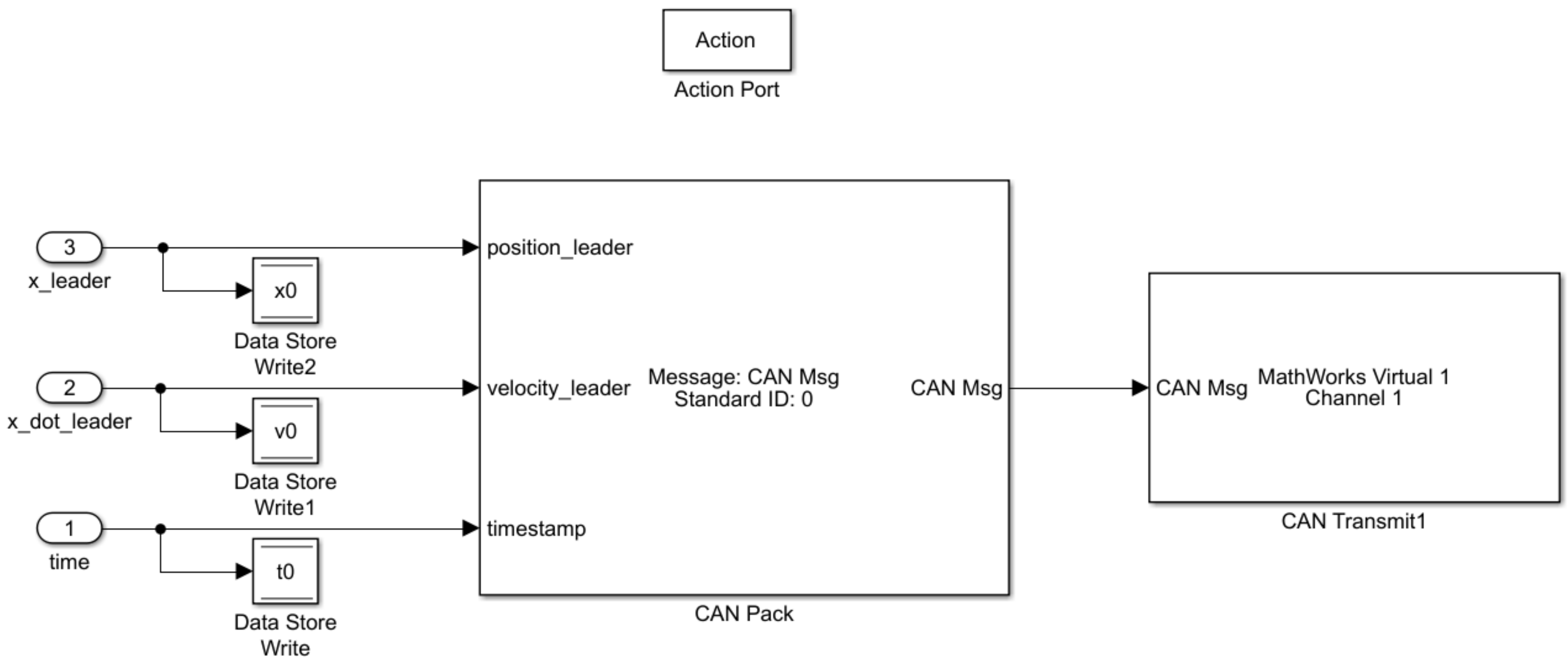
- The time elapsed since the last CAM generation **> 1000 ms**.
- The time elapsed since the last CAM generation **> 100 ms and** any of the following events has occurred:
    1. the absolute difference between the current **position** of the vehicle and its position included in the previous CAM **> 4 m**;
    2. the absolute difference between the current **speed** and the speed included in the previous CAM **> 0.5 m/s**;
    3. the absolute difference between the current **direction** of the vehicle and the direction included in the previous CAM **> 4°**.

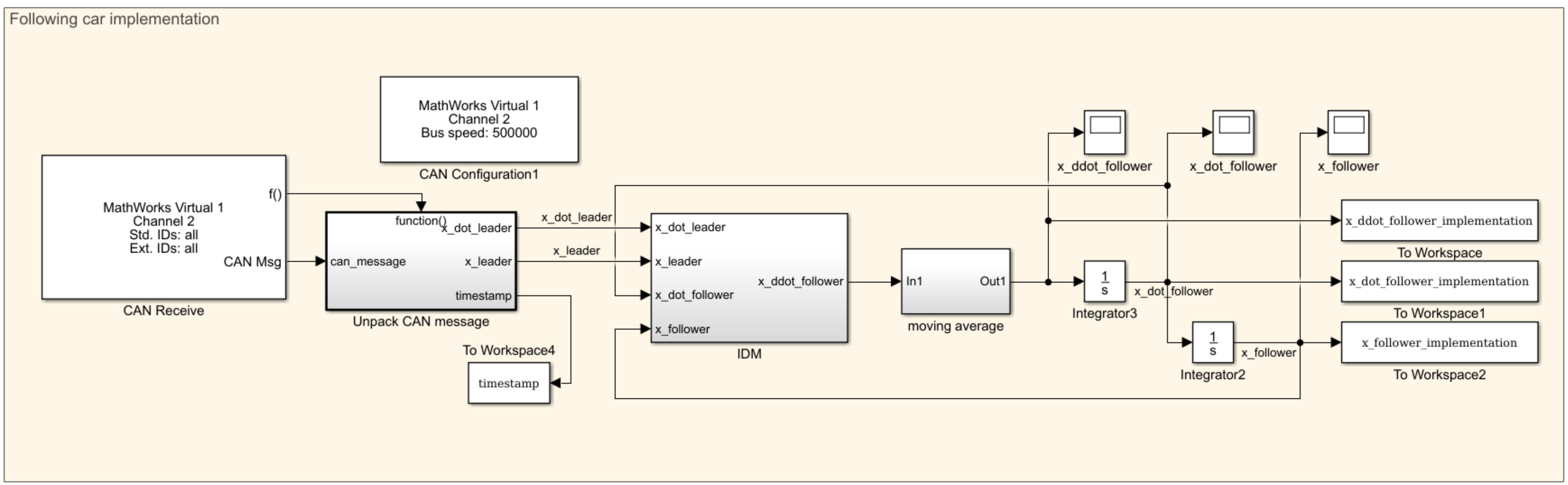
Leading car for implementation using ETSI-DCC protocol

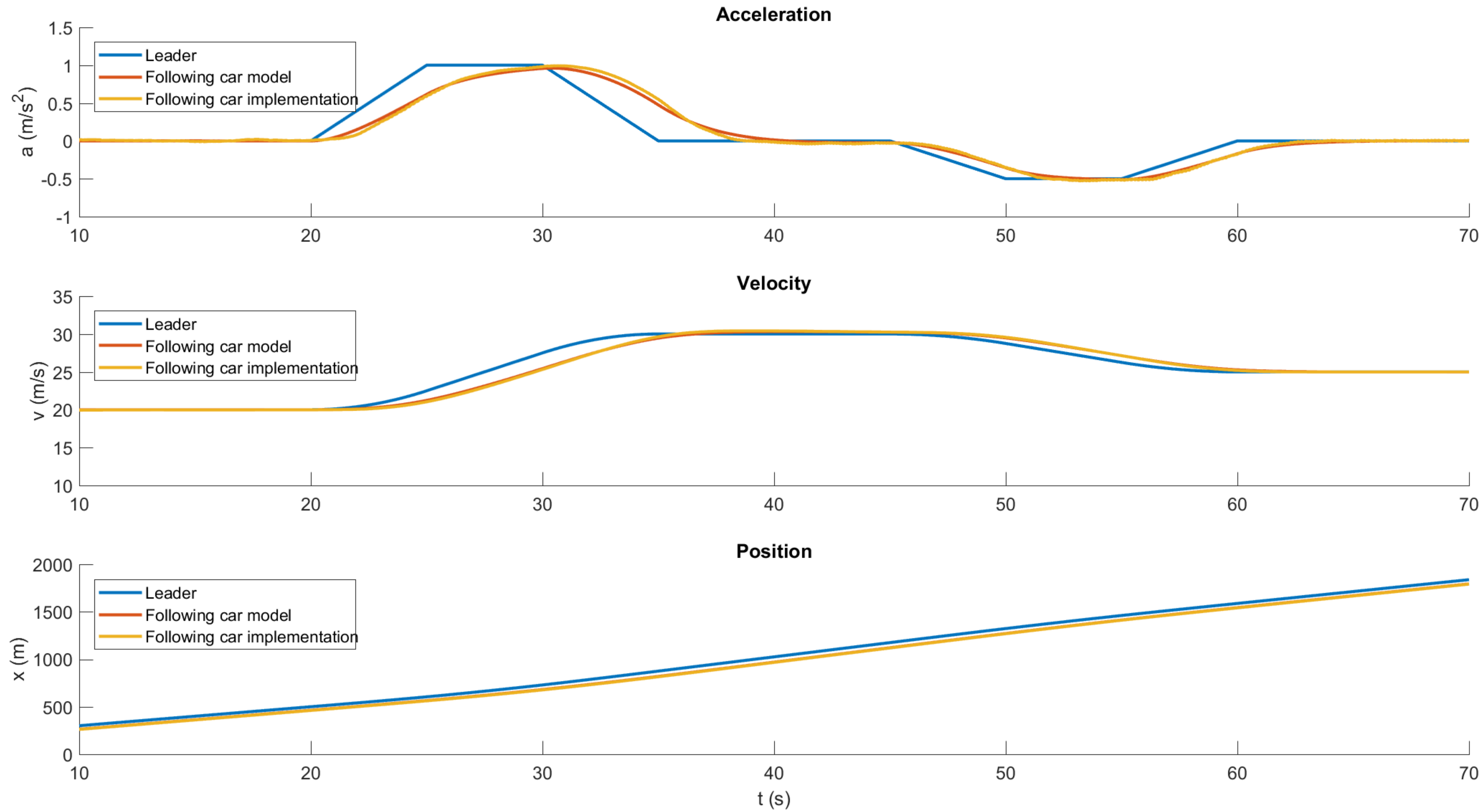


Distance check




Action
Action Port
3
x_leader
x0
Data Store Write2
2
x_dot_leader
v0
Data Store Write1
1
time
t0
Data Store Write
position_leader
velocity_leader
timestamp
Message: CAN Msg
Standard ID: 0
CAN Msg
CAN Pack
CAN Msg
MathWorks Virtual 1
Channel 1
CAN Transmit1


Following car implementation
MathWorks Virtual 1
Channel 2
Bus speed: 500000
CAN Configuration1
MathWorks Virtual 1
Channel 2
Std. IDs: all
Ext. IDs: all
f()
CAN Msg
CAN Receive
function()
can_message
x_dot_leader
x_leader
timestamp
Unpack CAN message
To Workspace4
timestamp
x_dot_leader
x_leader
x_dot_leader
x_leader
x_dot_follower
x_follower
x_ddot_follower
IDM
In1
Out1
moving average
x_ddot_follower
x_dot_follower
x_follower
Integrator3
x_dot_follower
Integrator2
x_follower
x_ddot_follower_implementation
To Workspace
x_dot_follower_implementation
To Workspace1
x_follower_implementation
To Workspace2



Acceleration
a (m/s^2)
Leader
Following car model
Following car implementation
Velocity
v (m/s)
Leader
Following car model
Following car implementation
Position
x (m)
Leader
Following car model
Following car implementation
t (s)

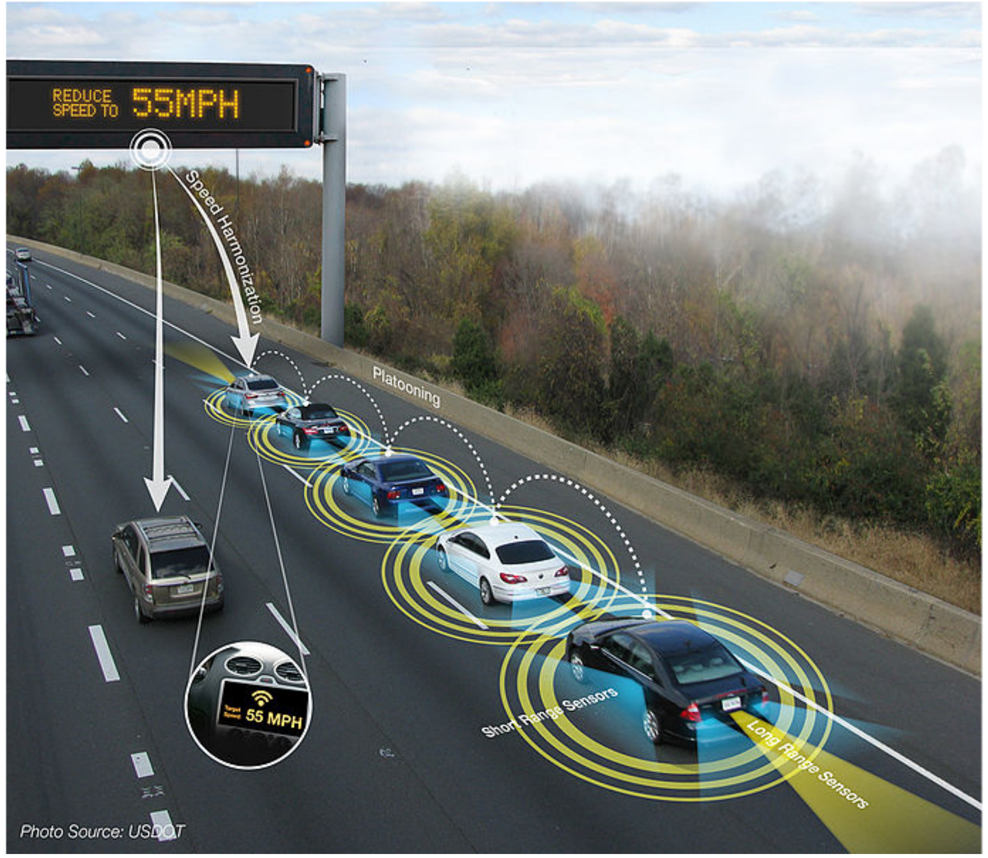
Conformance testing

Not connected

conform(τ, ε)?

=

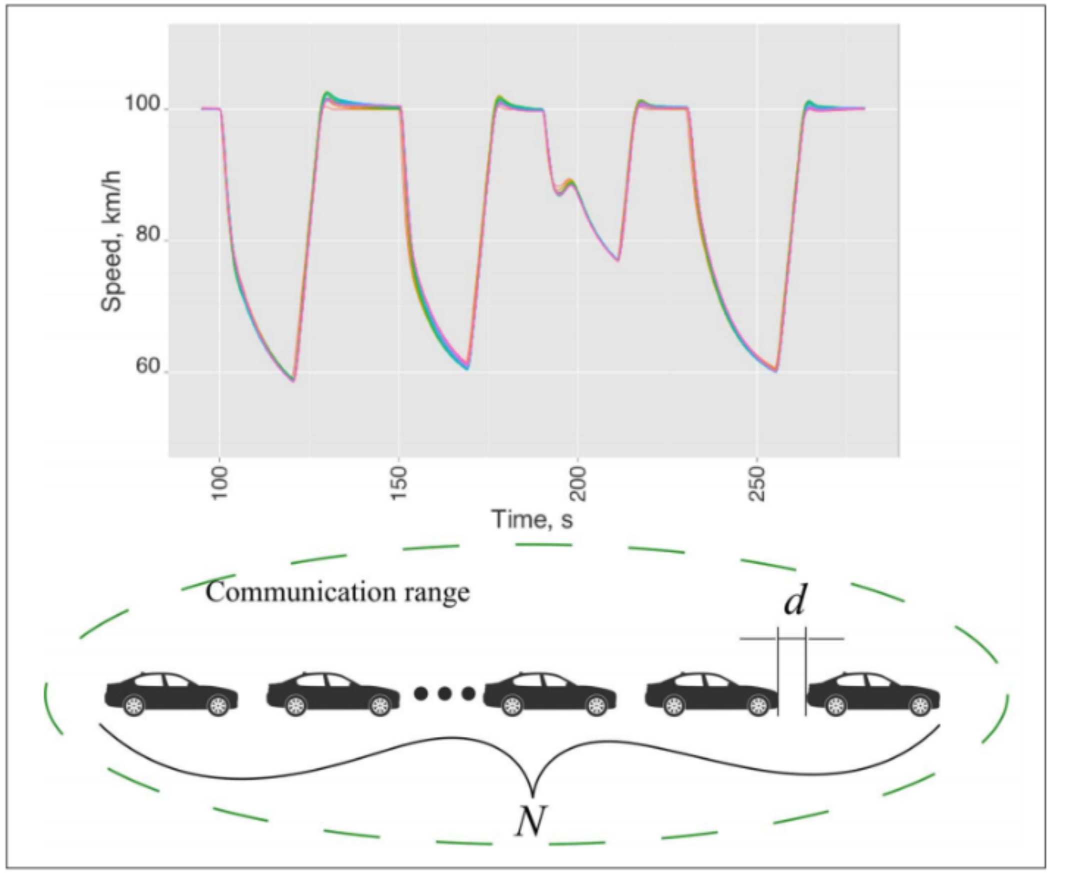
Connected



Goals

- Find speed profiles for the leader such that the followers cannot keep a safe distance.

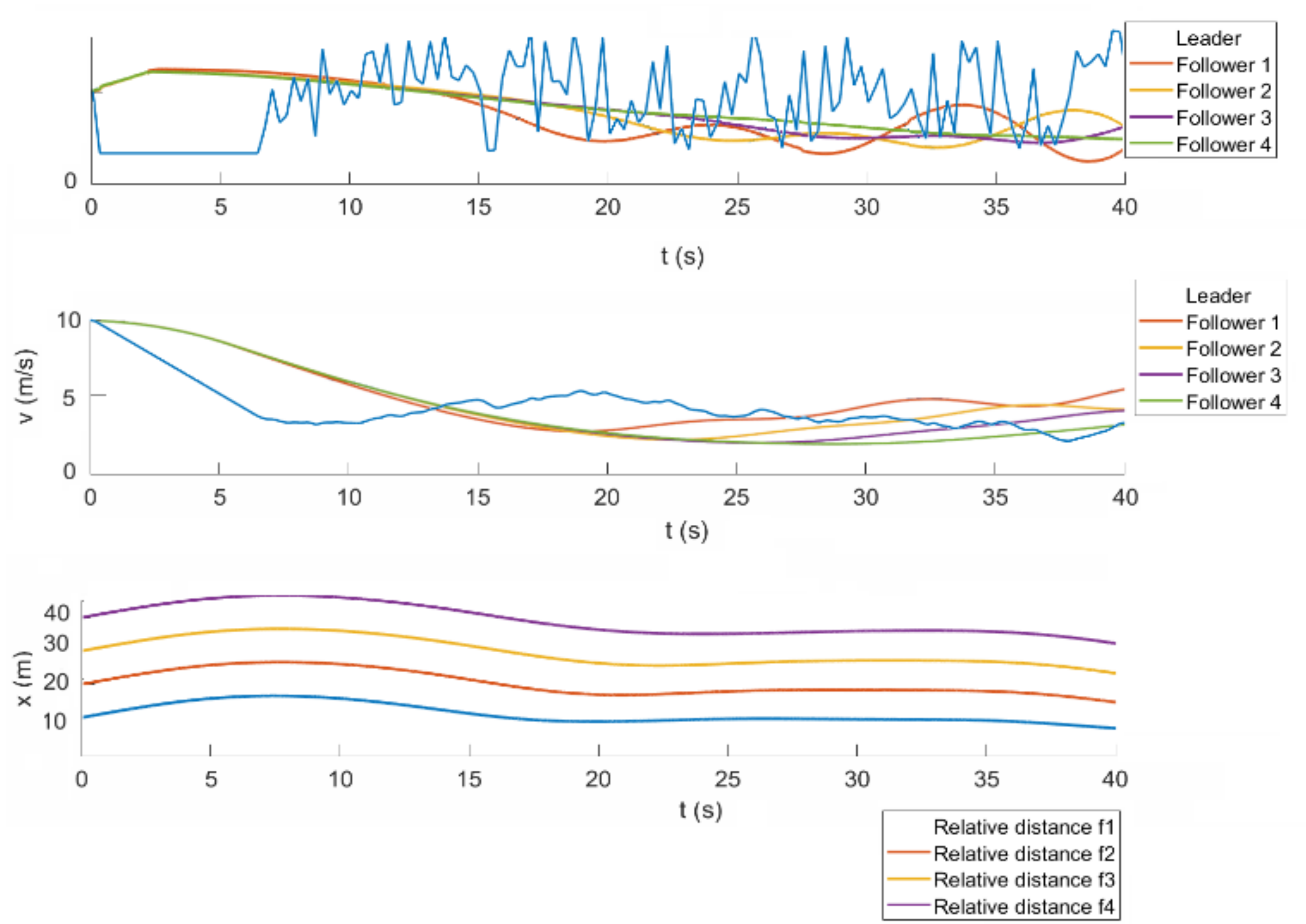
- Evaluate the protocol by changing its parameters and simulate with the speed profiles found earlier.



Strategy

- Find scenarios that maximise the data age
- Simulate the scenarios using different DCC parameters
  a. Too few or too much messages may result in collision
- Evaluate the results and fine-tune the protocol

Example of generated input



DCC Parameters Evaluation

	T_min	T_max	Distance	Speed	Average Data Age	Max Data Age	Min Distance	Max Distance	Collision
Increasead Frequency 1	0.08	0.6	3	0.3	0.38	0.48	4	8	No
Increasead Frequency 2	0.05	0.3	2	0.1	0.67	1.57	0	6	Yes
Default	0.1	1	4	0.5	0.42	0.79	2	11	No
Decreasead Frequency 1	0.3	1.3	5	0.7	0.41	0.81	6	11	No
Decreasead Frequency 2	0.5	1.5	6	0.9	0.49	1.19	7	13	No

	T_min	T_max	Distance	Speed	Average Data Age	Max Data Age	Min Distance	Max Distance	Collision
Increasead Frequency 1	0.08	0.6	3	0.3	0.47	0.9	6	10	No
Increasead Frequency 2	0.05	0.3	2	0.1	0.73	0.98	2	7	No
Default	0.1	1	4	0.5	0.46	1.07	4	12	No
Decreasead Frequency 1	0.3	1.3	5	0.7	0.39	0.85	6	10	No
Decreasead Frequency 2	0.5	1.5	6	0.9	0.52	0.92	3	11	No

	T_min	T_max	Distance	Speed	Average Data Age	Max Data Age	Min Distance	Max Distance	Collision
Increasead Frequency 1	0.08	0.6	3	0.3	0.71	1.12	3	7	No
Increasead Frequency 2	0.05	0.3	2	0.1	0.64	1.36	6	12	No
Default	0.1	1	4	0.5	0.43	1.08	4	9	No
Decreasead Frequency 1	0.3	1.3	5	0.7	0.41	0.95	4	8	No
Decreasead Frequency 2	0.5	1.5	6	0.9	0.89	1.62	0	9	Yes

Ongoing work: Catching the Cheaters

Detect software doping on Real Driving Emission test procedures by learning driving behaviour and using model checking techniques.

Context

Real Driving Emission test

- Verify xapollutant and particle emissions
- Uses a wide range of operating conditions on the road
  - Speed
  - Temperature
  - Altitude
  - Distance

Software Doping

A program is clean if for every standard parameter, whenever it is supplied with any input that deviates within "reasonable distance" from a given standard input, it exhibits a visible output which does not deviate beyond a "reasonable distance" xfrom the specified output corresponding to such standard input.

(Barthe et al., 2018) in Facets of Software Doping

Strategy

- Infer a hybrid model using **passive learning** algorithms
  - Helps with system comprehension, simulation and (off-line) testing
- Apply MBT to detect suspicious behaviour
  - Testing with real cars requires expensive setup
- Execute selected tests in a real driving setting
  - Replications cannot deviate beyond a “reasonable distance”

Done

- Test case **generation** algorithm for testing cyber-physical systems
- Investigated **soundness** bounds for conformance testing
- **Process** to apply the adjustments in the right order
- **Tool prototype** to implement the process:
  - soundness bound calculation,
  - test case execution, and
  - conformance analysis.

To Be Done

- Generalizing the **prototype**
  (open source tool, collaboration is very welcome)

- **Test input** generation:
  using **learning** techniques

- Extending the **case study**

https://github.com/hlsa/cps-conf-tool

Thank You Very Much!

mm789@le.ac.uk