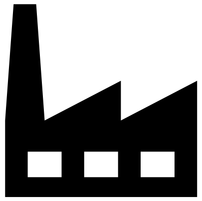
Multi-Paradigm Modeling for Design and Operation of Intelligent Cyber-Physical Systems

Keynote Talk, First International Workshop on Multi-Paradigm Modeling of Cyber-Physical Systems (MPM4CPS)
Munich, Germany. September 10, 2019

Akshay Rajhans, PhD
arajhans@mathworks.com
https://arajhans.github.io

About me

- ‘CPS’ Practitioner before it was called CPS
  - Embedded controls for diesel engine applications
  - Programmable logic controller for industrial automation



- CPS Research at the intersection of
  - Model-based design and analysis
  - Formal methods
  - Software and system architecture

- CPS Research Scientist at MathWorks

Perspective shaped by my personal career trajectory

Academic Researcher

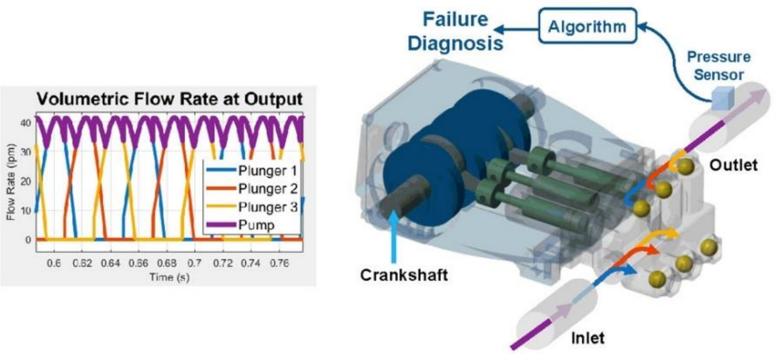
Tool Developer

Industry Practitioner

Outline

- Introduction
- Theoretical aspects of multi-paradigm model-based design for CPS
  - Architecture modeling and structural analysis
  - Semantic analysis and heterogeneous verification
  - Compositional analysis
- Practical aspects of a multi-domain simulation platform
  - Graphical modeling of hybrid dynamics using Simulink and Stateflow
- Recap and conclusions

Cyber-physical systems have societal scale applications



Smart Manufacturing[3]

Smart Infrastructure[4]

Smart Energy[2]

Smart Transportation[1]

Smart Health[5]

Traffic accidents are bad

Clock Facts

Fatalities per Day	
2017	102
2016	103
2015	97

Source: FARS

Pedestrian Fatalities per Day	
2017	16
2016	17
2015	15

Source: FARS

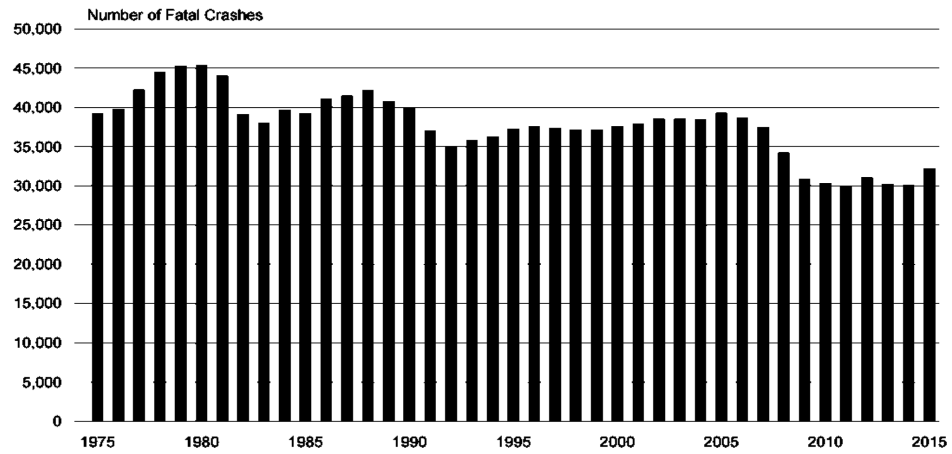
People Injured per Day	
2017	7,523
2016	8,363
2015	*6,693*

Source: GES/CRSS†

Pedestrians Injured per Day	
2017	195
2016	238
2015	*192*

Source: GES/CRSS†

Fatal Crashes, 1975-2015



Leading Cause of Death

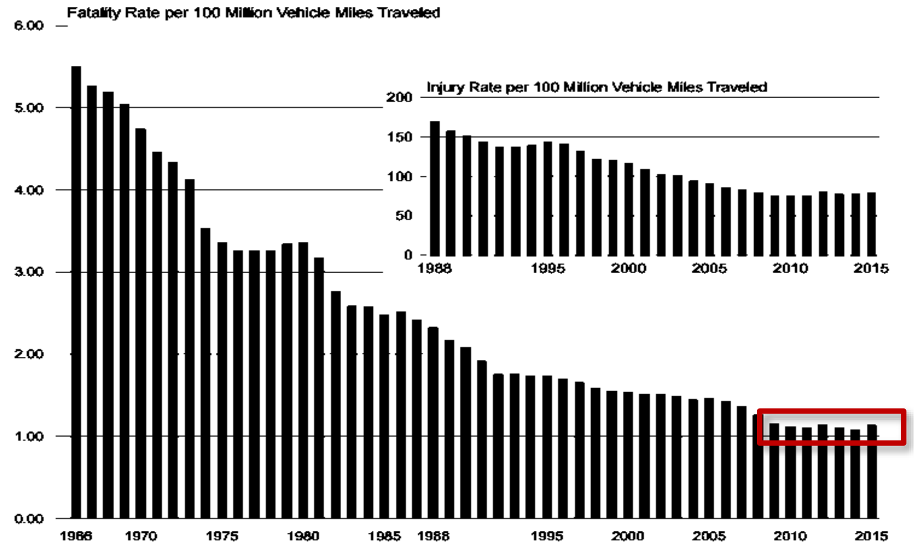
Motor vehicle crashes were the leading cause of death for age 10, 11 and 17 through 22 in 2016.

Source: Centers for Disease Control and Prevention, (2016) Leading Cause of Death, WISQARS

Economic and Comprehensive Costs to Society by Type of Crash
2010 Costs (in Billions)

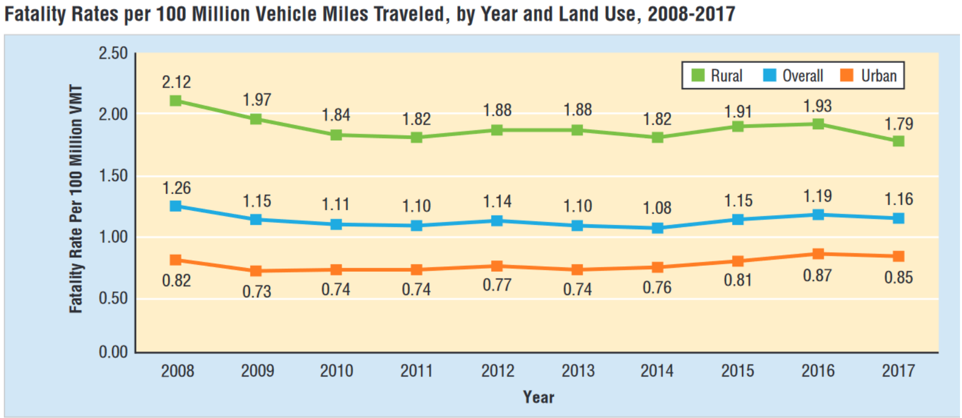
Crash Type	Economic Cost	Comprehensive Cost*
All	$242	$836

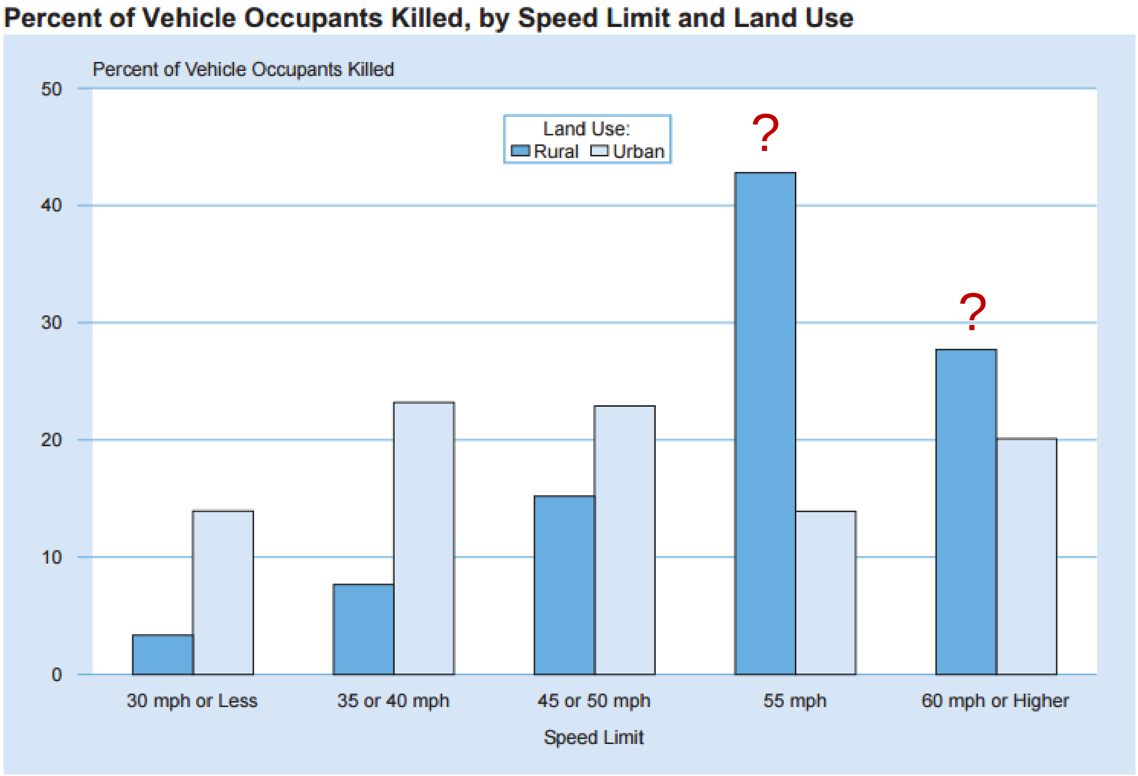
Motor Vehicle Fatality and Injury Rates per 100 Million Vehicle Miles Traveled, 1966-2015



Quick Facts 2017, NHTSA, https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812747

Traffic Safety Facts 2015, NHTSA, https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812384



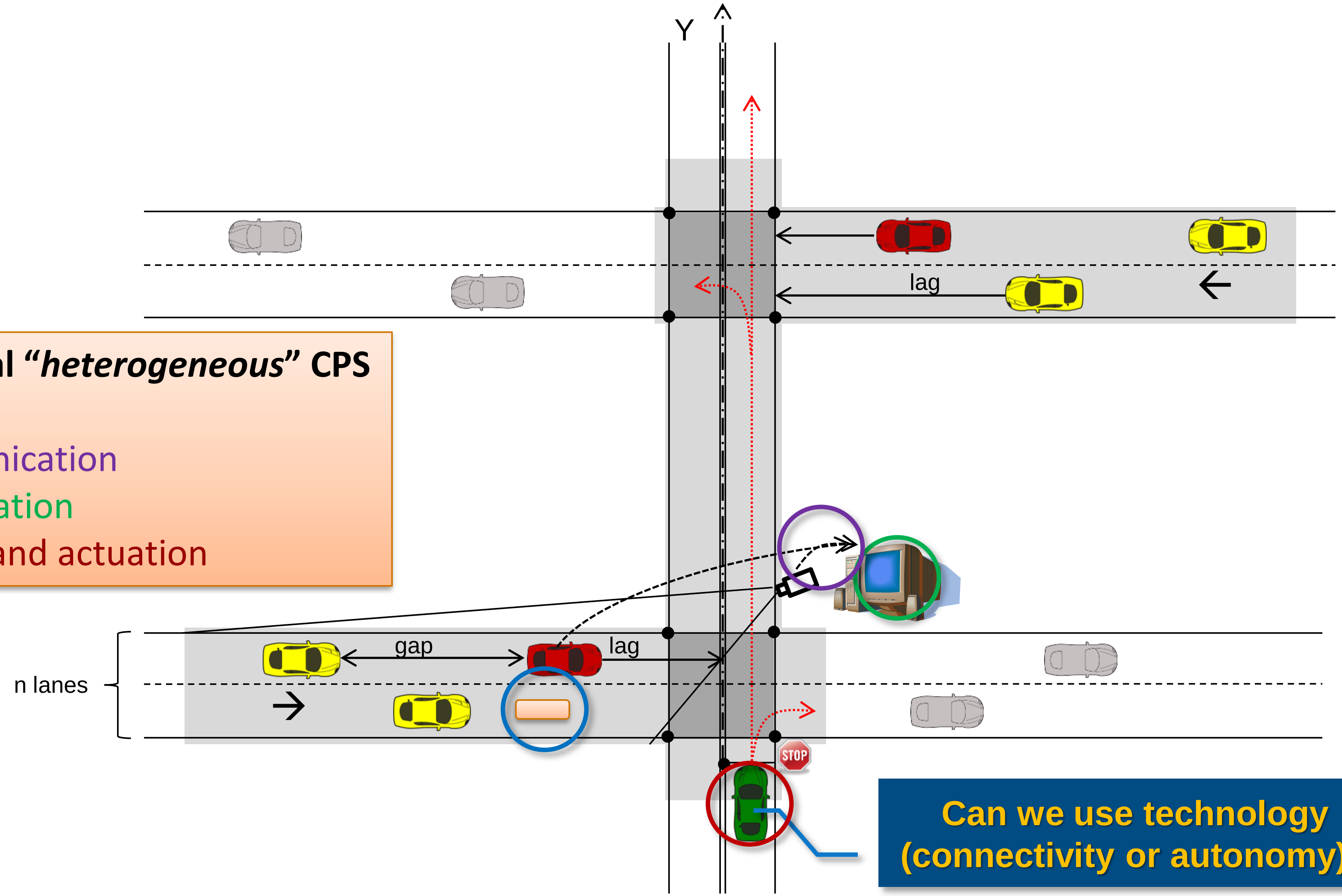


- According to the 2017 American Community Survey from the Census Bureau, an estimated 19 percent of the U.S. population lived in rural areas, and according to FHWA only 30 percent of the total vehicle miles traveled (VMT) in 2017 were in rural areas. However, rural areas accounted for 46 percent of all traffic fatalities in 2017.

Rural/Urban Comparison of Traffic Fatalities, NHTSA https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812741

Traffic Safety Facts 2015, NHTSA, https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812384

Intersection collision avoidance system

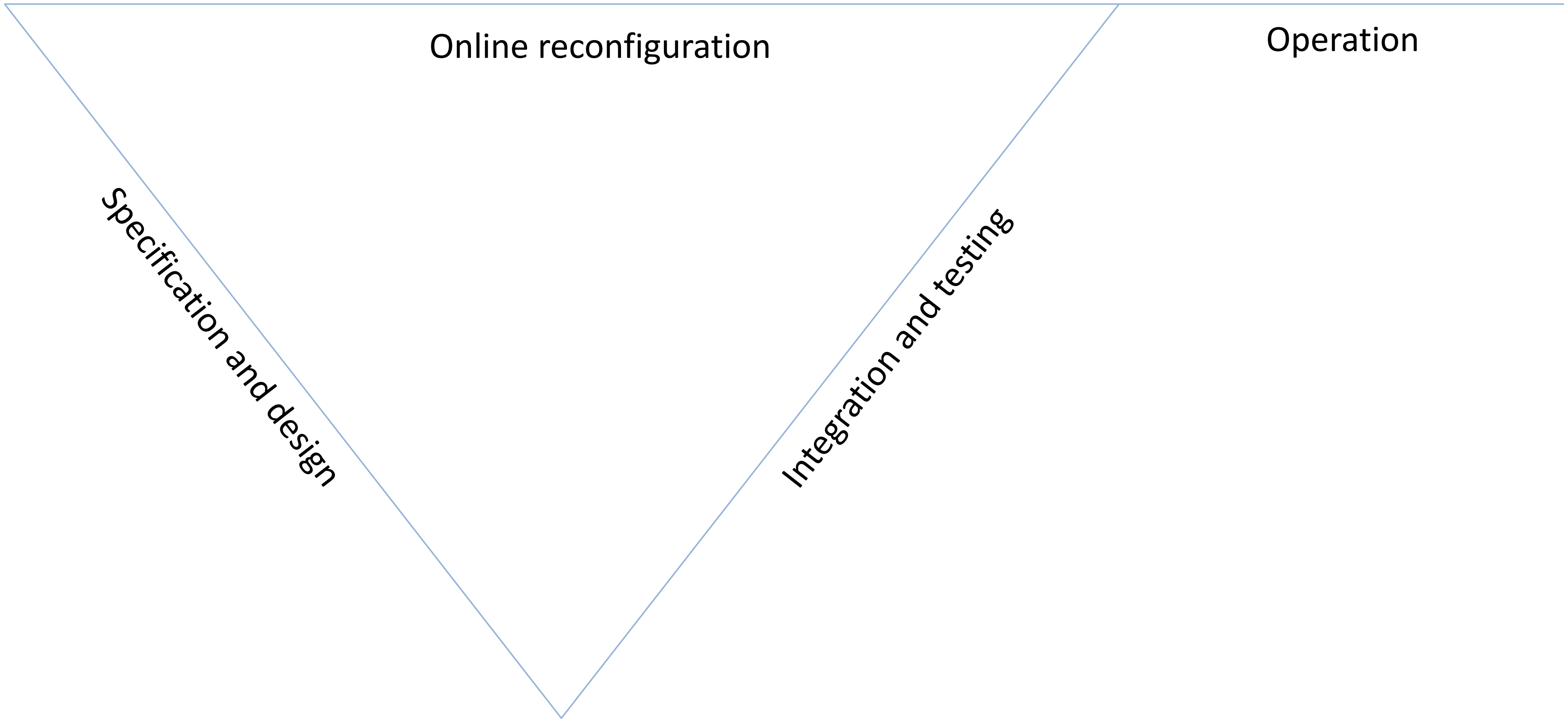


Prototypical "*heterogeneous*" CPS

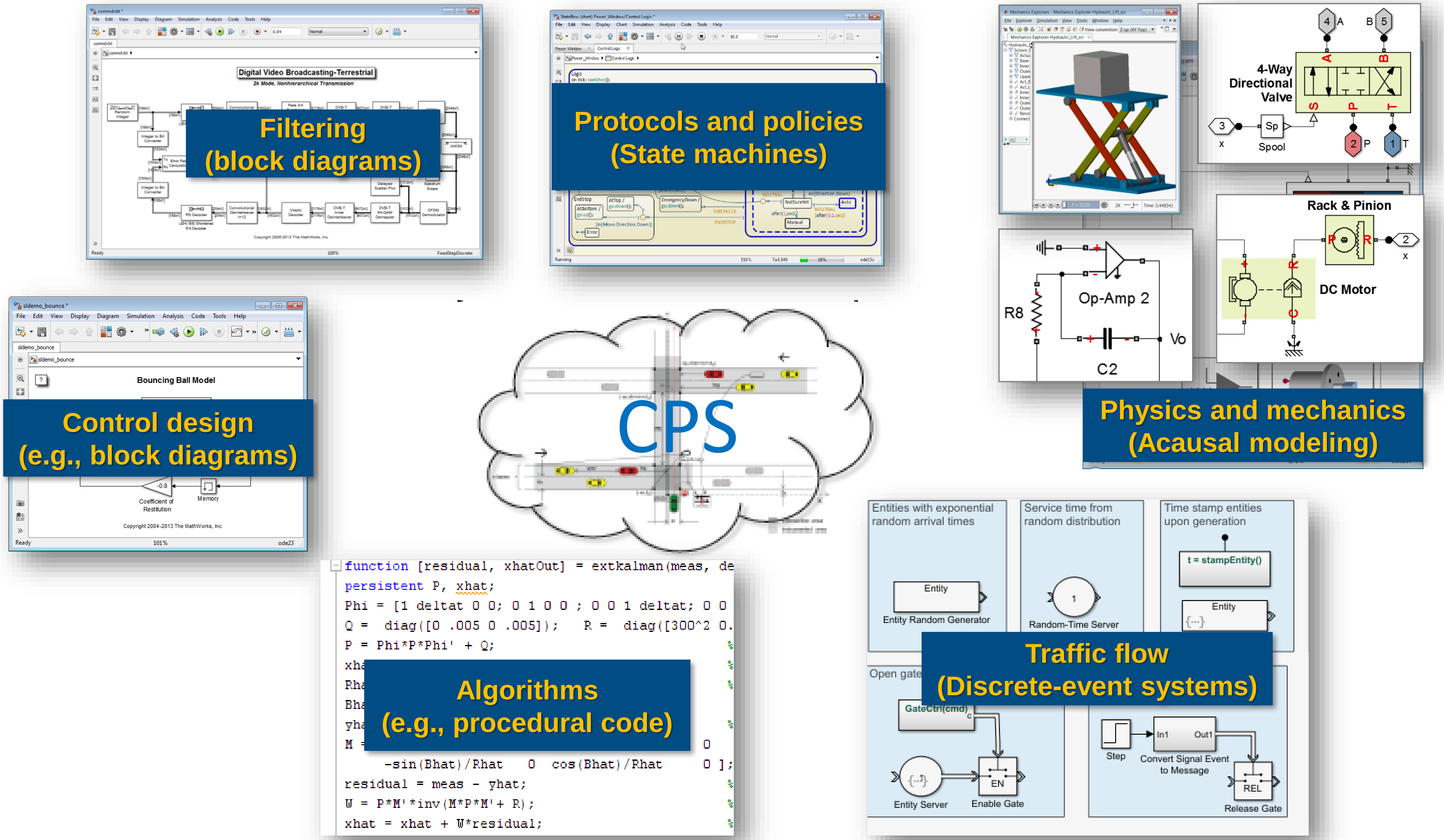
- Sensing
- Communication
- Computation
- Physics and actuation

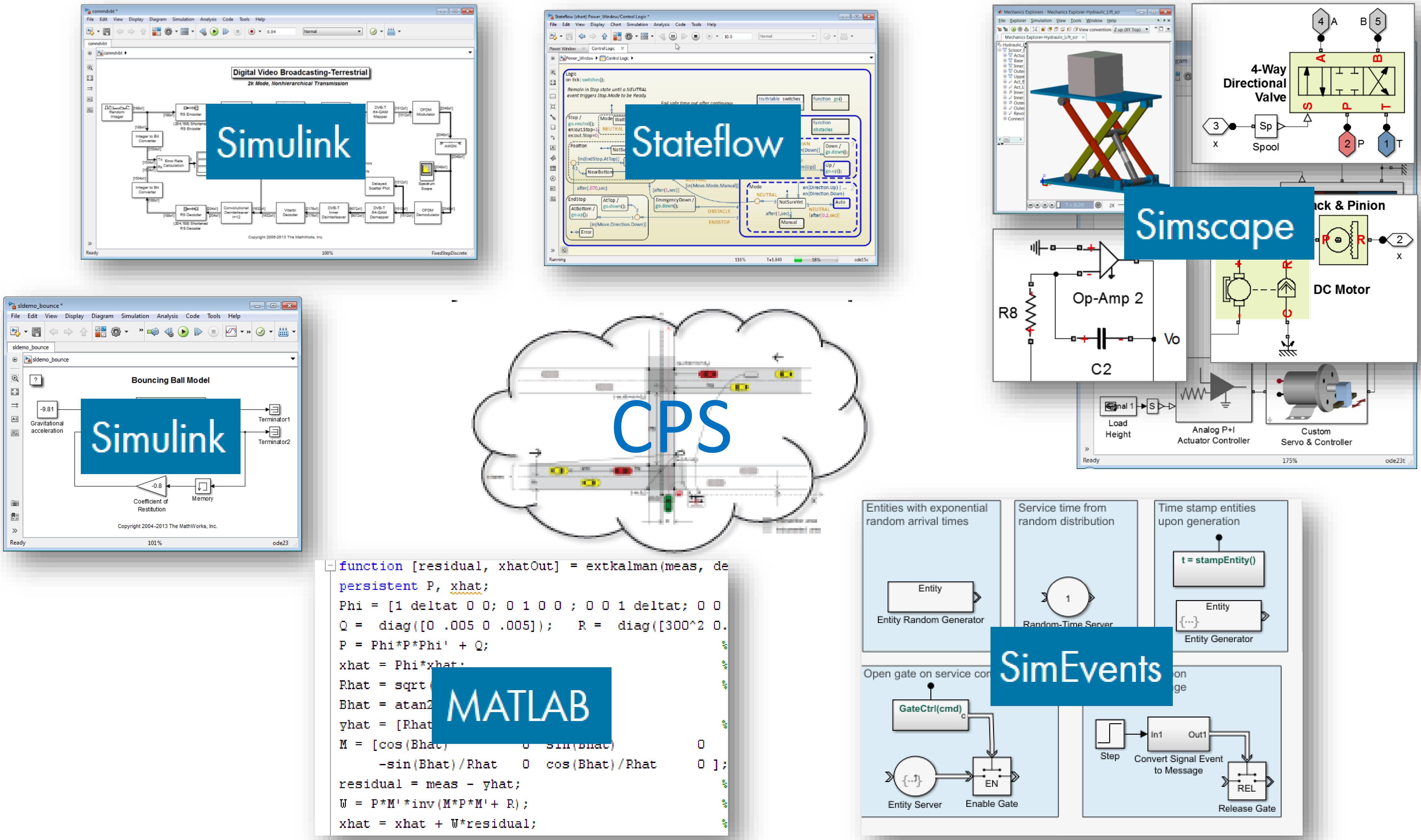
Can we use technology (connectivity or autonomy)?

Models are useful in both design and operation



Challenges in the Operation and Design of Intelligent Cyber-Physical Systems, S. Castro, P.J. Mosterman, A.H. Rajhans, and R.G. Valenti, book chapter, *Complexity Challenges in Cyber Physical Systems: Using Modeling and Simulation (M&S) to Support Intelligence, Adaptation and Autonomy*, S. Mittal and A. Tolk, eds., Wiley, 2019.


Filtering
(block diagrams)
Digital Video Broadcasting-Terrestrial
Protocols and policies
(State machines)
4-Way Directional Valve
Rack & Pinion
DC Motor
Op-Amp 2
R8
C2
Vo
Physics and mechanics
(Acausal modeling)
Bouncing Ball Model
Control design
(e.g., block diagrams)
CPS
function [residual, xhatOut] = extkalman(meas, de
persistent P, xhat;
Phi = [1 deltat 0 0; 0 1 0 0 ; 0 0 1 deltat; 0 0
Q = diag([0 .005 0 .005]); R = diag([300^2 0.
P = Phi*P*Phi' + Q;
-sin(Bhat)/Rhat 0 cos(Bhat)/Rhat 0];
residual = meas - yhat;
W = P*M'*inv(M*P*M'+ R);
xhat = xhat + W*residual;
Algorithms
(e.g., procedural code)
Entities with exponential random arrival times
Entity Random Generator
Service time from random distribution
Random-Time Server
Time stamp entities upon generation
t = stampEntity()
Open gate
GateCtrl(cmd)
Entity Server
Enable Gate
Step
Convert Signal Event to Message
Release Gate
Traffic flow
(Discrete-event systems)


Simulink
Digital Video Broadcasting-Terrestrial
2k Mode, Nonhierarchical Transmission
Stateflow
Simscape
4-Way Directional Valve
Spool
Rack & Pinion
DC Motor
Op-Amp 2
R8
C2
Vo
Load Height
Analog P+I Actuator Controller
Custom Servo & Controller
Simulink
Bouncing Ball Model
Gravitational acceleration
Coefficient of Restitution
Memory
CPS
function [residual, xhatOut] = extkalman(meas, de
persistent P, xhat;
Phi = [1 deltat 0 0; 0 1 0 0 ; 0 0 1 deltat; 0 0
Q = diag([0 .005 0 .005]); R = diag([300^2 0.
P = Phi*P*Phi' + Q;
xhat = Phi*xhat;
residual = meas - yhat;
W = P*M'*inv(M*P*M'+ R);
xhat = xhat + W*residual;
MATLAB
Entities with exponential random arrival times
Entity Random Generator
Service time from random distribution
Random-Time Server
Time stamp entities upon generation
t = stampEntity()
Entity Generator
Open gate on service com
GateCtrl(cmd)
Entity Server
Enable Gate
SimEvents
Step
Convert Signal Event to Message
Release Gate

- No ‘universal’ modeling formalism, modeling everything together intractable
- Different formalisms suited for different *aspects* of system design
- Each model represents *some* design aspect well, but not others
- Models make *interdependent assumptions* about each other and the system
- *Analysis tools* often specialized for a particular formalism

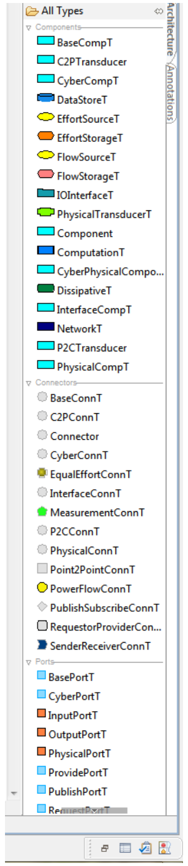
- ***Given all of these, how do we design a correct system?***

Outline

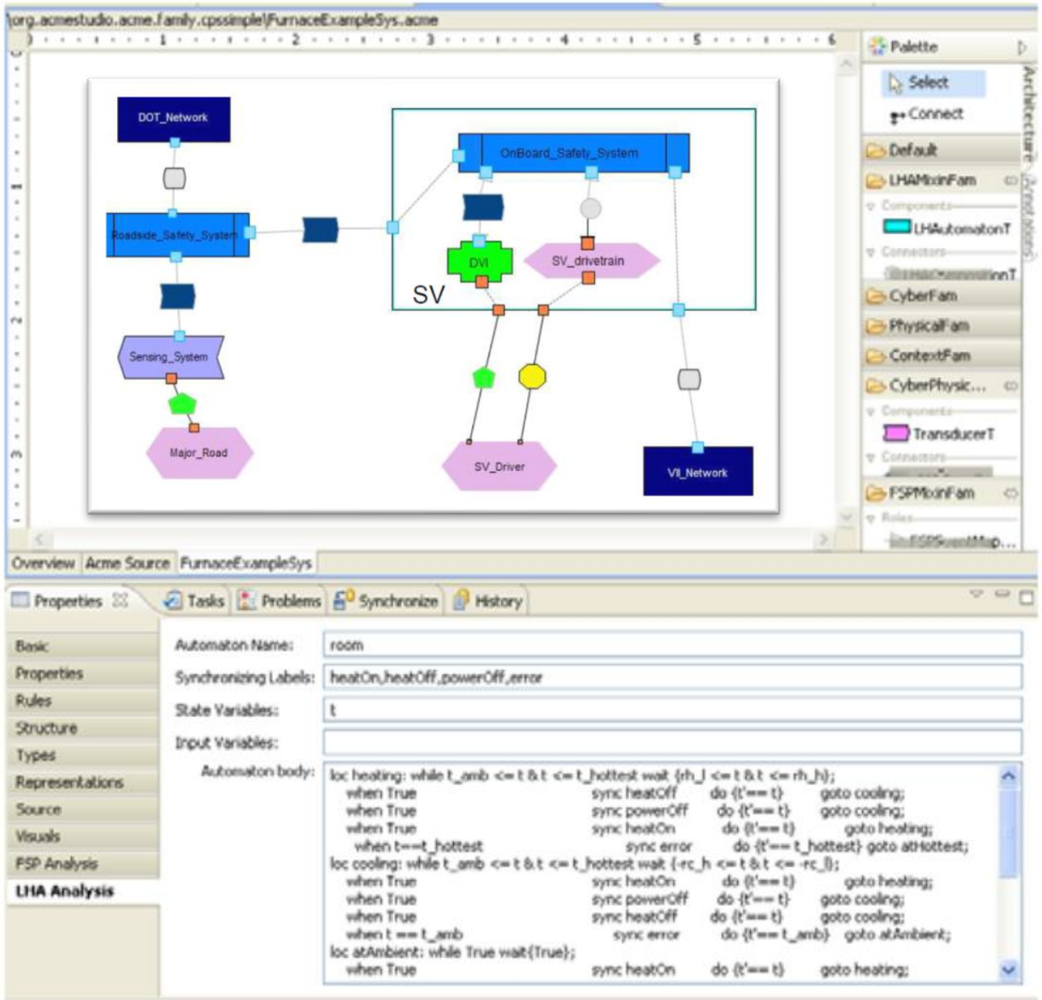
- Introduction
- Theoretical aspects of multi-paradigm model-based design for CPS
  - Architecture modeling and structural analysis
  - Semantic analysis and heterogeneous verification
  - Compositional analysis
- Practical aspects of a multi-domain simulation platform
  - Graphical modeling of hybrid dynamics using Simulink and Stateflow
- Recap and conclusions

From software architecture to CPS architecture

- **Even though there is no system-level model, there is a system architecture**
- **Extend software architecture vocabulary with physical elements**
- **Heterogeneous component models are annotations on the architecture elements**

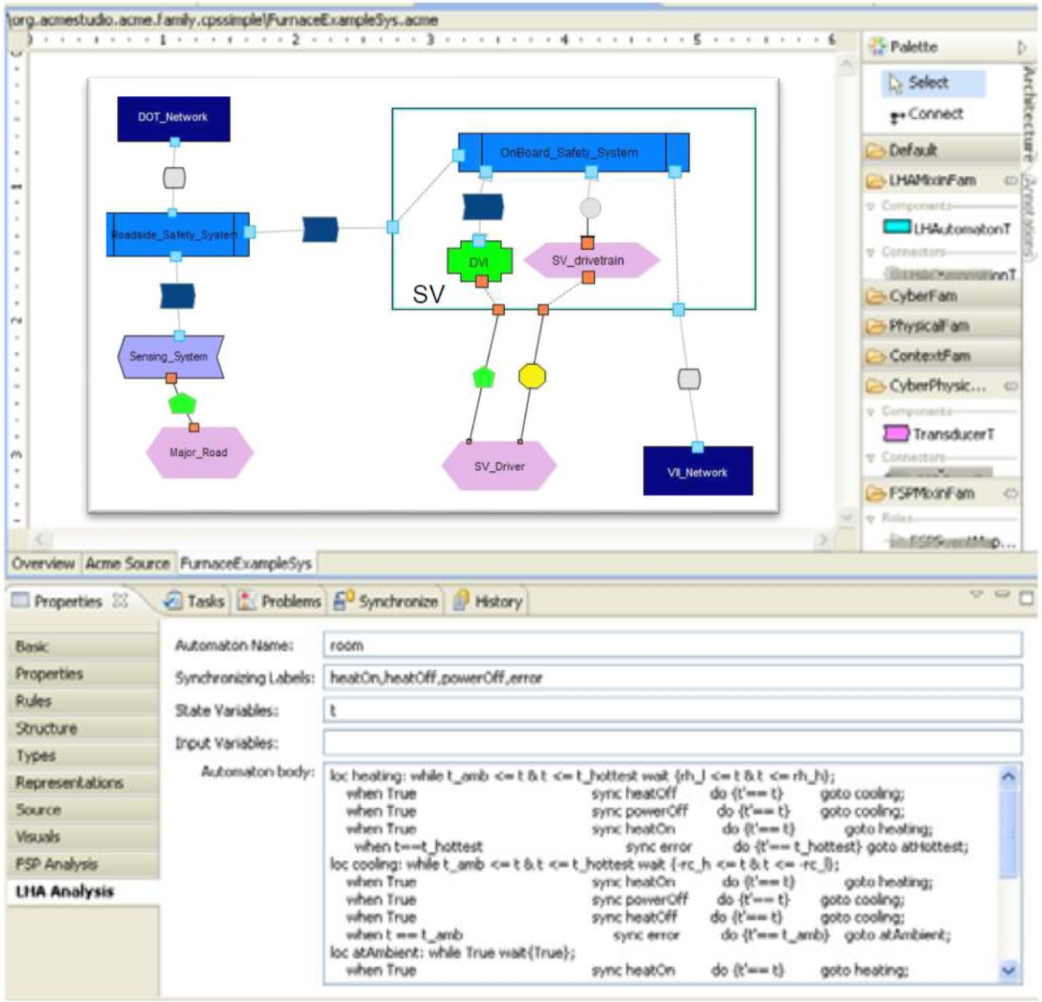


CPS architectural style palette in [AcmeStudio]

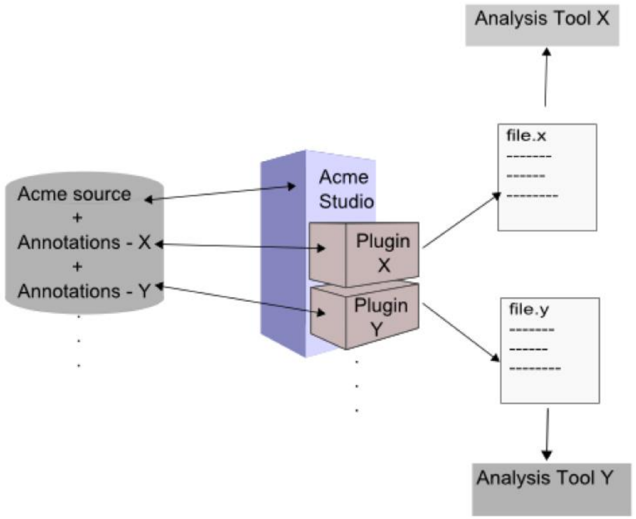


[RCS+09] A. Rajhans et al., "*An Architectural Approach to the Design and Analysis of Cyber-Physical Systems*," Third International Workshop on Multi-Paradigm Modeling (MPM), 2009

From software architecture to CPS architecture



Heterogeneous component models are annotations on the architecture elements



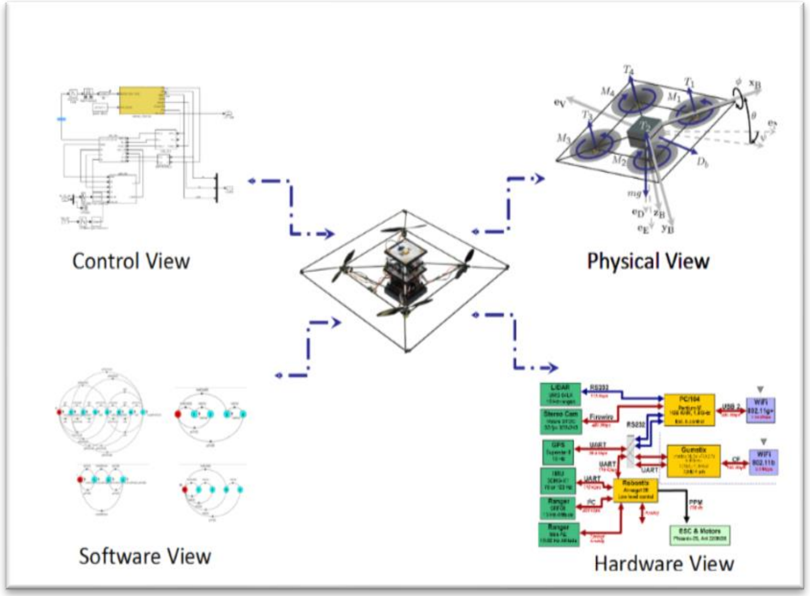
Implicit assumption: models composed of the same structure as the architecture

[RCS+09] A. Rajhans et al., "*An Architectural Approach to the Design and Analysis of Cyber-Physical Systems*," Third International Workshop on Multi-Paradigm Modeling (MPM), 2009

Base architecture and architecture views

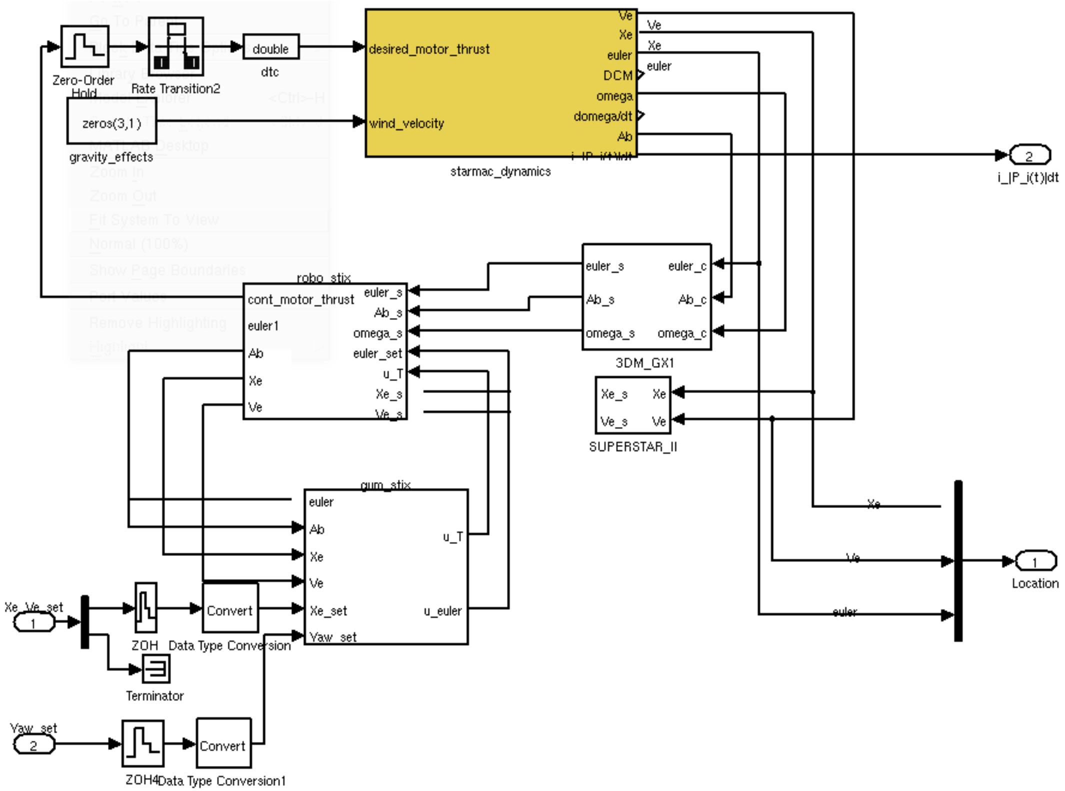
- **Models have their own structure. What gets abstracted away depends on the paradigm.**
- **Architectures extracted from model structure are 'views' of the base architecture.**
- **There are relations between the views and the base architecture.**

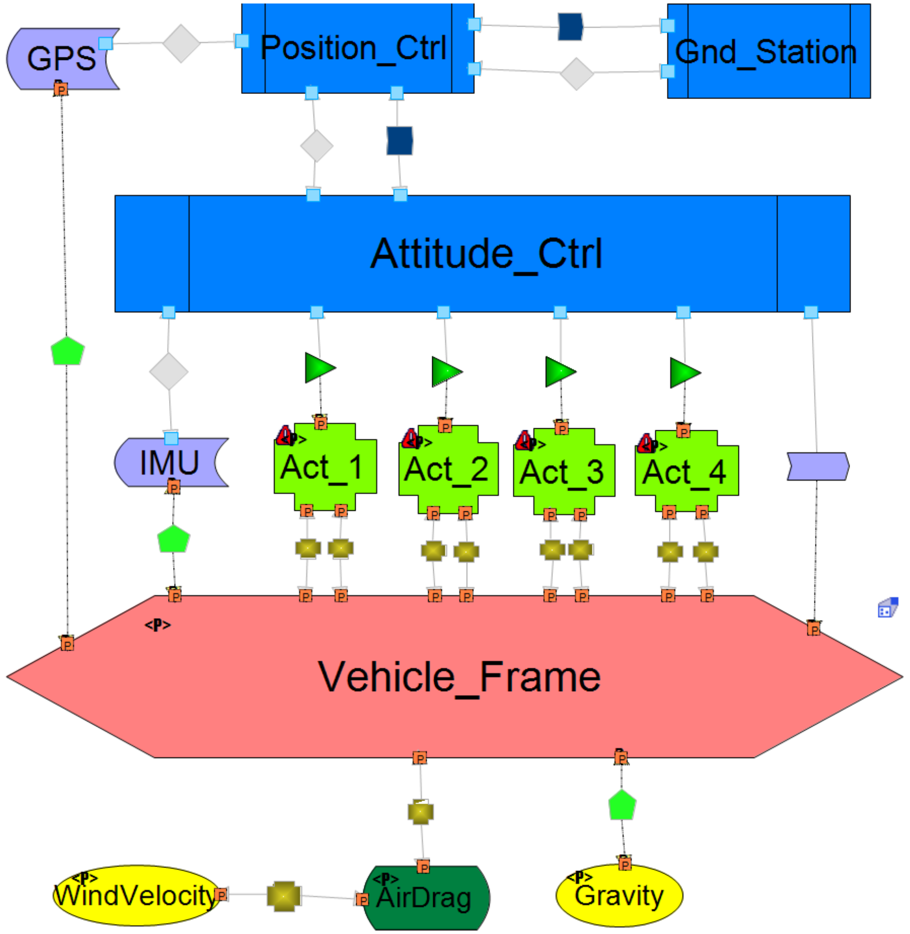
STARMAC
Quadrotor



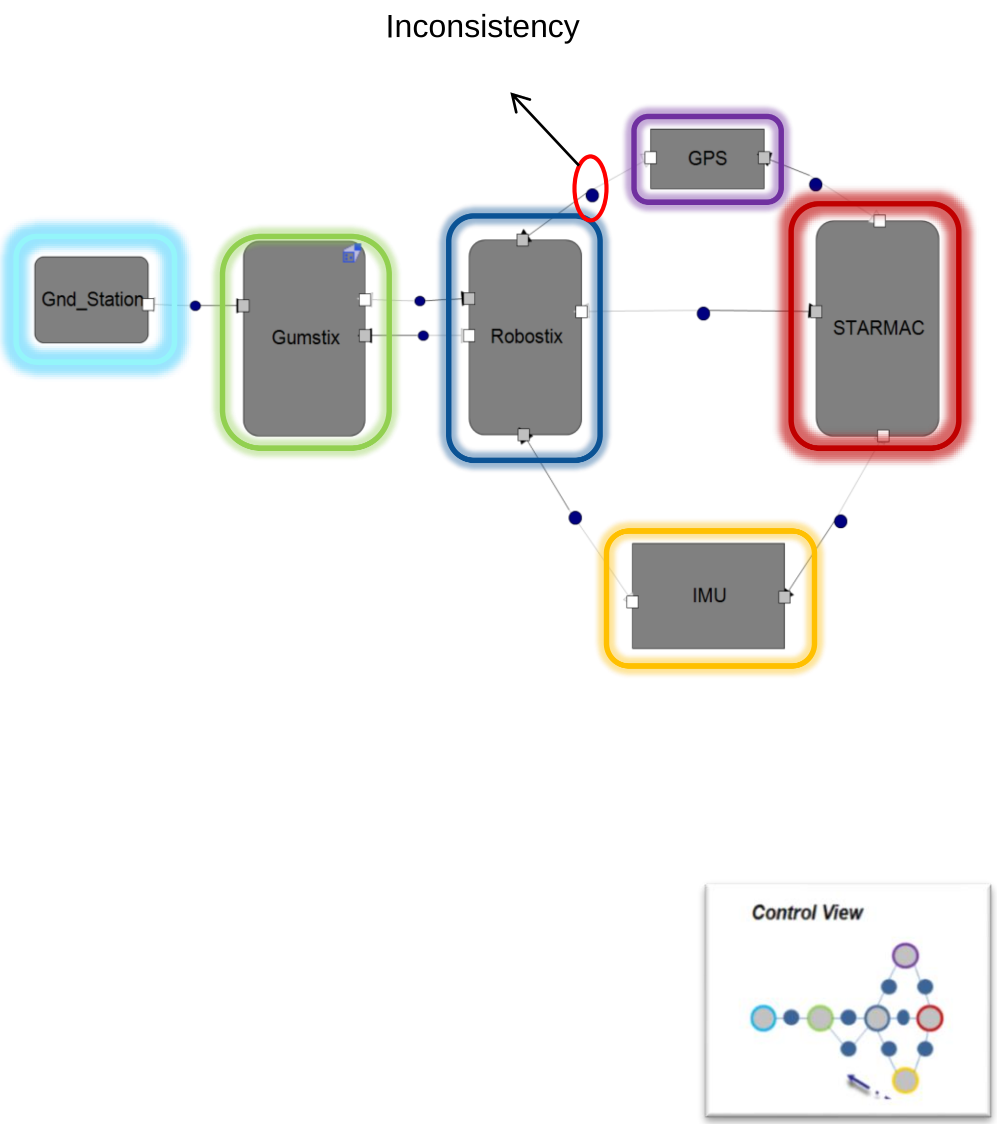
[BDK+10b] A. Bhave, D. Garlan, B. Krogh, A. Rajhans, and B. Schmerl, "*Augmenting Software Architectures with Physical Components*," Embedded Real Time Software and Systems (ERTS^2), 2010

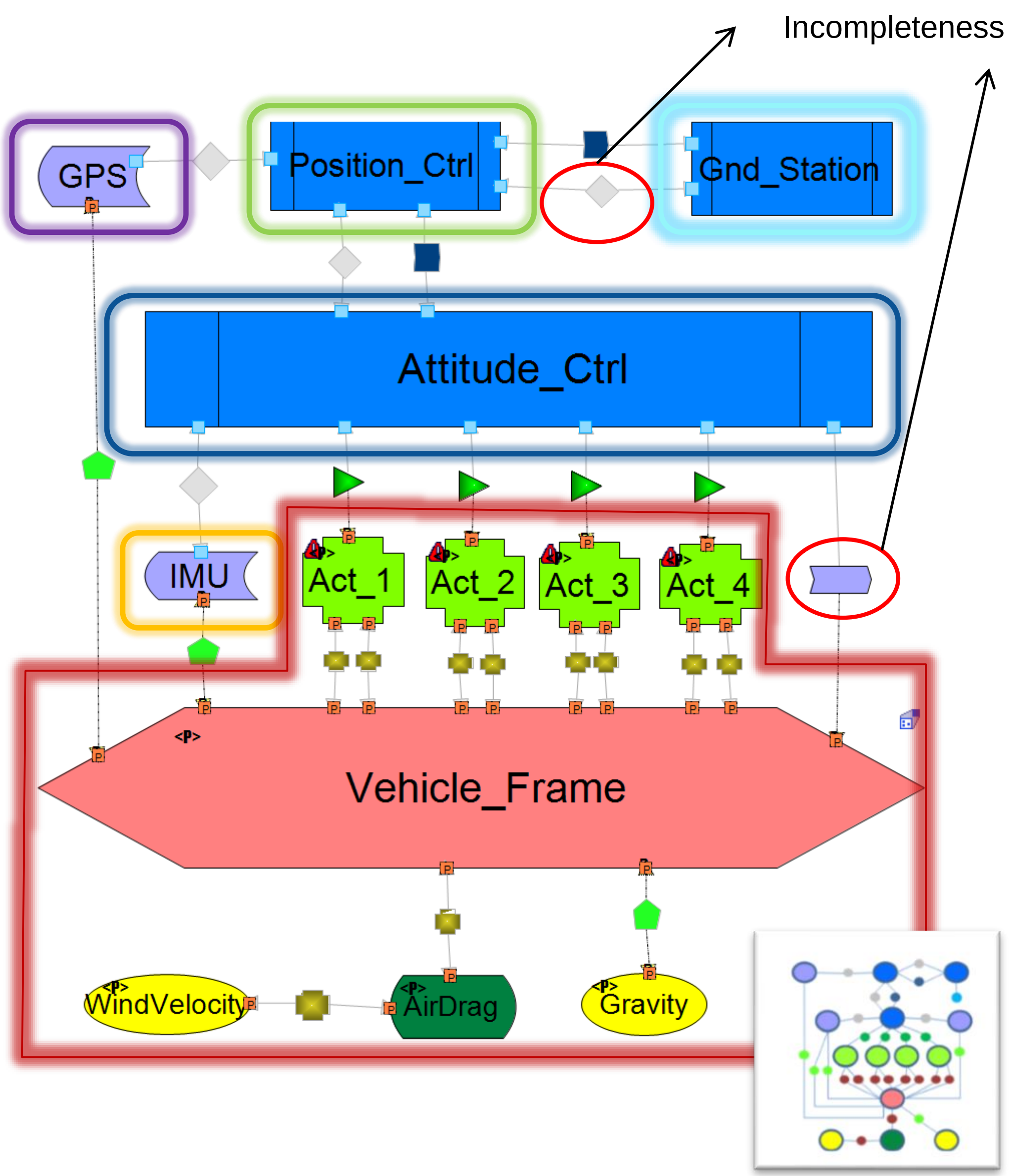
Simulink architecture view





Simulink architecture view

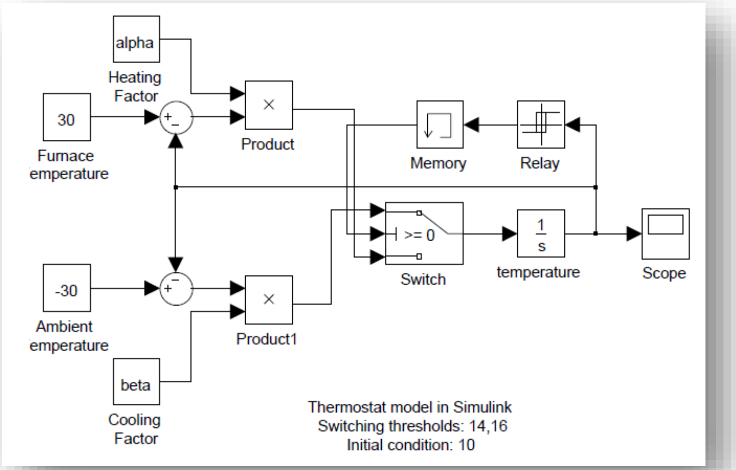




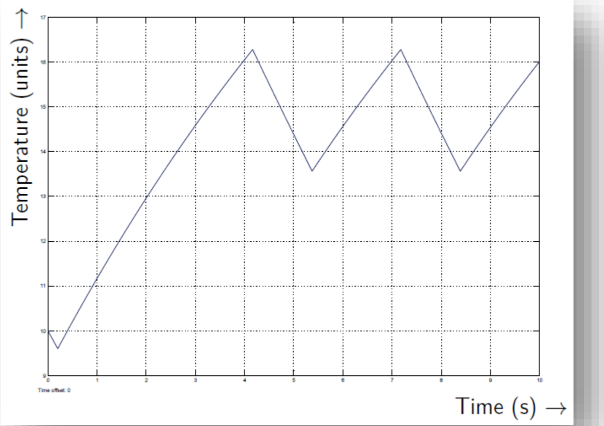
Outline

- Introduction
- Theoretical aspects of multi-paradigm model-based design for CPS
  - Architecture modeling and structural analysis
  - ➢ Semantic analysis and heterogeneous verification
  - Compositional analysis
- Practical aspects of a multi-domain simulation platform
  - Graphical modeling of hybrid dynamics using Simulink and Stateflow
- Recap and conclusions

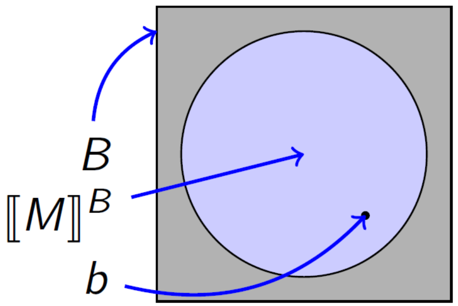
Semantic interpretations of models and specifications



Model M



A behavior b that M exhibits



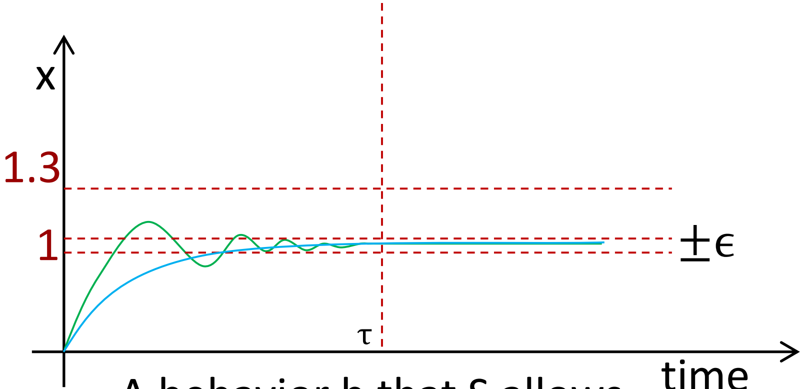
$[\![M]\!]^B$: "semantic interpretation" of M in a behavior domain B

(A set of all behaviors that M exhibits in B)

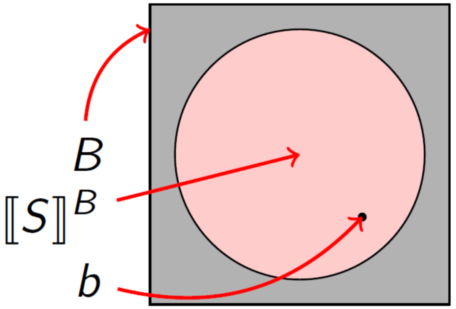
1) "overshoot is no more than 1.3 units and settling time is less than τ"

2) $\square(x < 1.3) \wedge \diamond_{\tau}(x \in [1 \pm \epsilon])$

Specification S



A behavior b that S allows



$[\![S]\!]^B$: "semantic interpretation" of S in B

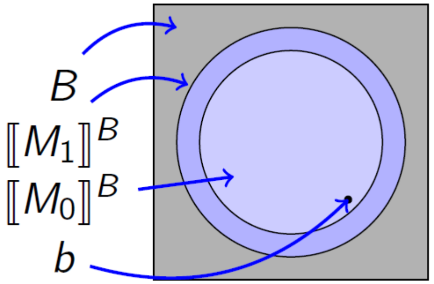
(A set of all behaviors that S allows in B)

[RBL+11] A. Rajhans, A. Bhave, S. Loos, B. H. Krogh, A. Platzer, and D. Garlan, "*Using Parameters in Architectural Views to Support Heterogeneous Design and Verification*," 50th IEEE Conference on Decision and Control, 2011

Abstraction, implication, and satisfaction as behavior set inclusions

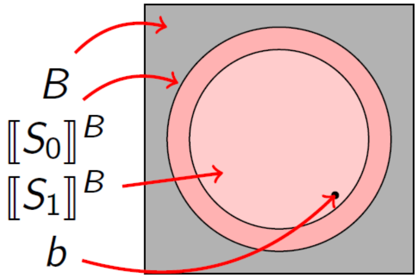
- Model M_1 abstracts M_0 in B, written $M_0 \sqsubseteq^B M_1$
  if
  $$[\![M_0]\!]^B \subseteq [\![M_1]\!]^B$$

can be heterogeneous



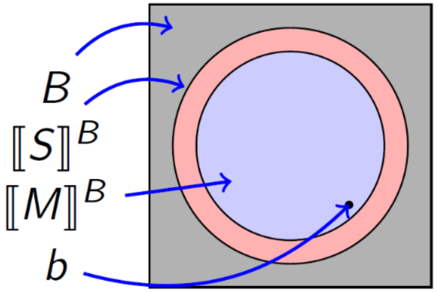
- Specification S_1 implies S_0 in B, written $S_1 \Rightarrow^B S_0$
  if
  $$[\![S_1]\!]^B \subseteq [\![S_0]\!]^B$$

can be heterogeneous



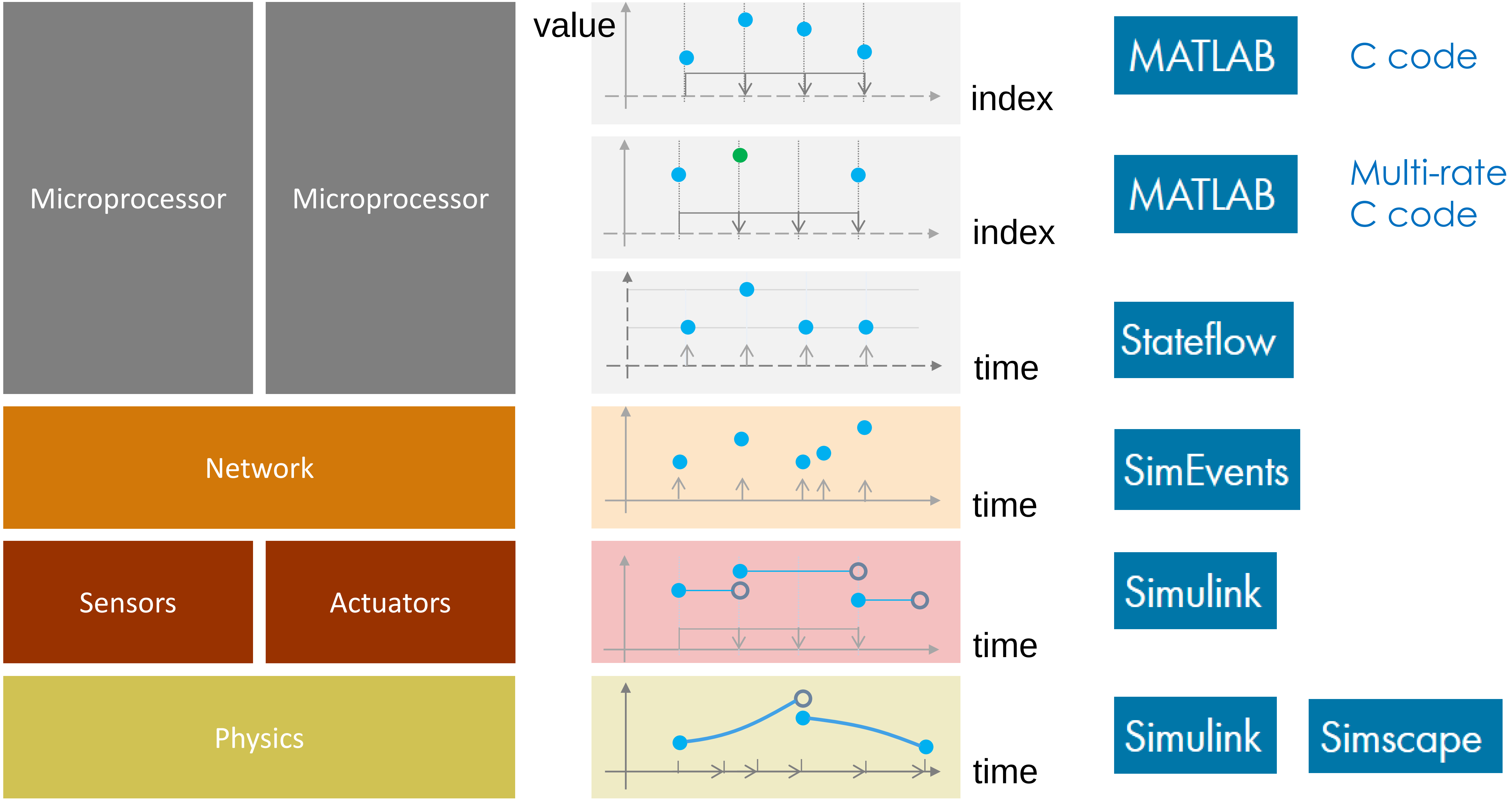
- Model M satisfies specification S in B, written $M \models^B S$
  if $[\![M]\!]^B \subseteq [\![S]\!]^B$

often heterogeneous



Homogeneous in B: Same B everywhere

Simulation of Hybrid Dynamic Systems, P.J. Mosterman, A. Rajhans, A. Mavrommati, R.G. Valenti, Springer Encyclopedia of Systems and Control, Second Edition, submitted.

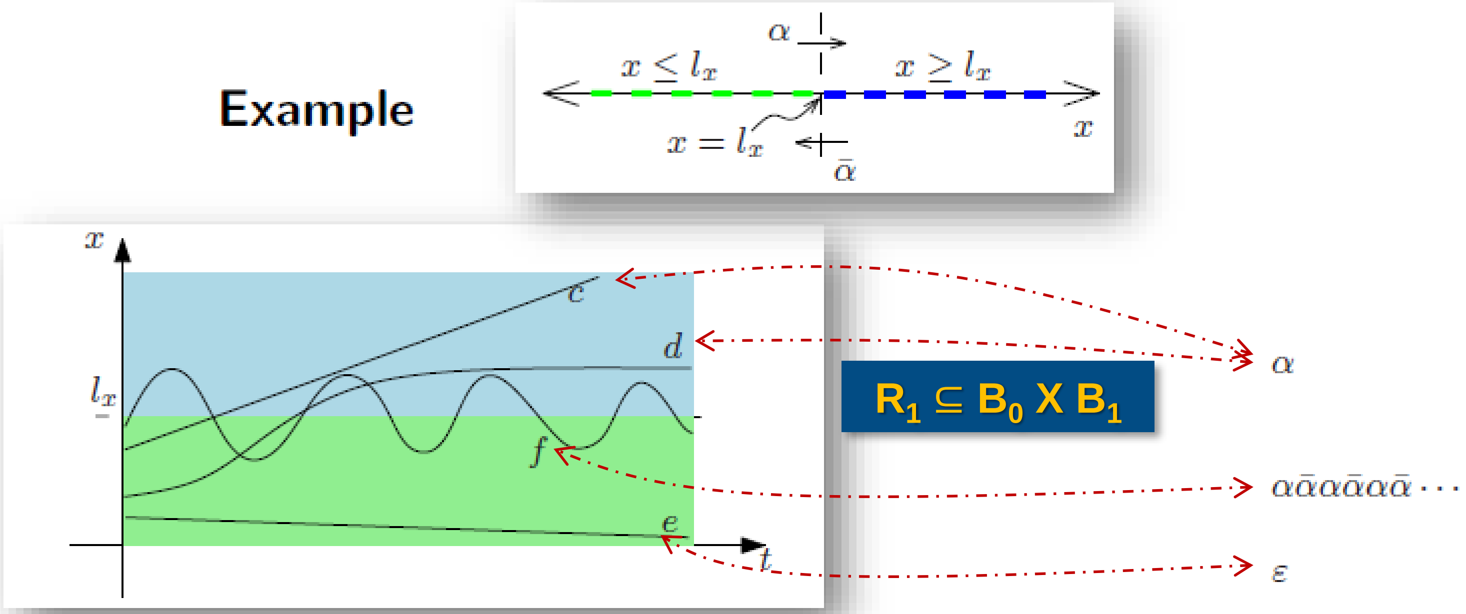


Simulation of Hybrid Dynamic Systems, P.J. Mosterman, A. Rajhans, A. Mavrommati, R.G. Valenti, Springer Encyclopedia of Systems and Control, Second Edition, submitted.

Mappings between semantic domains via *behavior relations*

- *Approach:* Create “relations” between behavior domains

Example



$R_1 \subseteq B_0 \times B_1$

B_0 : 1-d continuous trajectories in x

$B_1 = \{\alpha, \bar{\alpha}\}^* \cup \{\alpha, \bar{\alpha}\}^\omega$

Given $R_1 \subseteq B_0 \times B_1$ set-based inverse map R_1^{-1} (‘α’)={c,d,…}

[RK12] A. Rajhans and B. H. Krogh, “*Heterogeneous Verification of Cyber-Physical Systems Using Behavior Relations*,” 15th ACM International Conference on Hybrid Systems: Computation and Control, 2012

Heterogeneous abstraction, implication, and satisfaction

Heterogeneous Abstraction

$M_0 \sqsubseteq^{R_1} M_1$, if

A $\qquad [\![M_0]\!]^{B_0} \subseteq R_1^{-1}([\![M_1]\!]^{B_1})$.

Heterogeneous Specification Implication

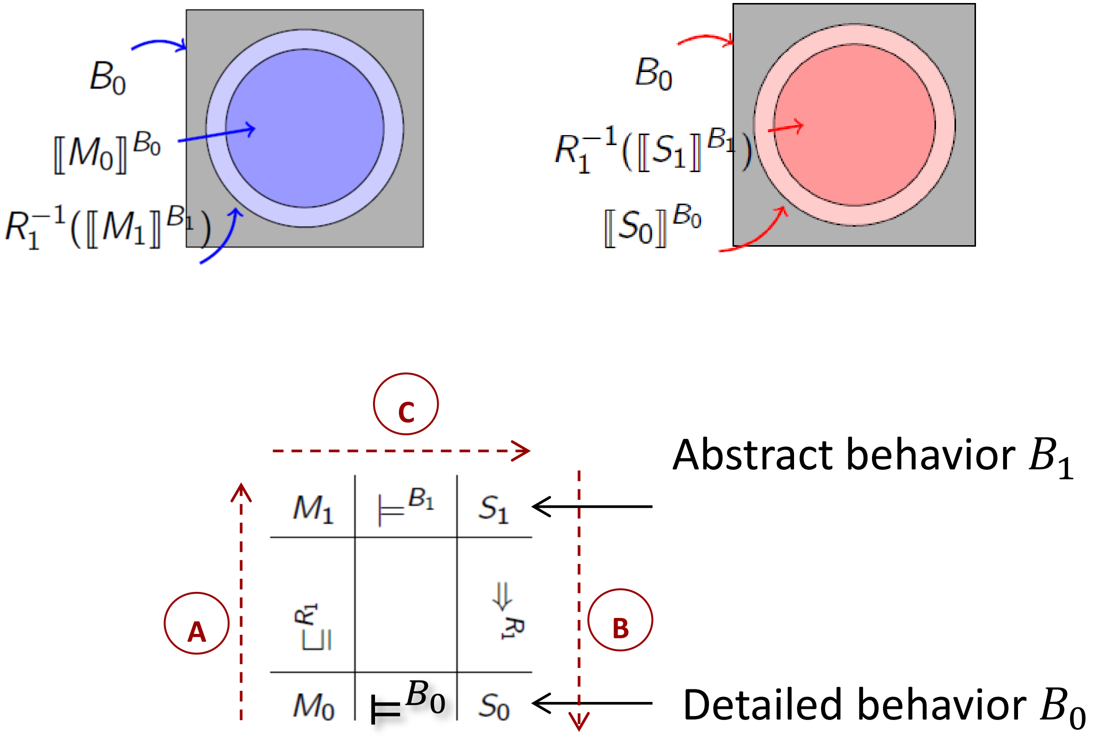
$S_1 \Rightarrow^{R_1} S_0$, if

B $\qquad R_1^{-1}([\![S_1]\!]^{B_1}) \subseteq [\![S_0]\!]^{B_0}$.

Heterogeneous Verification

If $M_0 \sqsubseteq^{R_1} M_1$, $M_1 \models^{B_1} S_1$ and $S_1 \Rightarrow^{R_1} S_0$, then $M_0 \models^{B_0} S_0$. C

(in words)



(in pictures)

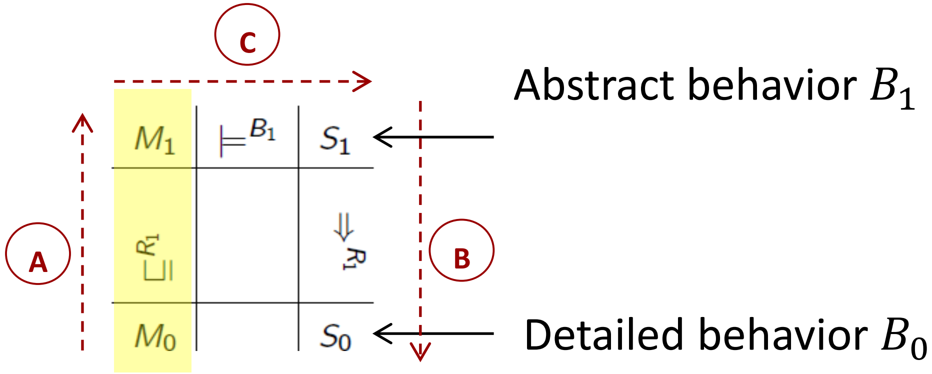
[RK12] A. Rajhans and B. H. Krogh, "*Heterogeneous Verification of Cyber-Physical Systems Using Behavior Relations*," 15th ACM International Conference on Hybrid Systems: Computation and Control, 2012

Outline

- Introduction
- Theoretical aspects of multi-paradigm model-based design for CPS
  - Architecture modeling and structural analysis
  - Semantic analysis and heterogeneous verification
  - ➢ Compositional analysis
- Practical aspects of a multi-domain simulation platform
  - Graphical modeling of hybrid dynamics using Simulink and Stateflow
- Recap and conclusions

Compositional heterogeneous abstraction

Heterogeneous Verification



(pictorially)

Compositional
Heterogeneous Verification

If $M_0 = P_0 || Q_0$ and $M_1 = P_1 || Q_1$,
can we analyze Ps and Qs independently?

“Models as composition of components”
Analysis: Compositional Abstraction

Objective: Conclude heterogeneous abstraction of the composition by establishing that of the components

Rationale: Component’s local semantics defined in a behavior domain of smaller dimension

[RK13] A. Rajhans and B. H. Krogh, “*Compositional Heterogeneous Abstraction*,” 16th ACM International Conference on Hybrid Systems: Computation and Control, 2013

Leveraging compositionality for heterogeneous abstraction

Objective: Conclude heterogeneous abstraction of the composition by establishing that of the components

Rationale: Component’s local semantics defined in a behavior domain of smaller dimension

Need

- Behavior abstraction functions $\mathcal{A}$: behavior relations that are also functions
- Mappings between local/global behavior domains of the same type
- Mappings between local/global abstraction functions

$M_0 \sqsubseteq^A M_1$

Abstract composition behavior domain B_1

$$[\![M_0]\!]^{B_0} \sqsubseteq A^{-1}([\![M_1]\!]^{B_1})$$

Detailed composition behavior domain B_0

$P_0 \sqsubseteq^{A^P} P_1$

Abstract component behavior domain B_1^P

$$[\![P_0]\!]^{B_0^P} \sqsubseteq {A^P}^{-1}([\![P_1]\!]^{B_1^P})$$

Detailed component behavior domain B_0^P

$Q_0 \sqsubseteq^{A^Q} Q_1$

Abstract component behavior domain B_1^Q

$$[\![Q_0]\!]^{B_0^Q} \sqsubseteq {A^Q}^{-1}([\![Q_1]\!]^{B_1^Q})$$

Detailed component behavior domain B_0^Q

[RK13] A. Rajhans and B. H. Krogh, “*Compositional Heterogeneous Abstraction*,” 16th ACM International Conference on Hybrid Systems: Computation and Control, 2013

Compositionality conditions

conclude $[\![M_0]\!]^{B_0} \subseteq A^{-1}([\![M_1]\!]^{B_1})$

using $[\![P_0]\!]^{B_0^P} \subseteq {A^P}^{-1}([\![P_1]\!]^{B_1^P})$ and $[\![Q_0]\!]^{B_0^Q} \subseteq {A^Q}^{-1}([\![Q_1]\!]^{B_1^Q})$

Behavior localization (projections)

$B_0 \downarrow_0^P = B_0^P \qquad B_1 \downarrow_1^P = B_1^P$

Abstraction function localization (projections)

$A \Downarrow^P = A^P$

Commutative diagram

$$\begin{tikzcd} B_1' & B_1 \arrow[l, "\downarrow_1"'] \\ B_0' \arrow[u, "\mathcal{A}'"] & B_0 \arrow[l, "\downarrow_0"] \arrow[u, "\mathcal{A}"'] \end{tikzcd}$$

Centralized Development

Start with $\mathcal{A}$, *localize* to get $\mathcal{A}^{\mathcal{P}}$, $\mathcal{A}^{\mathcal{Q}}$

If localizations of $\mathcal{A}$ are $\mathcal{A}^{\mathcal{P}}$ and $\mathcal{A}^{\mathcal{Q}}$, then compositional heterogeneous abstraction via $\mathcal{A}$ holds

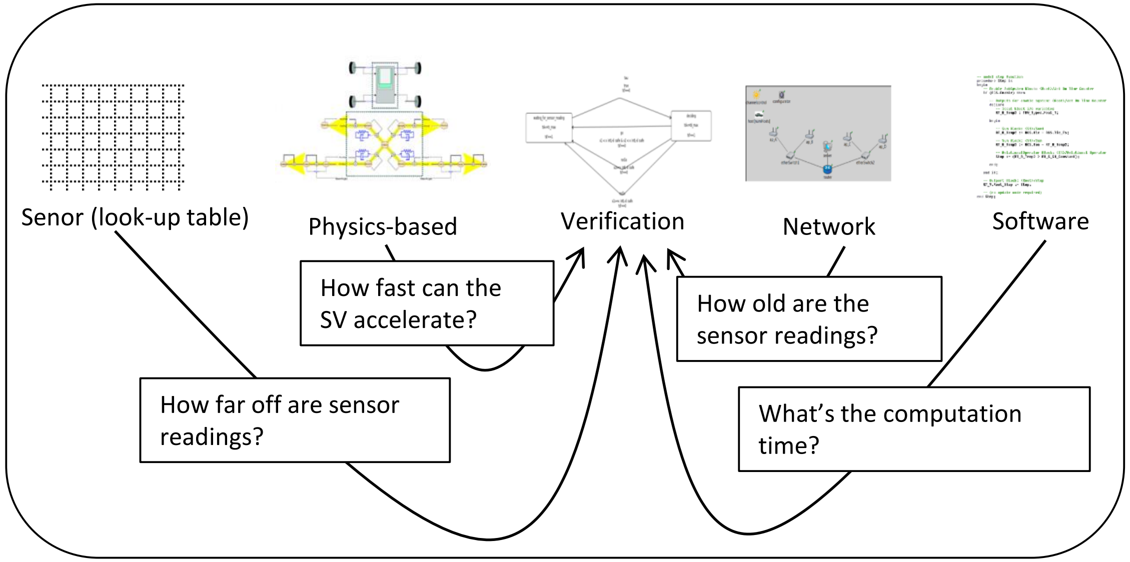
Decentralized Development

Start with $\mathcal{A}^{\mathcal{P}}$, $\mathcal{A}^{\mathcal{Q}}$, *globalize* to get $\mathcal{A}$

If globalizations of $\mathcal{A}^{\mathcal{P}}$, $\mathcal{A}^{\mathcal{Q}}$ are consistent (call it $\mathcal{A}$), then compositional heterogeneous abstraction via $\mathcal{A}$ holds

[RK13] A. Rajhans and B. H. Krogh, "*Compositional Heterogeneous Abstraction*," 16th ACM International Conference on Hybrid Systems: Computation and Control, 2013

Semantic assumptions as parameter constraints



Dependencies that cut across formalisms captured as parameter constraints

Ensures semantic (parameter) consistency using external SMT solvers or provers

Problem

- *Semantic interdependencies* across formalisms
- *Consistency*

Challenge

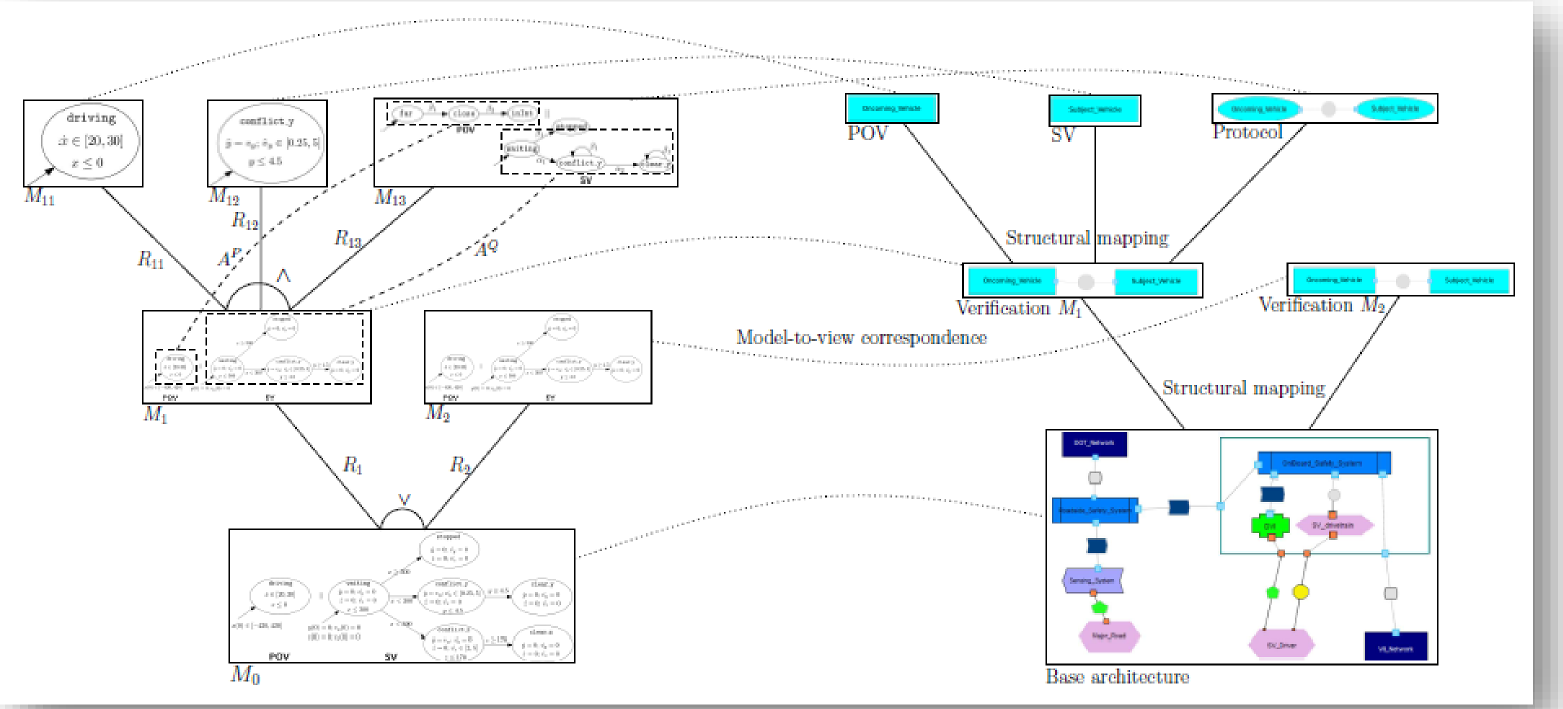
- *Formal representation* that is *universal* to all modeling formalisms

Approach

- interdependencies as an *auxiliary constraint on parameters*
- Find *effective constraint* on given model/spec. parameters (existential quantification)
- Use *SMT solvers* or *theorem provers* to prove consistency

[RBL+11] A. Rajhans, A. Bhave, S. Loos, B. H. Krogh, A. Platzer, and D. Garlan, "*Using Parameters in Architectural Views to Support Heterogeneous Design and Verification*," 50th IEEE Conference on Decision and Control, 2011

Completing the picture: Semantic and structural hierarchies

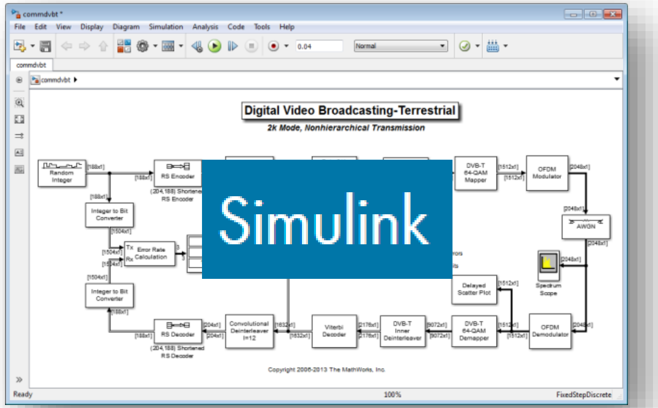


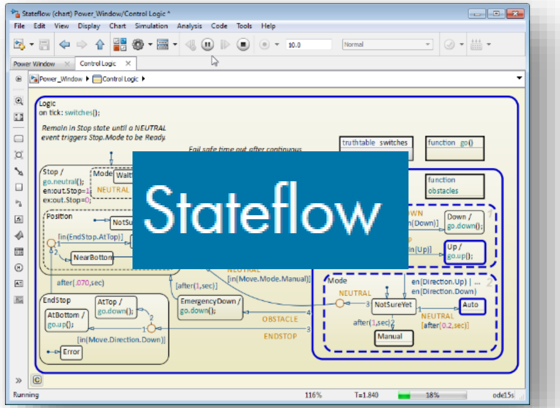
Semantic side **Structural side**

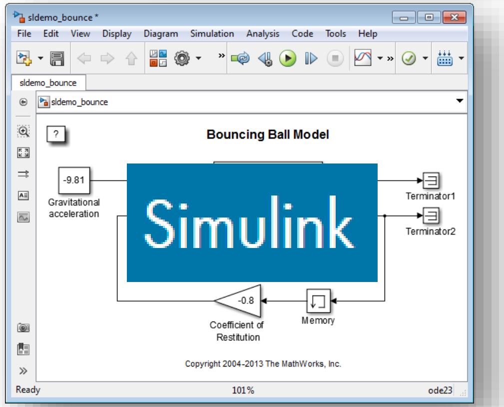
[RBR+14] A. Rajhans et al., "*Supporting Heterogeneity in Cyber-Physical System Architectures*", IEEE Transactions on Automatic Control's Special Issue on Control of Cyber-Physical Systems, Vol. 59, Issue 12, pages 3178-3193

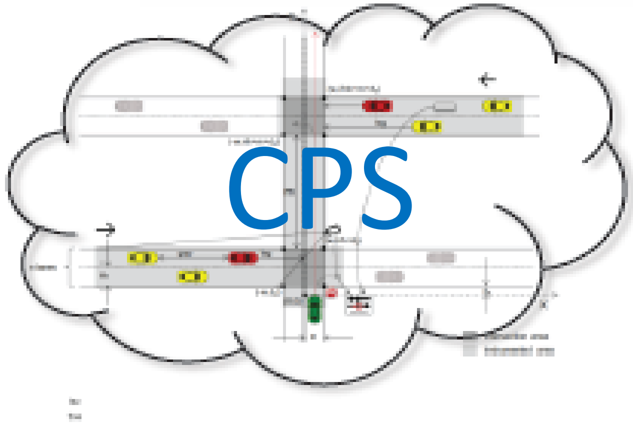
Outline

- Introduction
- Theoretical aspects of multi-paradigm model-based design for CPS
  - Architecture modeling and structural analysis
  - Semantic analysis and heterogeneous verification
  - Compositional analysis
- ➢ Practical aspects of a multi-domain simulation platform
  - Graphical modeling of hybrid dynamics using Simulink and Stateflow
- Recap and conclusions

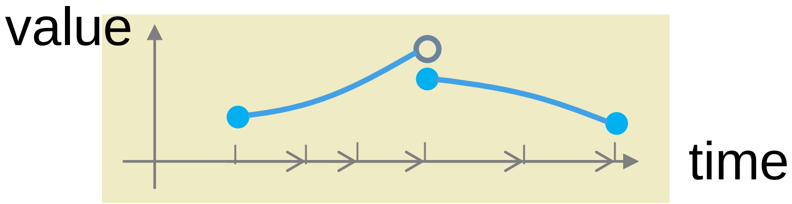

Simulink
Digital Video Broadcasting-Terrestrial
2k Mode, Nonhierarchical Transmission


Stateflow


Simulink
Bouncing Ball Model
Gravitational acceleration
Terminator1
Terminator2
Coefficient of Restitution
Memory
Copyright 2004-2013 The MathWorks, Inc.


CPS

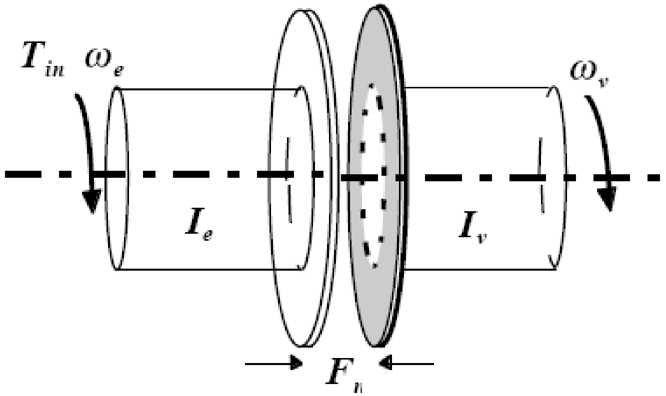
Modeling hybrid (discrete + continuous) dynamics graphically using Simulink and Stateflow

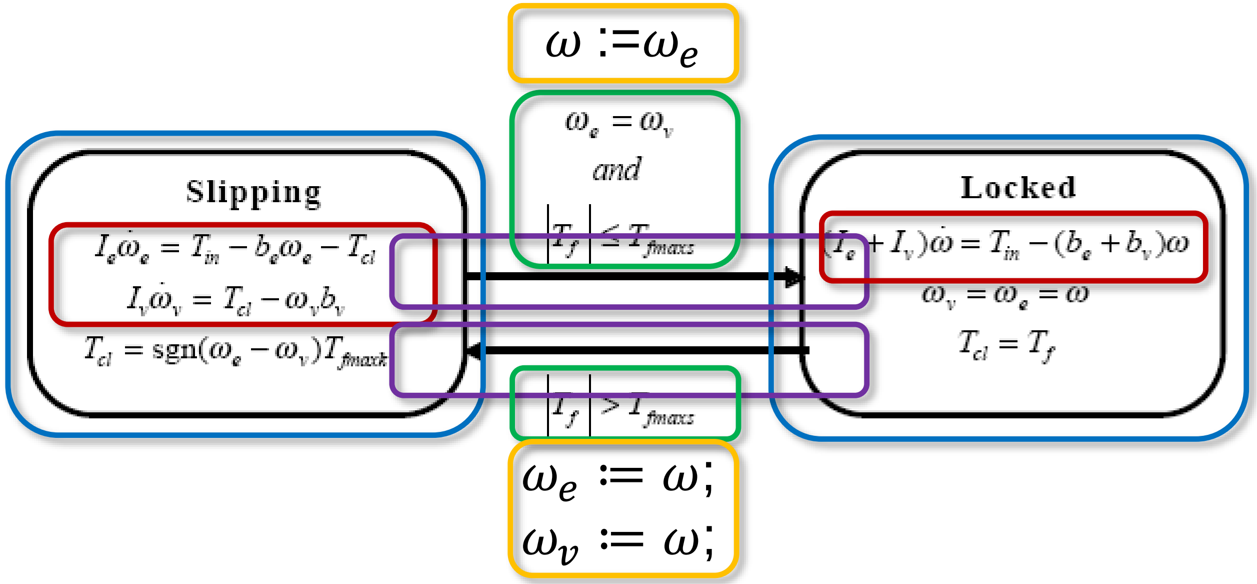


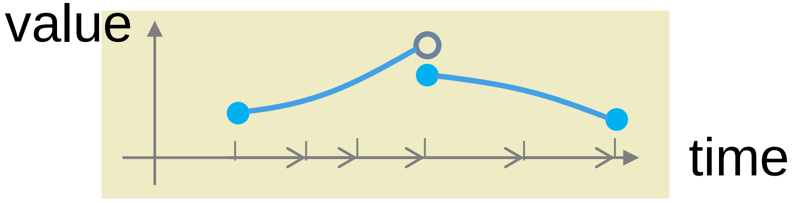
Simulink Stateflow

Hybrid dynamics arise in CPS models quite often

Example: clutch



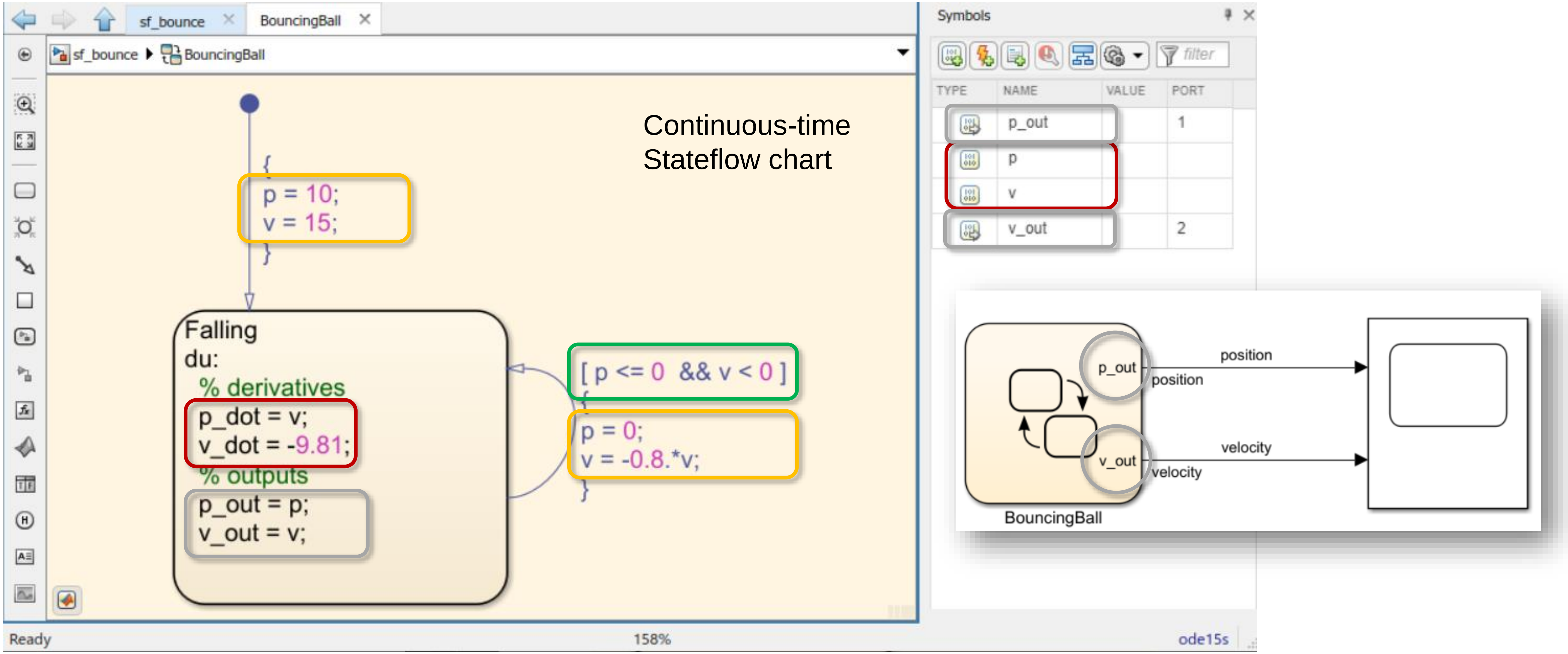




- Need to model and orchestrate
  1. Continuous dynamics
  2. Discrete modes
  3. Mode switching
     - Guard conditions
     - State handoff

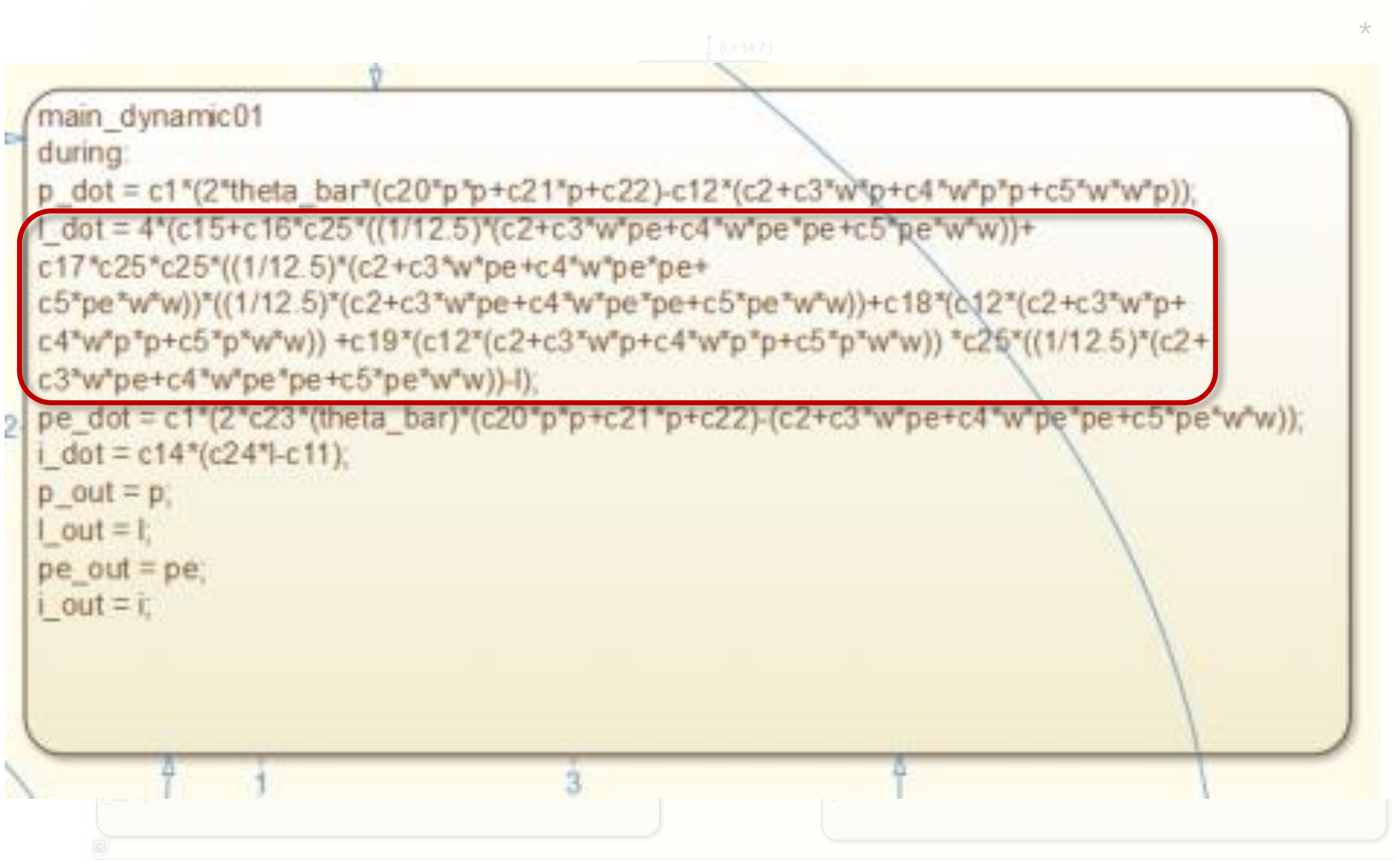
https://www.mathworks.com/help/simulink/slref/building-a-clutch-lock-up-model.html

Modeling hybrid dynamics [Option 1]: Entirely in Stateflow



`sf_bounce`

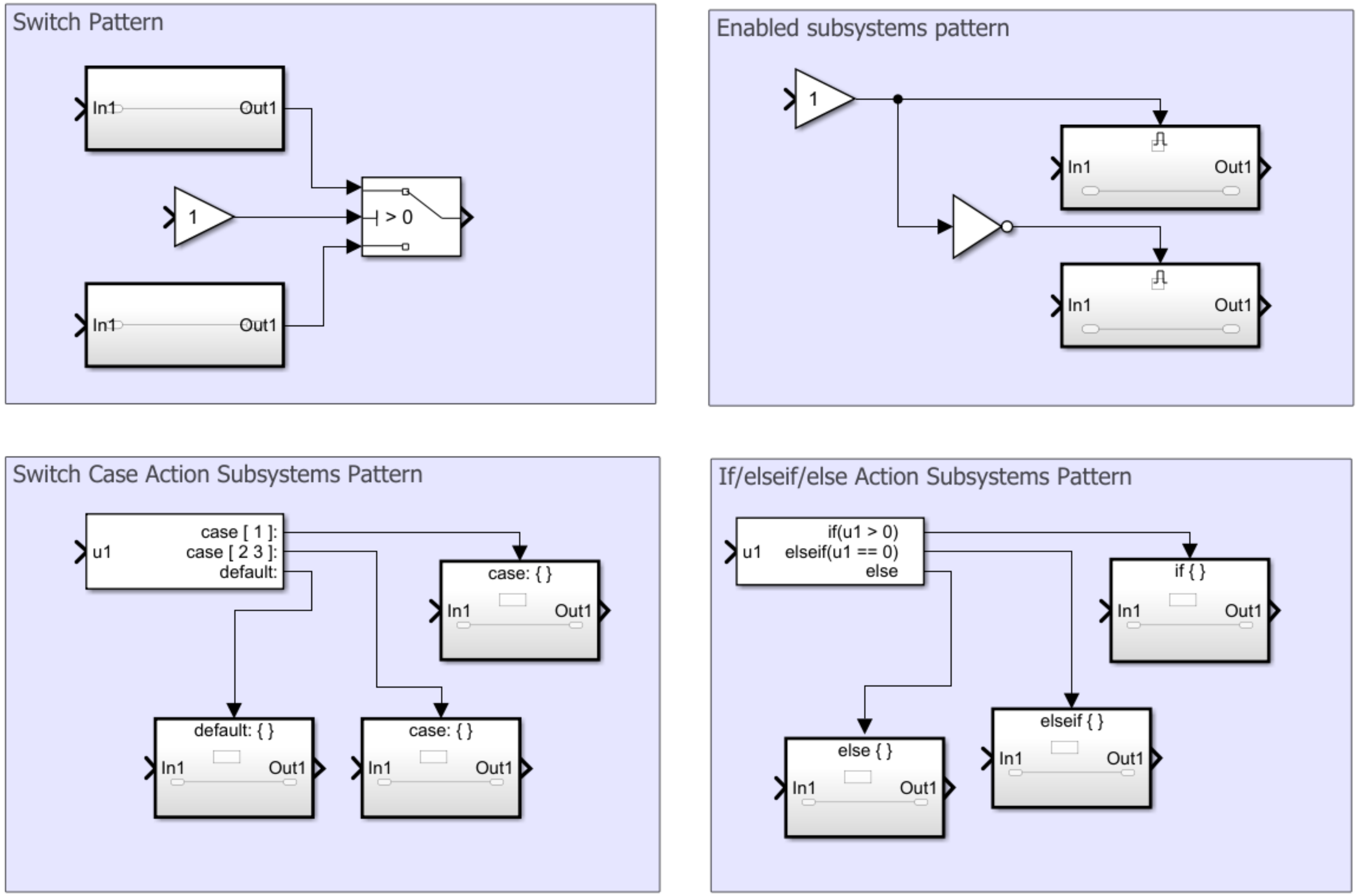
Can get cumbersome for complex ODE dynamics



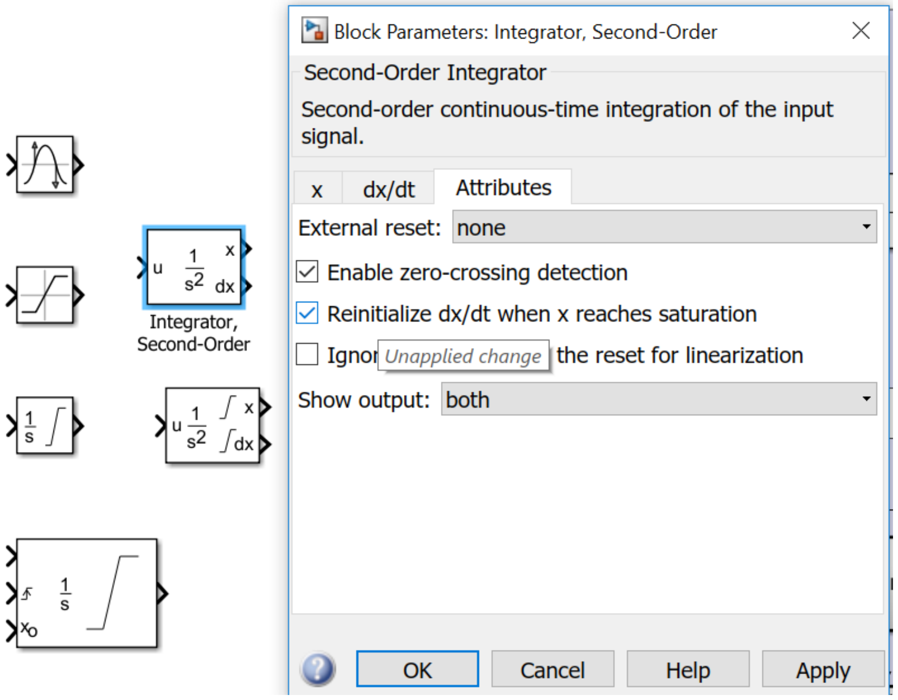
+ Intuitive for discrete dynamics
- ‘hand-coding’, difficult to debug

[Meeting a Powertrain Verification Challenge]
[Progress on Powertrain Verification Challenge with C2E2]

Modeling hybrid dynamics [Option 2]: Entirely in Simulink

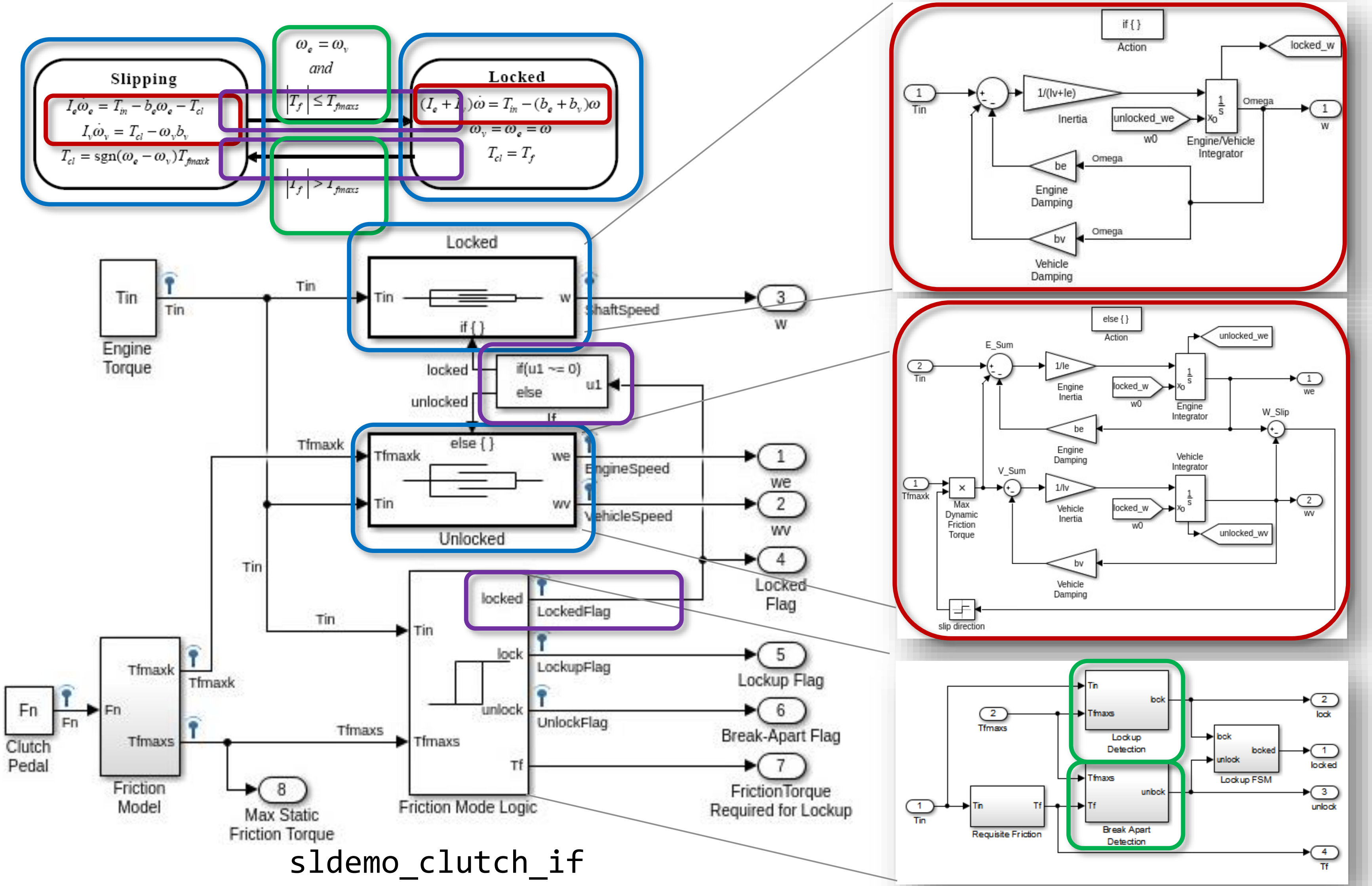


Explicit mode switching examples



Implicit mode switching examples

Modeling hybrid dynamics [Option 2]: Entirely in Simulink



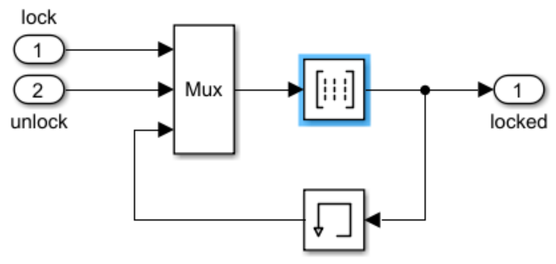
sldemo_clutch_if

Lock	Unlock	Lock$^-$	Locked
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

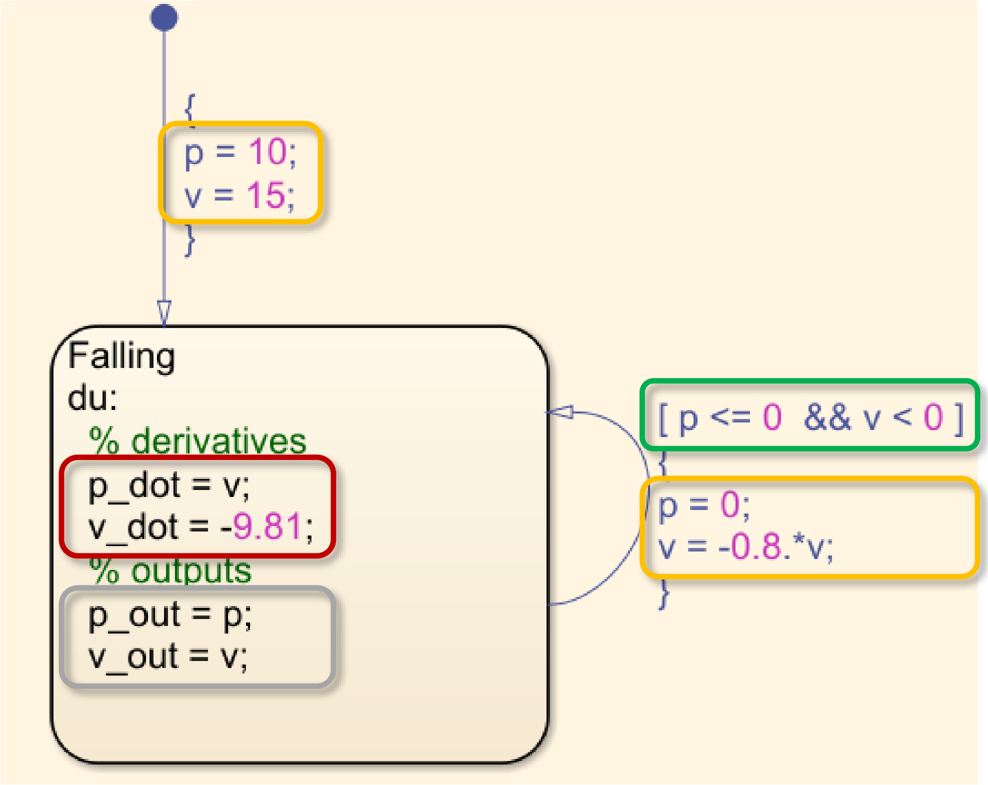
Parameters

Truth table:

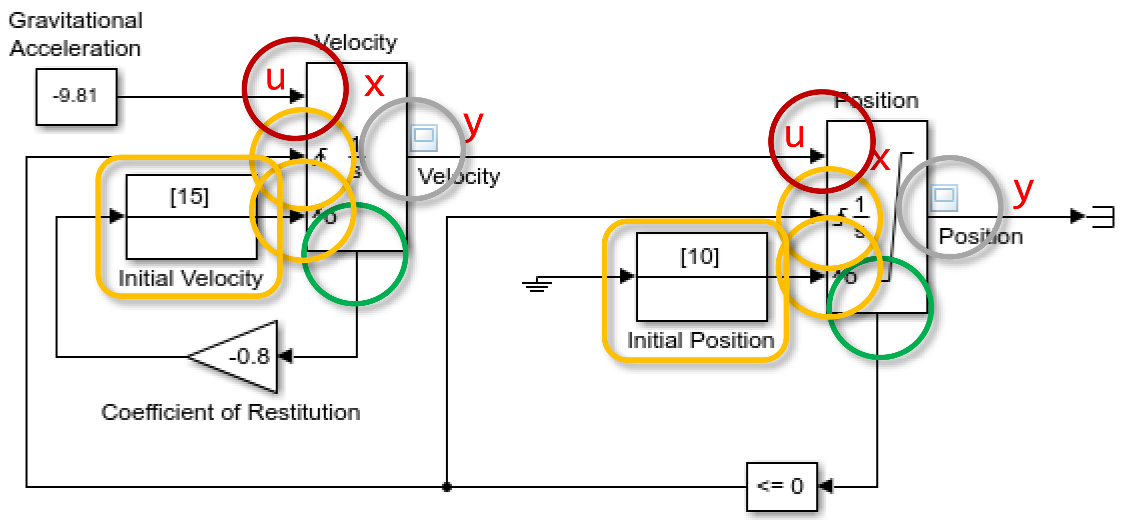
[0;1;0;0;1;1;1;0]



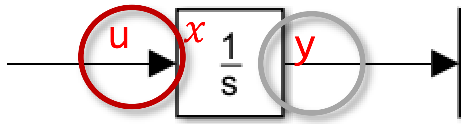
State handoff considerations



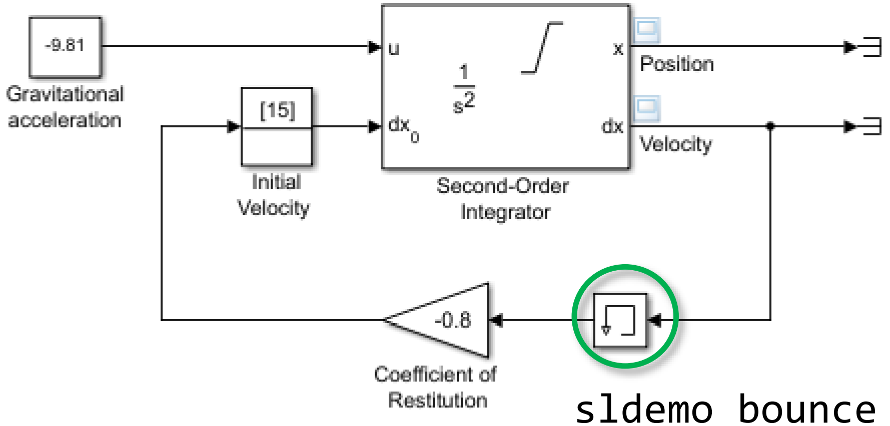
sf_bounce



sldemo_bounce_two_integrators

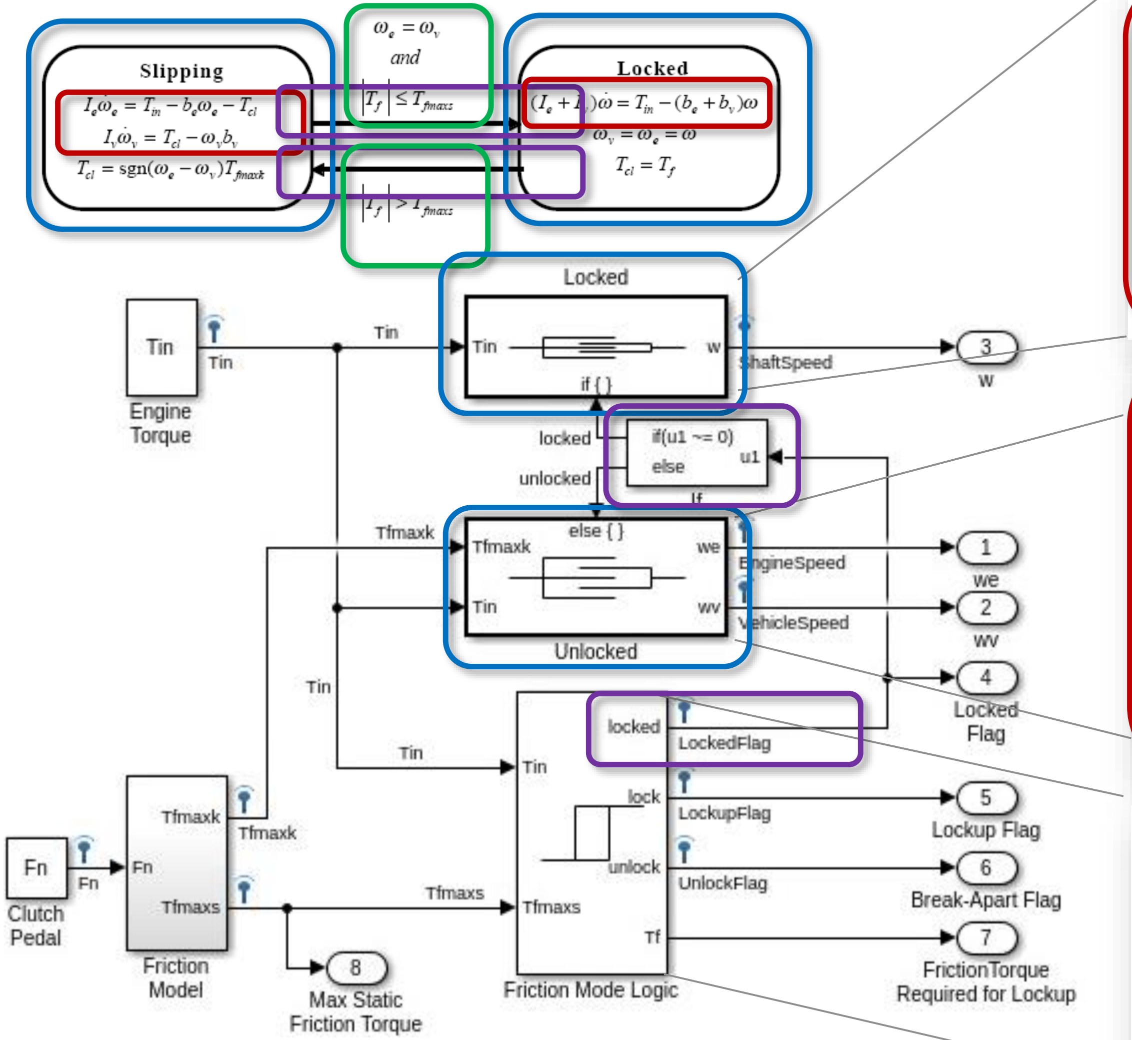


$$y = x$$
$$\dot{x} = u$$

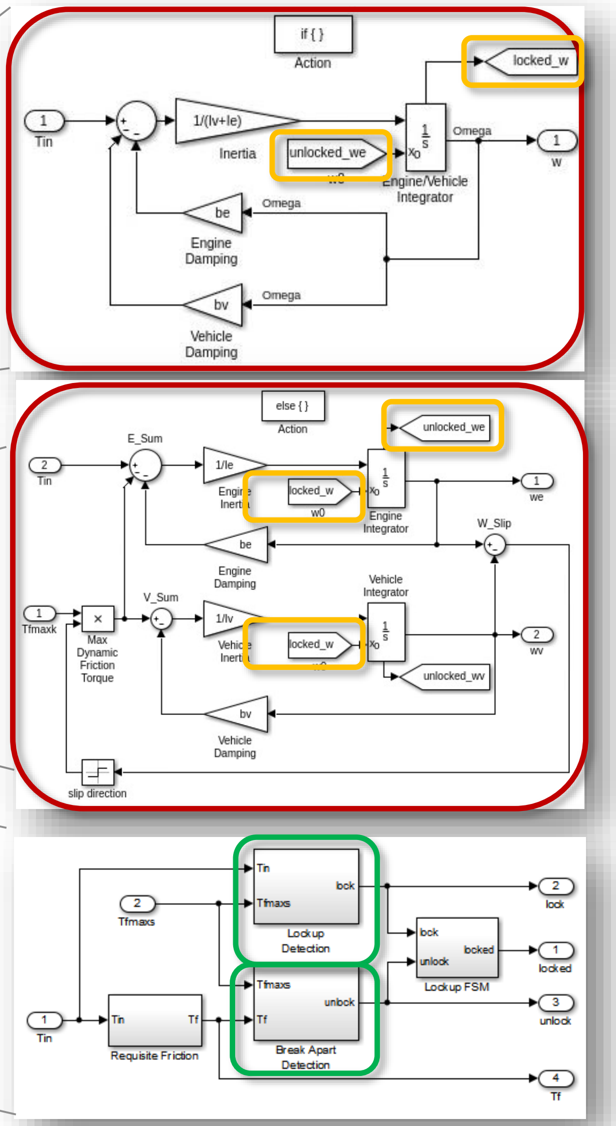


sldemo_bounce

Modeling hybrid dynamics [Option 2]: Entirely in Simulink



sldemo_clutch_if

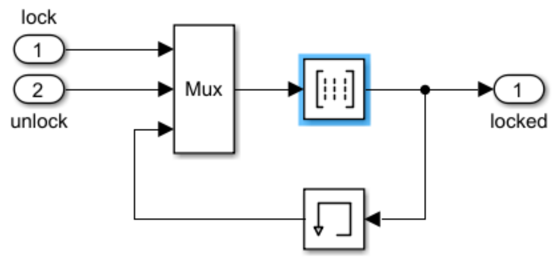


Lock	Unlock	Lock_	Locked
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

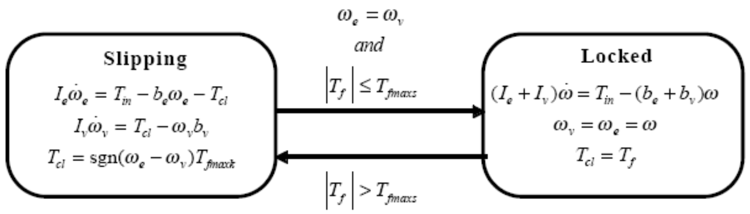
Parameters

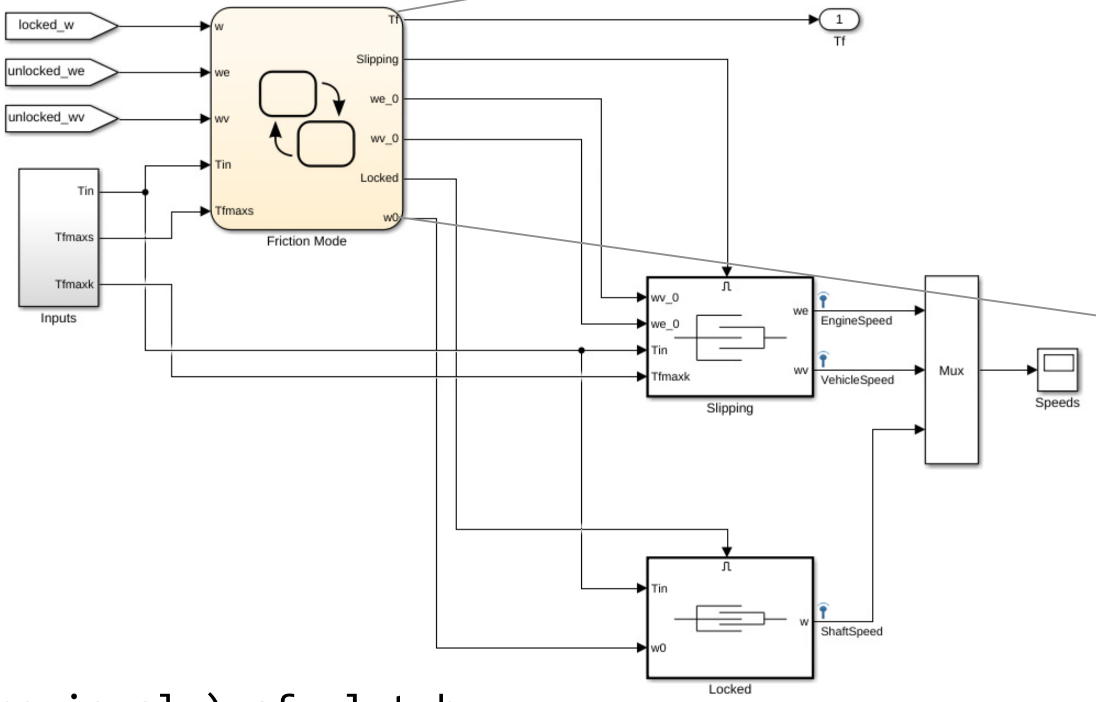
Truth table:

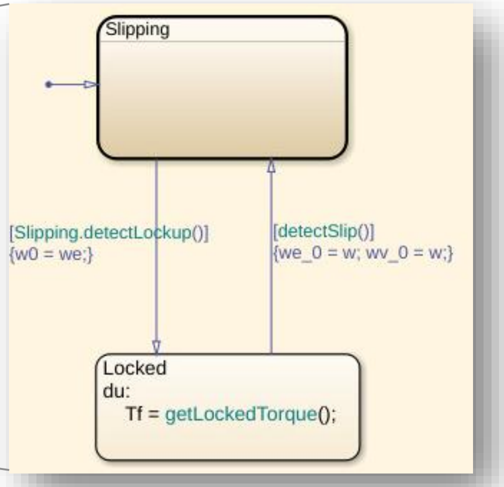
[0;1;0;0;1;1;1;0]



Modeling hybrid dynamics [Option 3]: Stateflow drives Simulink



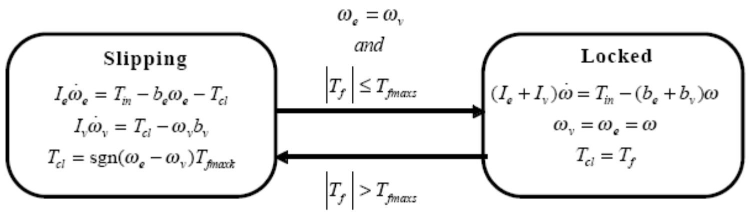


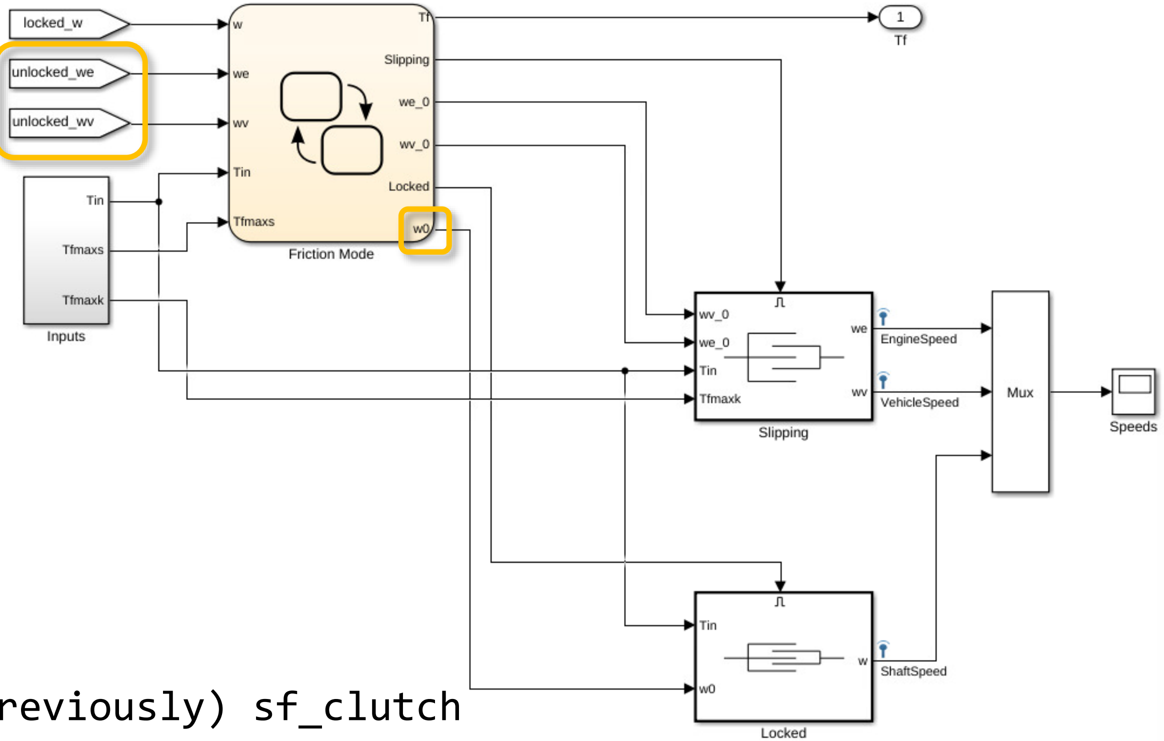


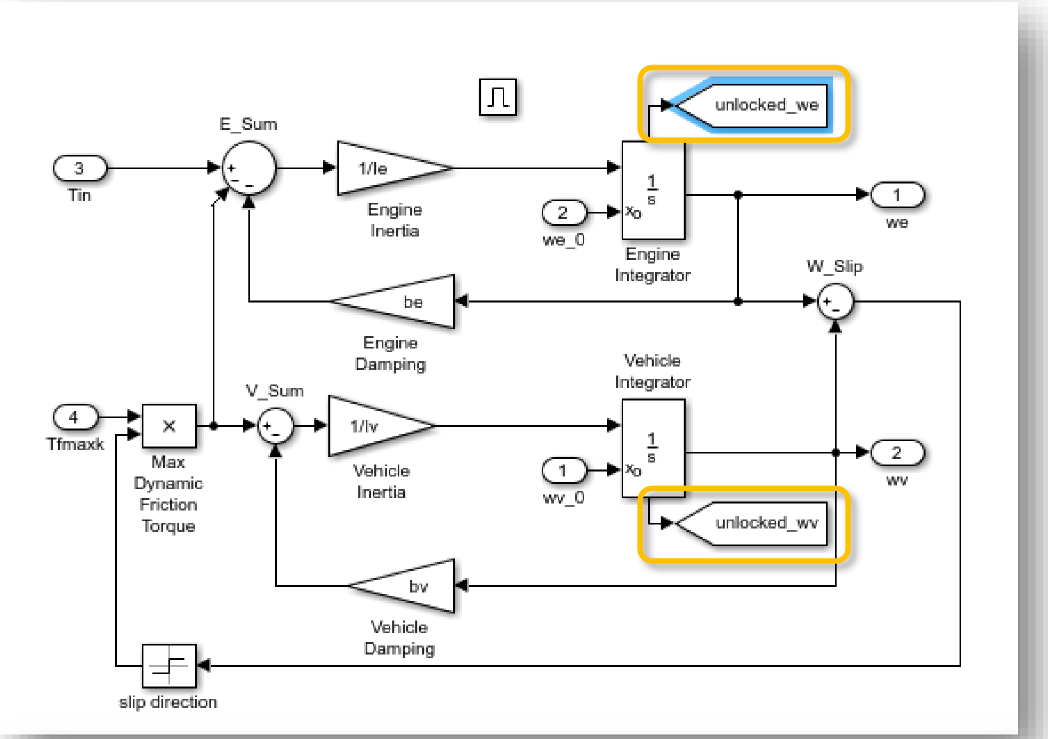
+ Intuitive for continuous dynamics
+ Intuitive for discrete dynamics
- Intuitive for hybrid dynamics? Can do better
  - Signal lines between Simulink and Stateflow
  - State handoff

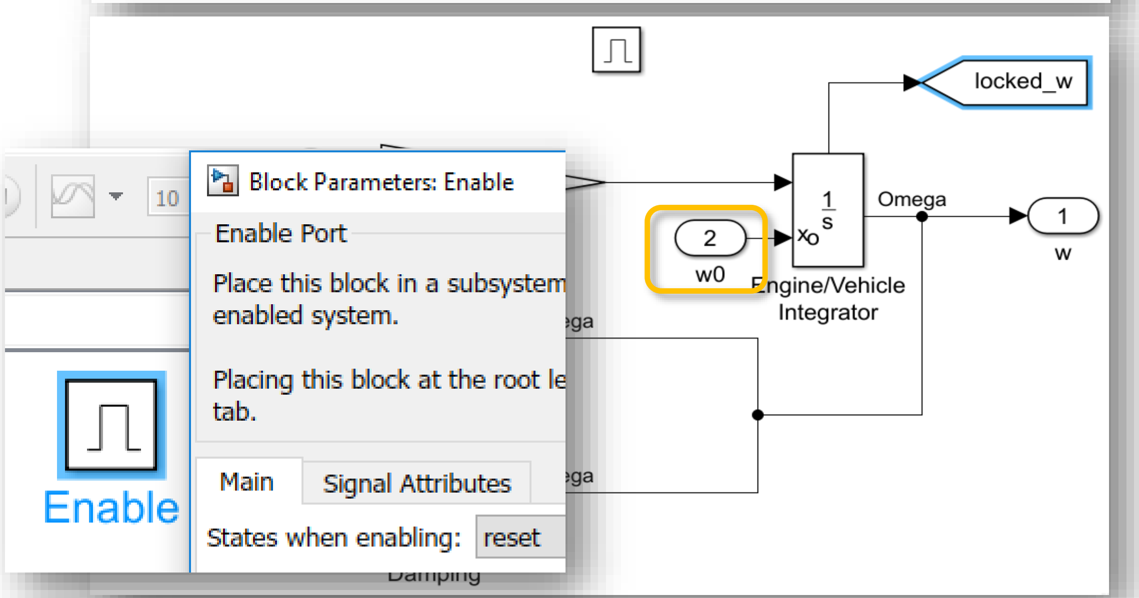
```
(previously) sf_clutch
(now) sf_clutch_enabled_subsystems
```

State handoff considerations









(previously) sf_clutch

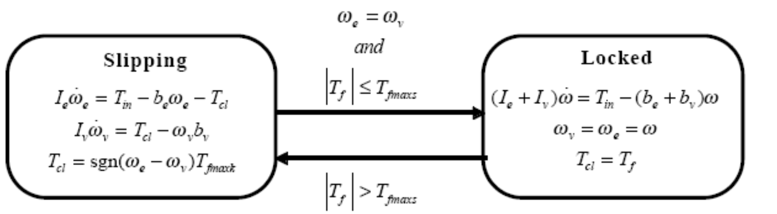
(now) sf_clutch_enabled_subsystems

Simulink-based states in Stateflow

R2017b

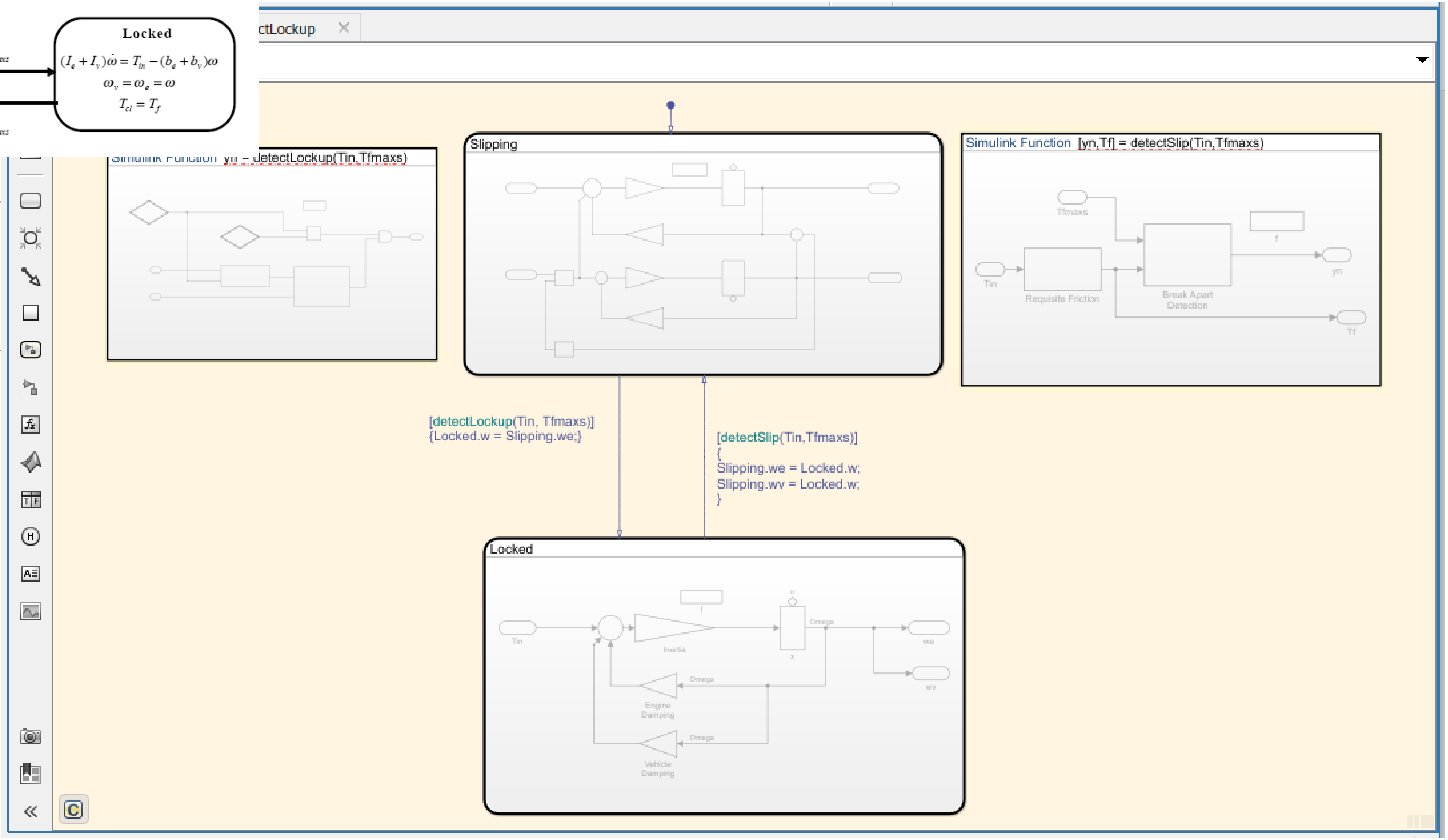
[RAC+18a] A. Rajhans et al., "*Graphical Modeling of Hybrid Dynamics with Simulink and Stateflow*," 21st ACM International Conference on Hybrid Systems: Computation and Control, 2018

Simulink-based states in Stateflow

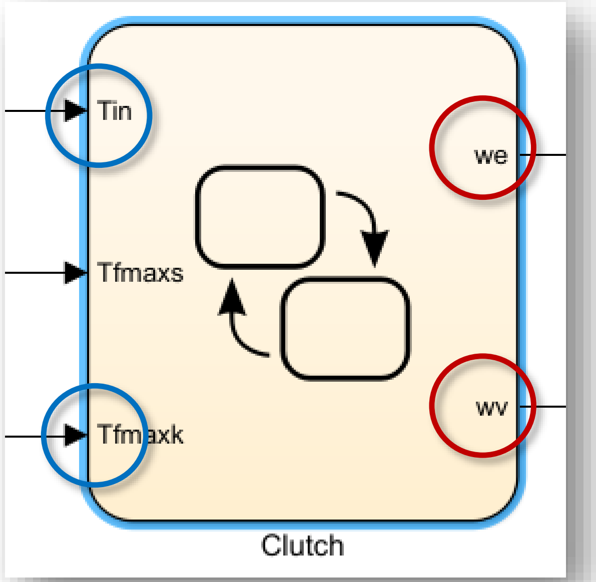


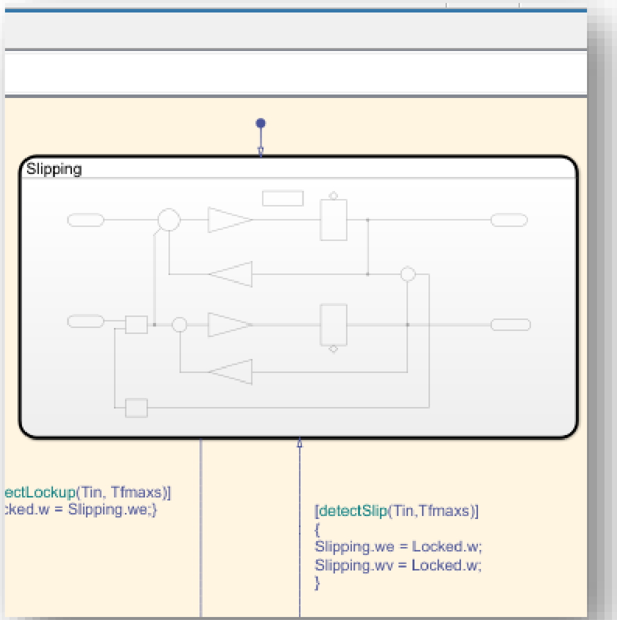
regular state

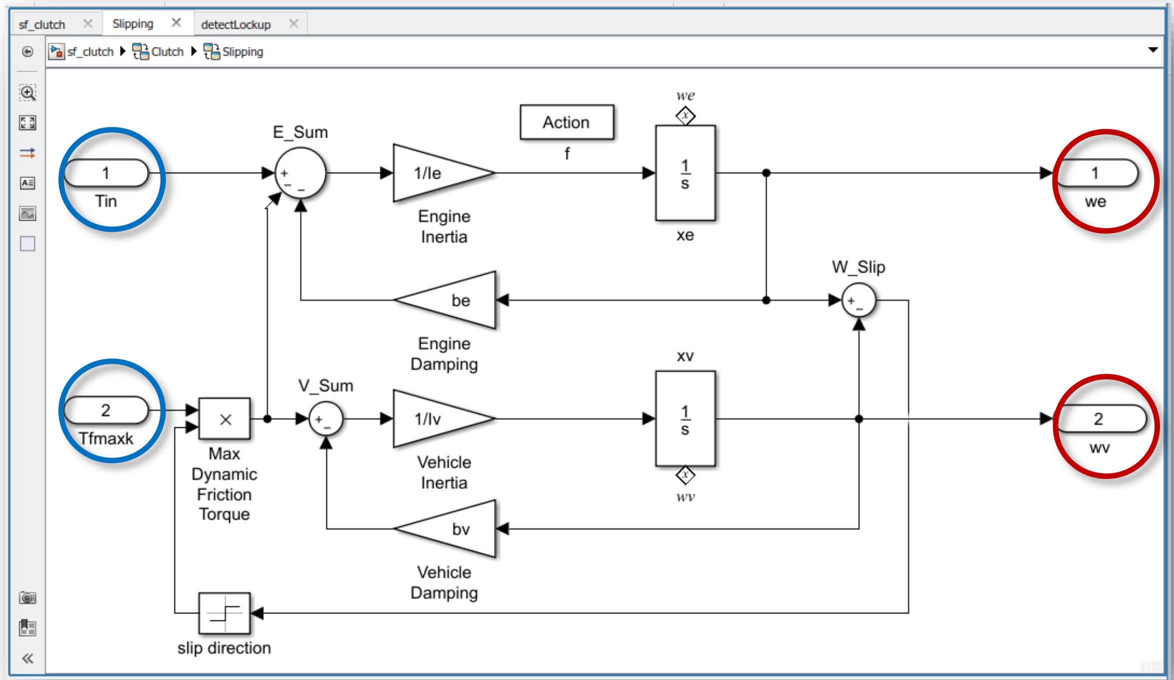
Simulink-based state



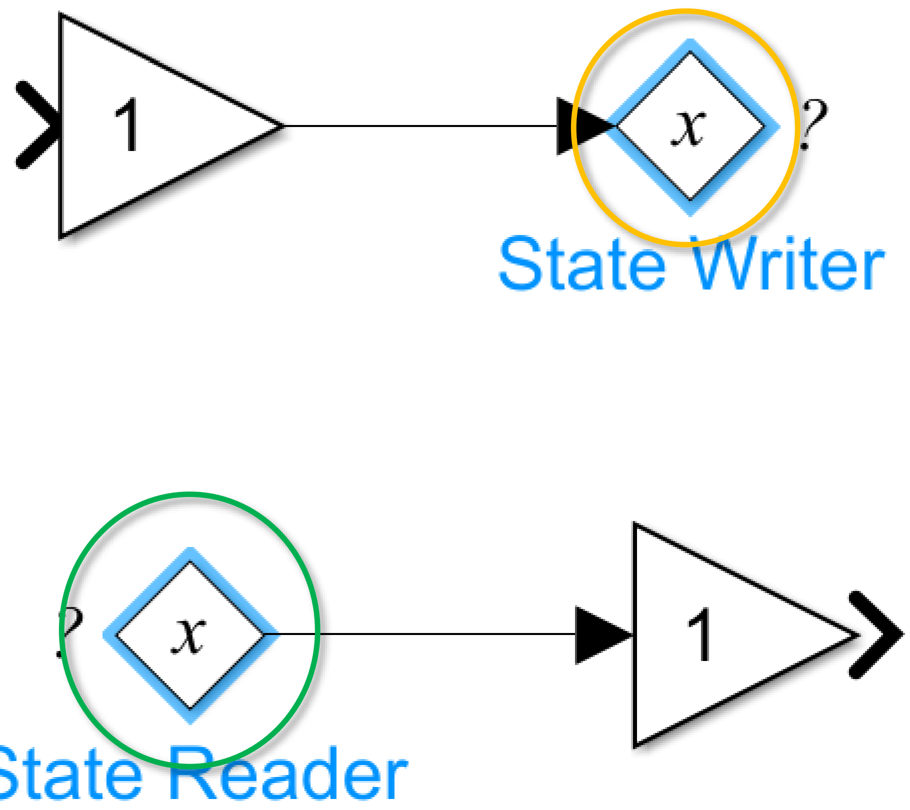
Simulink-based states in Stateflow

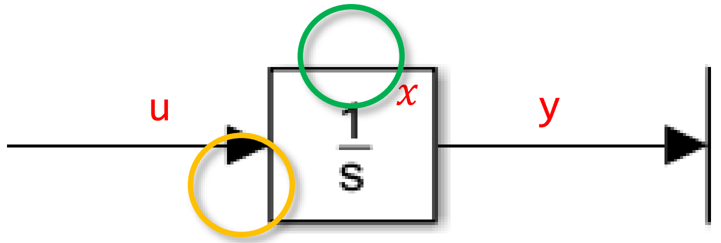






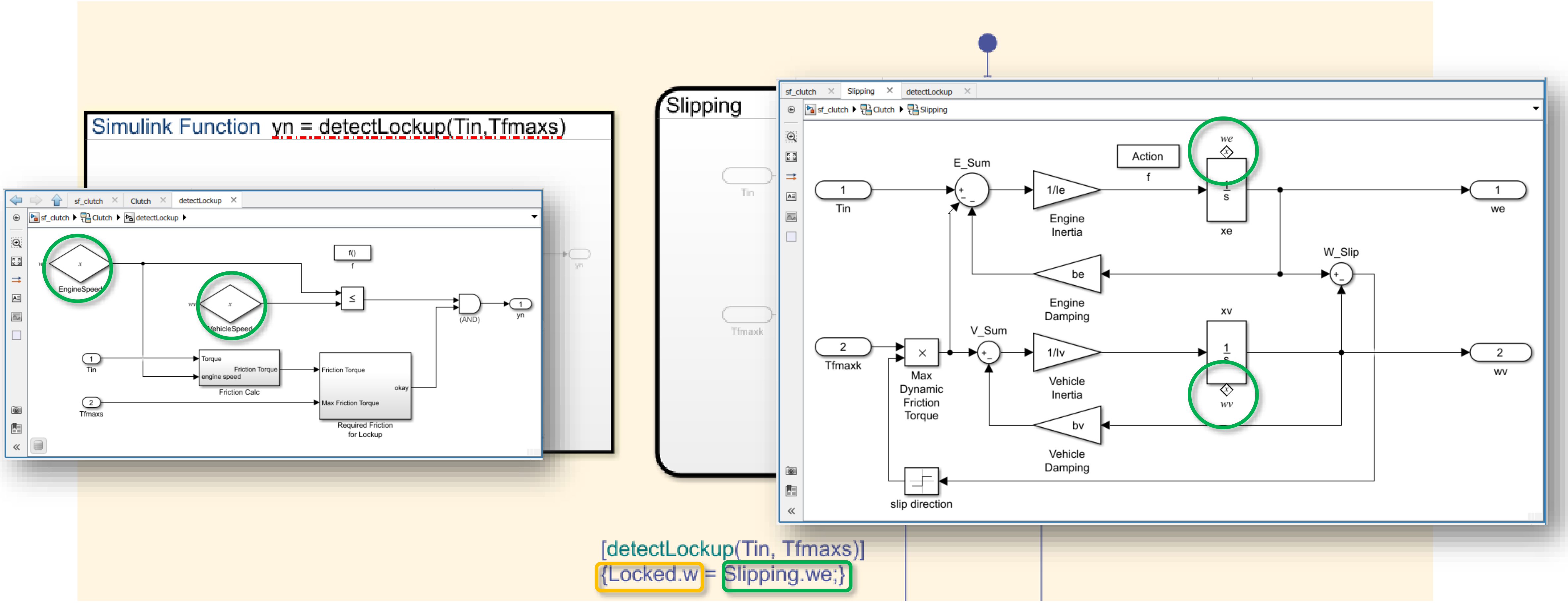
Graphical remote state access



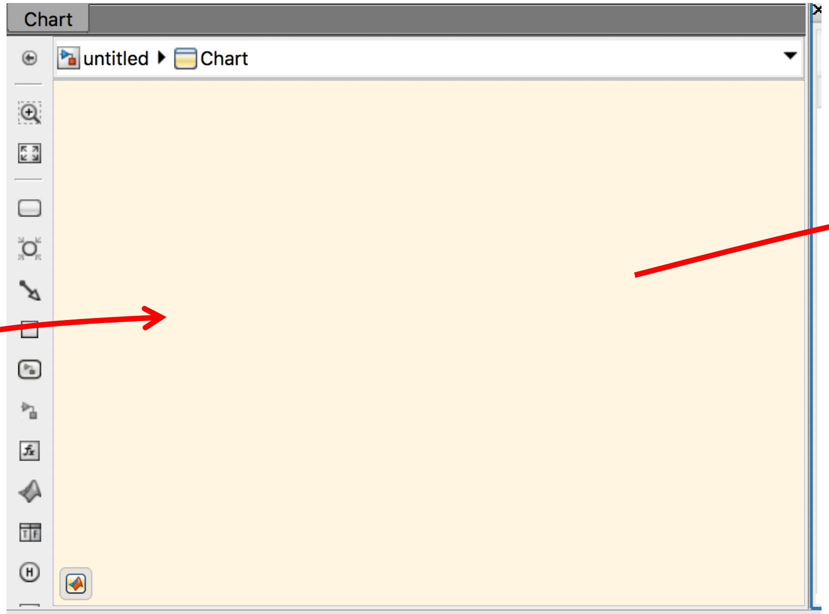


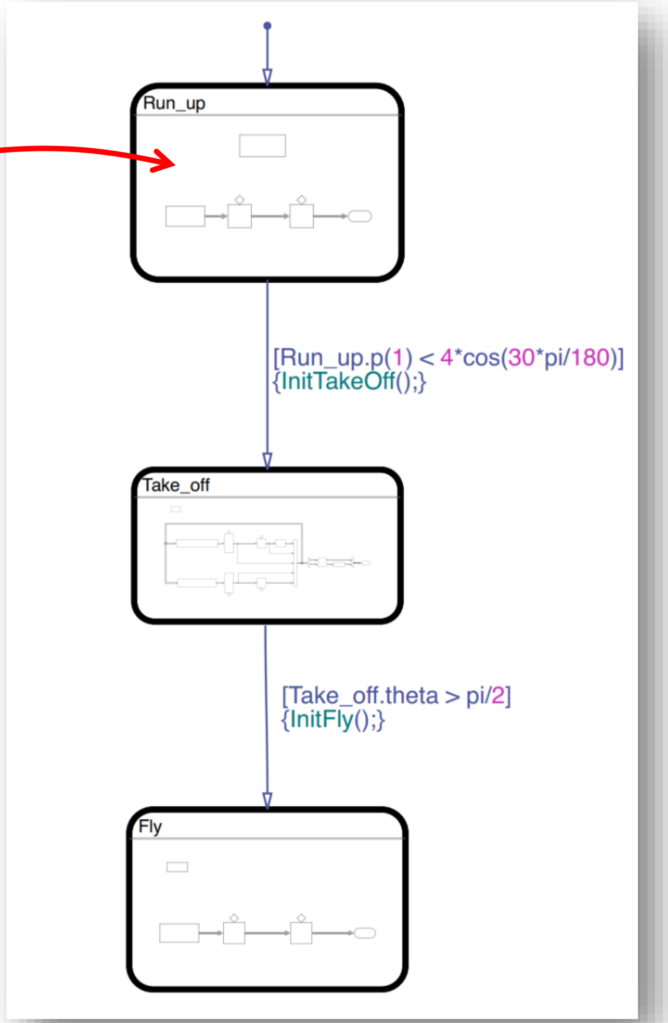
$$y = x$$
$$\dot{x} = u$$

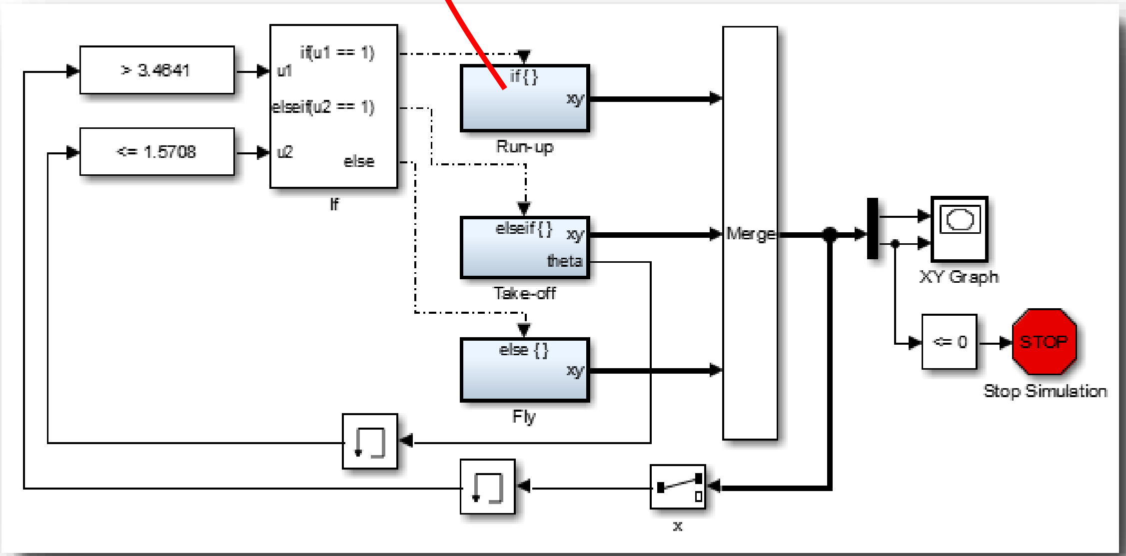
Graphical and textual remote state access



Easy copy-paste workflow







Outline

- Introduction
- Theoretical aspects of multi-paradigm model-based design for CPS
  - Architecture modeling and structural analysis
  - Semantic analysis and heterogeneous verification
  - Compositional analysis
- Practical aspects of a multi-domain simulation platform
  - Graphical modeling of hybrid dynamics using Simulink and Stateflow
- ➢ Recap

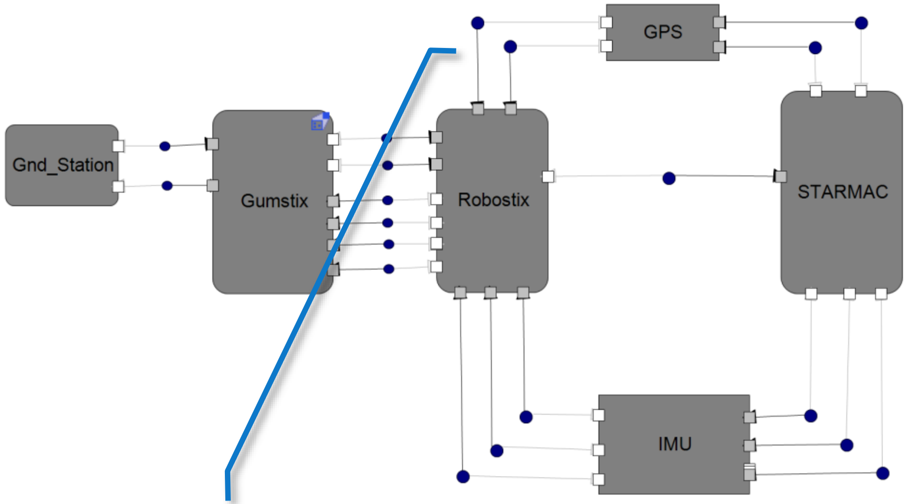
Recap

- CPS have a global societal scale impact – challenges and opportunities
- Models are used in design and operation of complex CPS

- Heterogeneity due to multiple paradigms presents a research challenge
  - Architecture presents an anchoring framework and enables structural analysis
  - Behavior domain associations enable semantic analysis

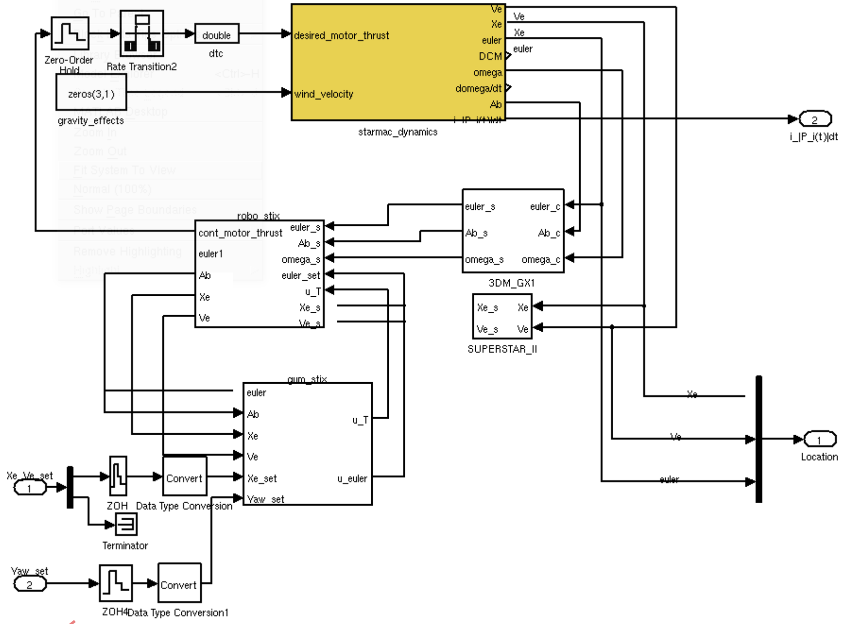
- Particulars of bridging the gap across formalisms in a simulation platform
  - Discussed one specific connection between two specific formalisms
  - Many other interesting details across other formalisms

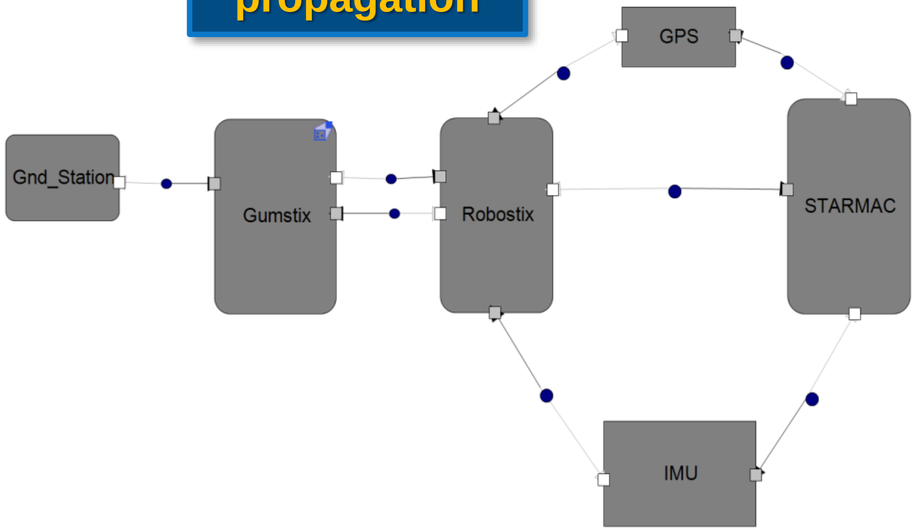
Simulink Architecture ←→ Simulink Model: Manual Step in 2010



Behavior construction

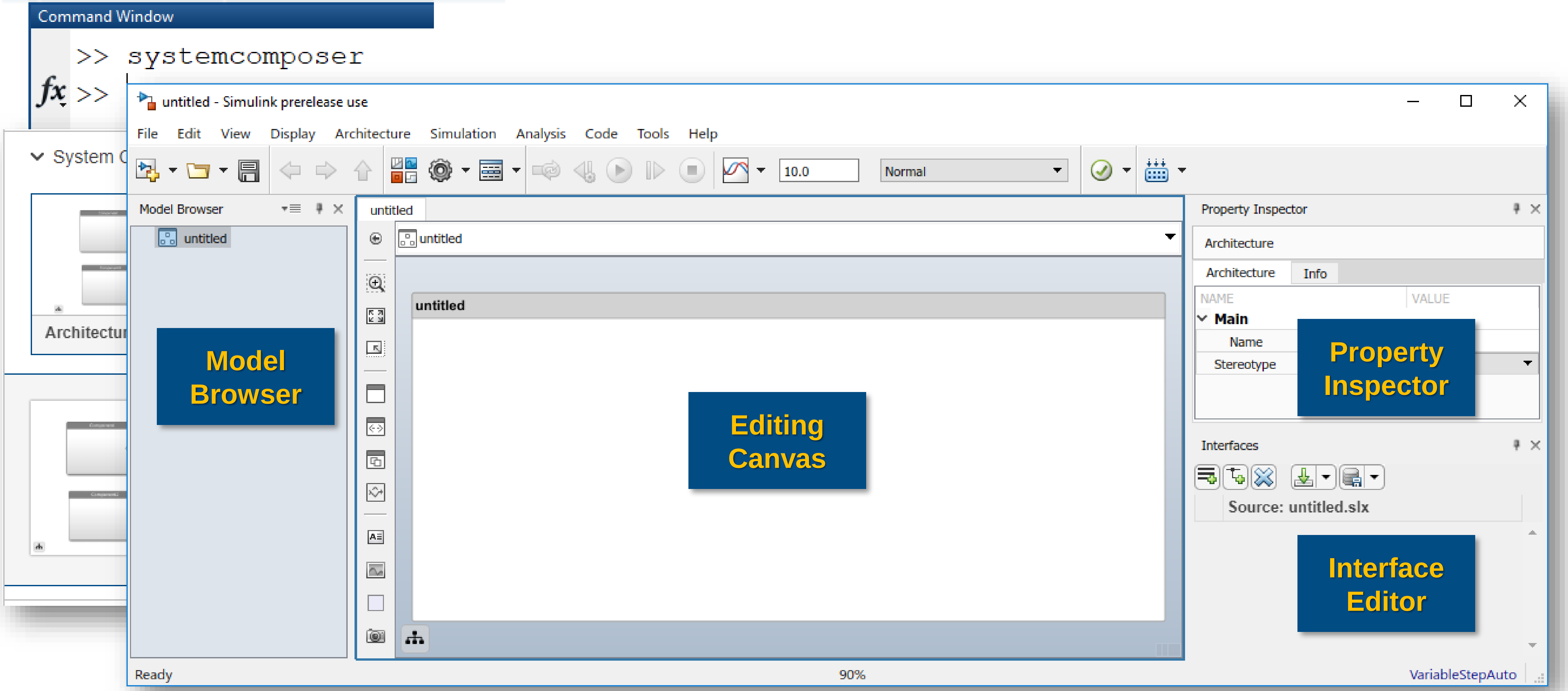
Fix propagation





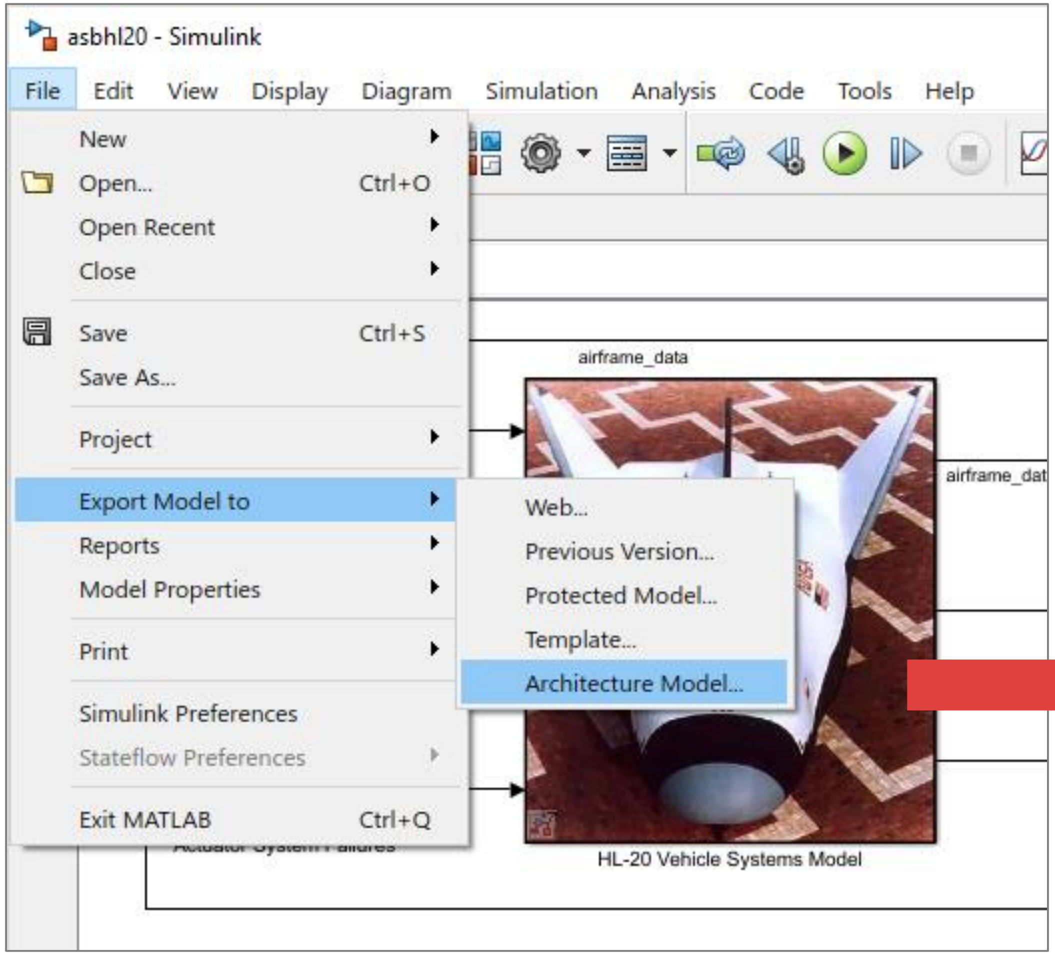
Structure extraction

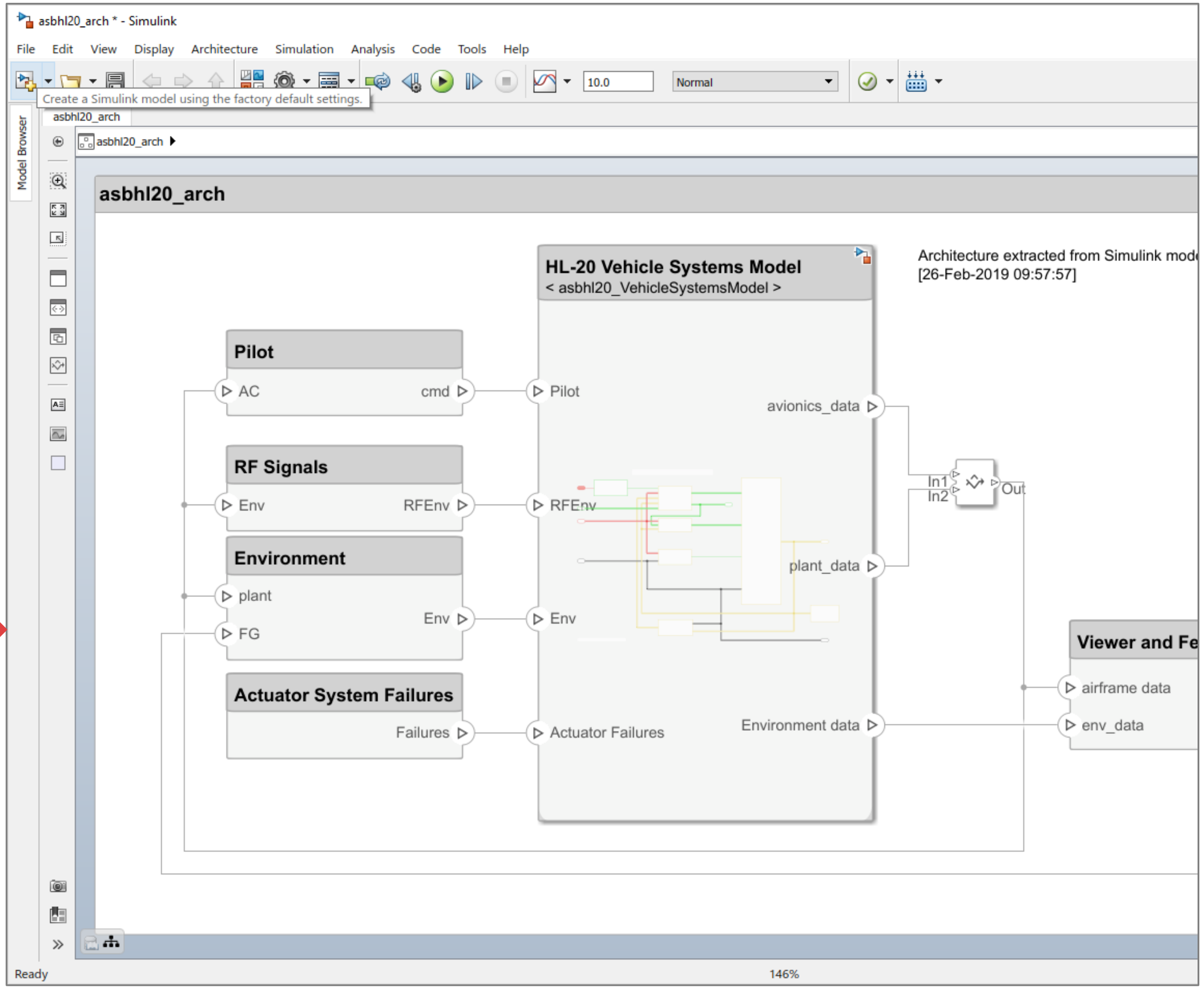
System Composer NEW PRODUCT



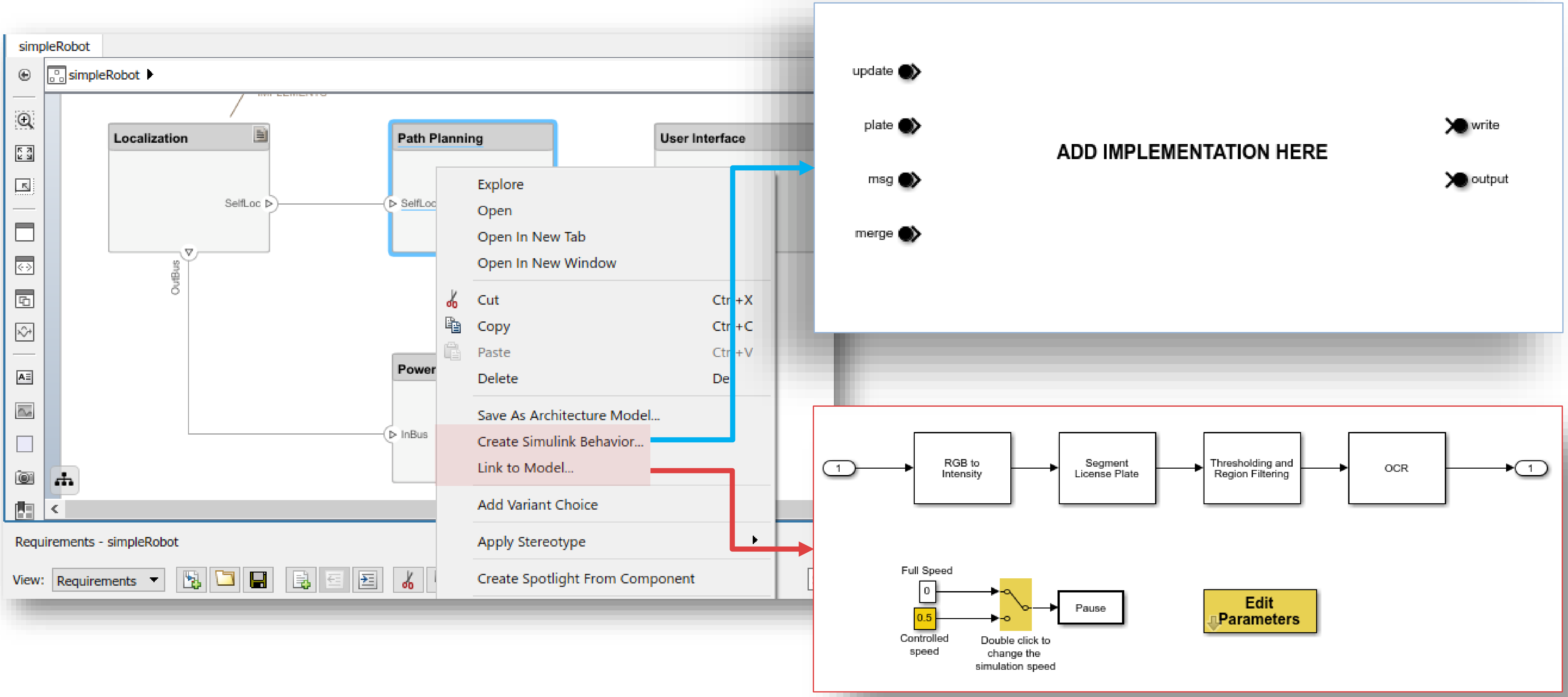
https://www.mathworks.com/products/system-composer.html

Simulink to architecture





Architecture to Simulink



Interesting connections across other formalisms

▪ Messages	Simulink	(drives)	SimEvents,
	Stateflow	(drives)	SimEvents
▪ Function calls	SimEvents	(calls)	Simulink,
	Stateflow	(calls)	Simulink
▪ MATLAB Function	Stateflow	(calls)	MATLAB,
	Simulink	(uses)	MATLAB
▪ System Objects	MATLAB	(calls)	Simulink
▪ Stateflow for MATLAB	MATLAB	(calls)	Stateflow
▪ MATLAB DES Block	SimEvents	(uses)	MATLAB
▪ DES Chart	SimEvents	(uses)	Stateflow

Acknowledgments

- Architectures and multi-model heterogeneous design and analysis
  - Ajinkya Bhave, Bruce Krogh, David Garlan, Ivan Ruchkin, Bradley Schmerl
- Graphical hybrid automata using Simulink and Stateflow
  - Srinath Avadhanula, Alongkrit Chutinan, Pieter Mosterman, Fu Zhang

References

- [RCS+09] A. Rajhans et al., "*An Architectural Approach to the Design and Analysis of Cyber-Physical Systems*," Third International Workshop on Multi-Paradigm Modeling (MPM), 2009. [Preprint (PDF)]
- [BDK+10b] A. Bhave et al., "*Augmenting Software Architectures with Physical Components*," Embedded Real Time Software and Systems (ERTS^2), 2010. [Preprint (PDF)]
- [RBL+11] A. Rajhans et al., "*Using Parameters in Architectural Views to Support Heterogeneous Design and Verification*," 50th IEEE CDC, 2011. [Preprint (PDF)]
- [RK12] A. Rajhans and B. H. Krogh, "*Heterogeneous Verification of Cyber-Physical Systems Using Behavior Relations*," 15th ACM HSCC, 2012. [Preprint (PDF)]
- [RK13] A. Rajhans and B. H. Krogh, "*Compositional Heterogeneous Abstraction*," 16th ACM HSCC, 2013. [Preprint (PDF)]
- [R13] A. Rajhans, "*Multi-Model Heterogeneous Verification of Cyber-Physical Systems*," **PhD Thesis**, Carnegie Mellon University, 2013. [Thesis (Abstract with a link to Fulltext PDF)]
- [RBR+14] A. Rajhans et al., "*Supporting Heterogeneity in Cyber-Physical System Architectures*", IEEE TAC's Special Issue on Control of CPS, Vol. 59, Issue 12, pages 3178-3193. [Preprint (PDF)]
- [RAC+18a] A. Rajhans et al., "*Graphical Modeling of Hybrid Dynamics with Simulink and Stateflow*," 21st ACM HSCC, 2018. ***Best Repeatability Evaluation Award Finalist***. [Preprint (PDF)]

https://arajhans.github.io
send email for unpublished papers: arajhans@mathworks.com

Accelerating the pace of engineering and science