This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 643921.

TOOLS INTEGRATION

UnCoVerCPS toolchain

Goran Frehse, *UGA*
Xavier Fornari, *ANSYS*

UnCoVerCPS Workshop – June 6th,2018

CYBER-PHYSICAL SYSTEMS CHALLENGES

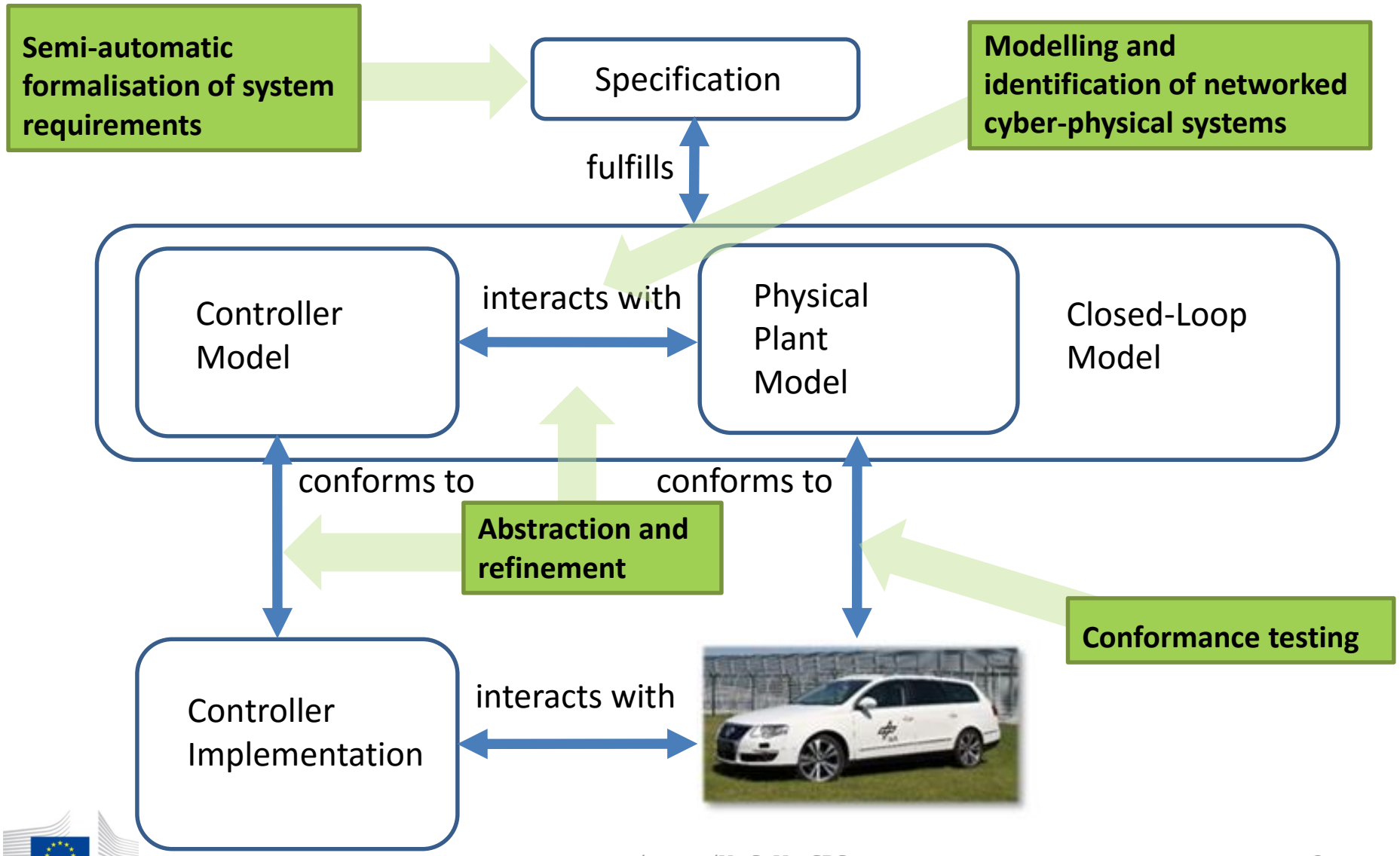
- Functionality and complexity are growing
- Mix of discrete and continuous dynamics (hard to control and verify)
- Unknown and changing environments
- Different tools with no interactions
- Difficulty to transfer results from on domain to the other

Source: CMU

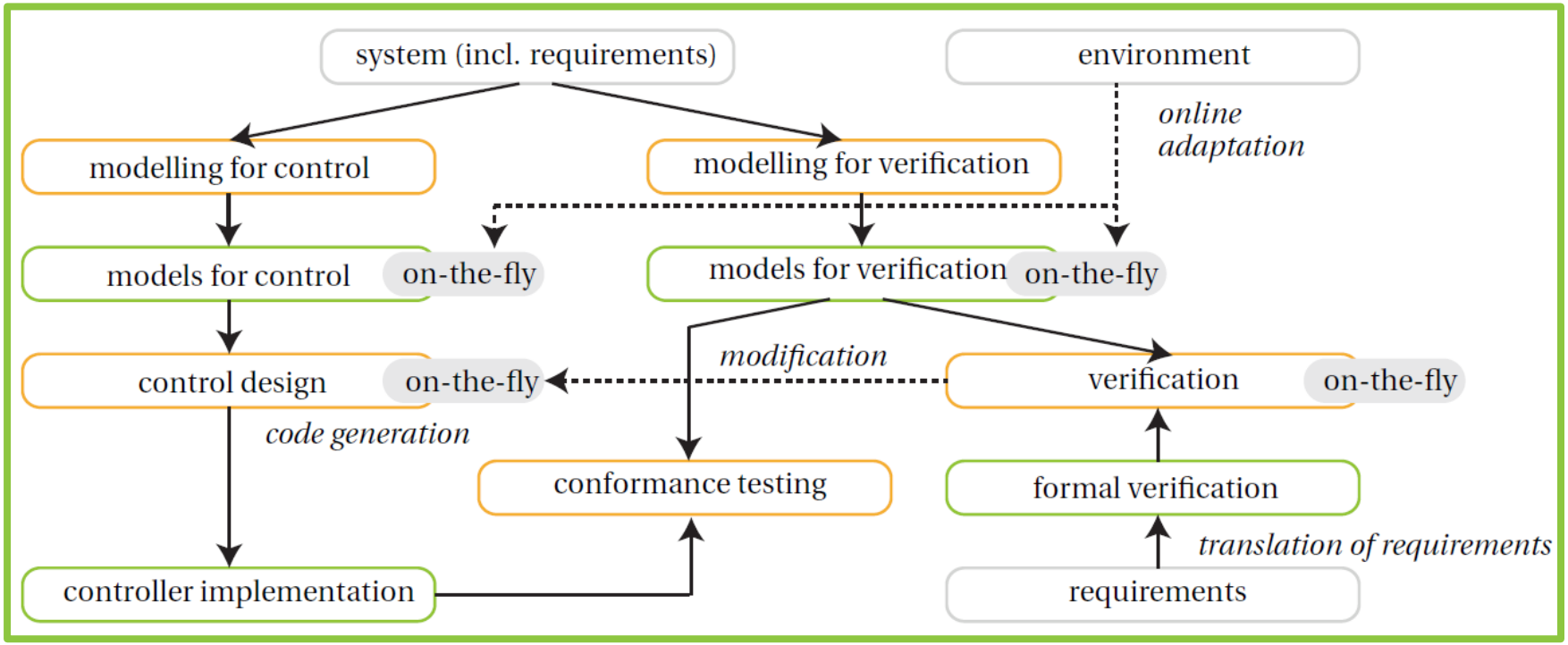
Source: Rethink Robotics

A cross-domain methodology and tool chain

CPS MODELLING

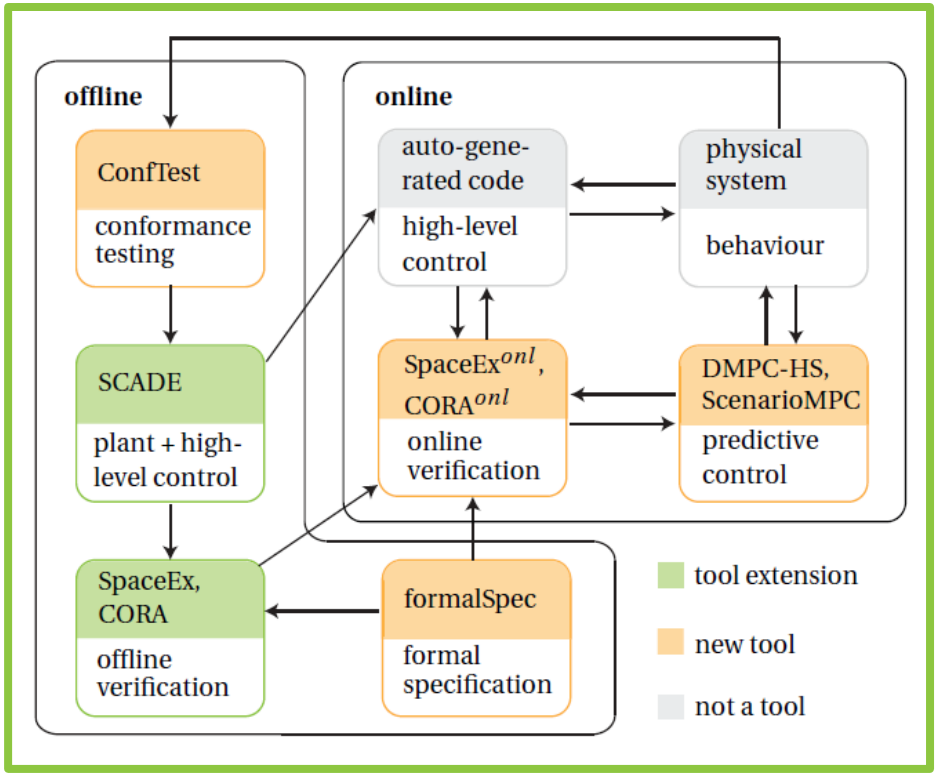


UNCOVERCPS METHODOLOGY



UNCOVERCPS TOOLCHAIN

- Formal specifications
- Modeling of networked CPS
- Reachability analysis
- Control design and implementation
- Conformance testing

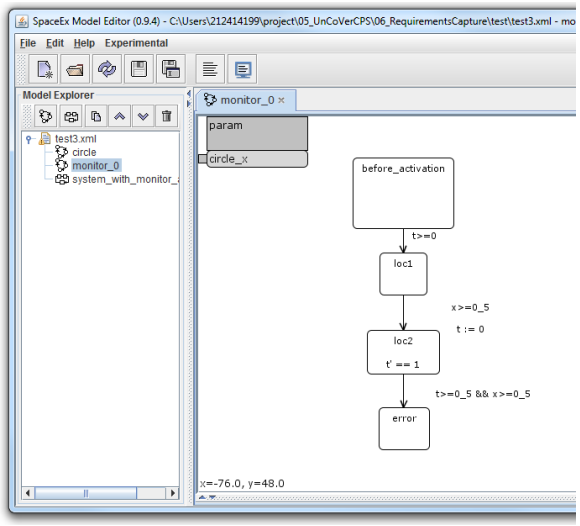


Specifications

SPECIFICATIONS

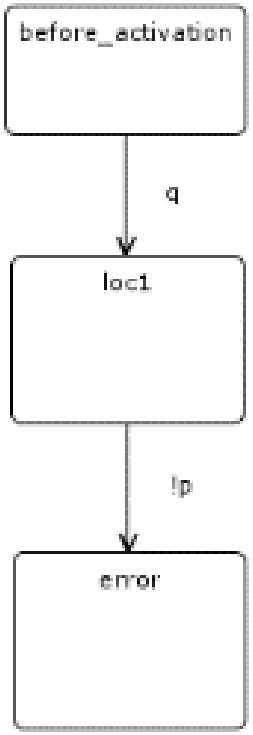
- Problems are:
  - natural language: ambiguous, translation errors
  - specific: learning curve, not cross-stakeholders
- *FormalSpec:* Semi-automatic formalization of system requirements in English
- Methodology: Patterns regularly found in specifications
  - Reduce repeated translation
  - Reduce translation errors
  - Generate formal output

- After t>=0, it is always the case that once x>=0.5 becomes satisfied it holds for less than 0.5 time units.
- Globally, it is always the case that y<ymax holds.
- ...

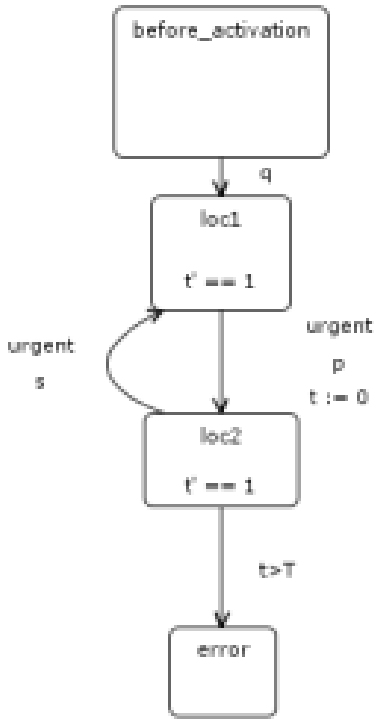


FORMALSPEC: STANDARD MONITOR AUTOMATA

after q, it is always the case that p holds



After q, it is always the case that if p holds, then s holds after at most c time unit(s).



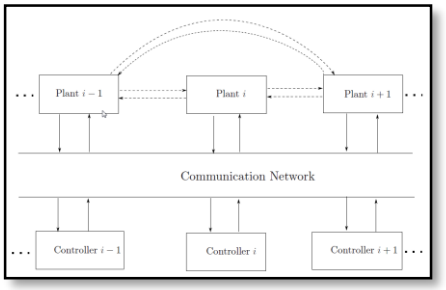
14 Standard hybrid automata represented in graphical and machine readable format.

System Modeling

A GENERAL MODEL FOR CPS

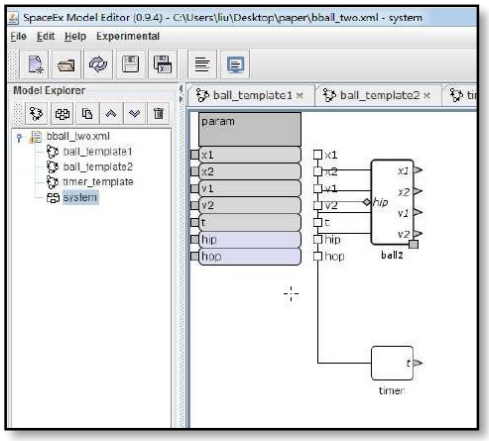
Model Properties:

- modular system structure with coupling through variables and communication
- hybrid system dynamics for the subsystem to account for different situations
- time-variance of constraints to model subsystem interaction
- continuous-time and discrete-time versions



SpaceEx:

- provide a common modeling frame
- ensure that the use cases can be
- connect the model format with the tools for control design
- use a model structure enabling decomposition and use of heterogeneous dynamics



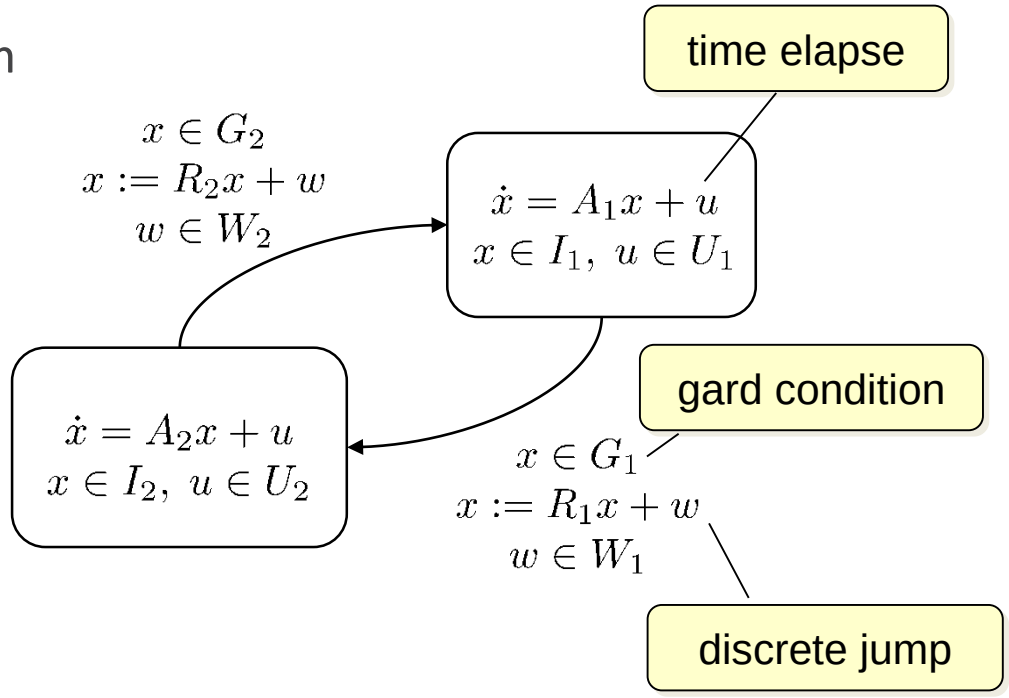
Example of a networked, two ball model in SpaceEx

SX MODELING LANGUAGE

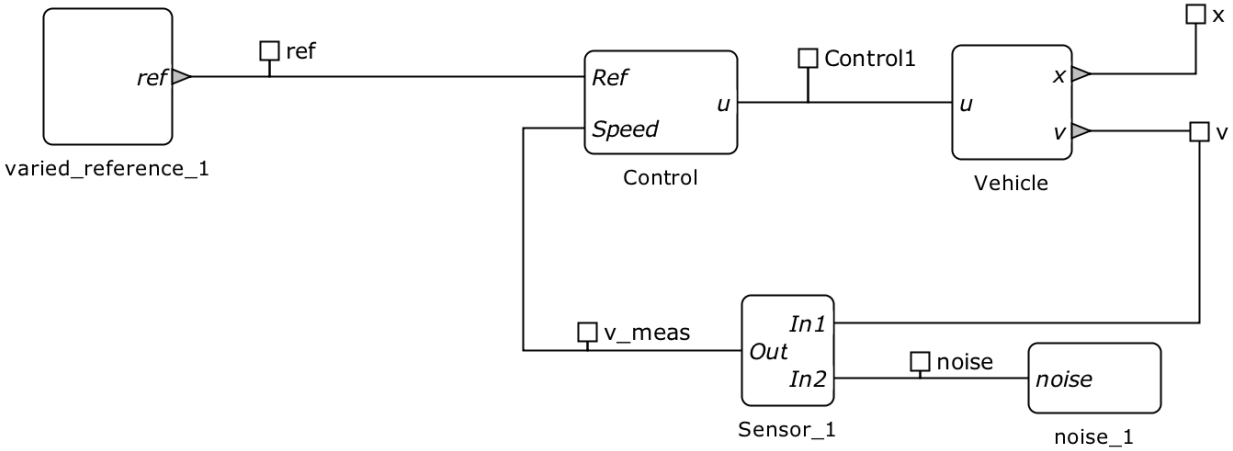
Powerful, near-universal language for continuous and discrete behaviors

Base components: hybrid automata (Alur et al. 1993)

- differential-algebraic equations
- enriched with nondeterminism
  - bounded disturbances
  - noise
  - underspecified behavior
- locations (modes)
- transitions between locations
  - instantaneous
  - deterministic or not



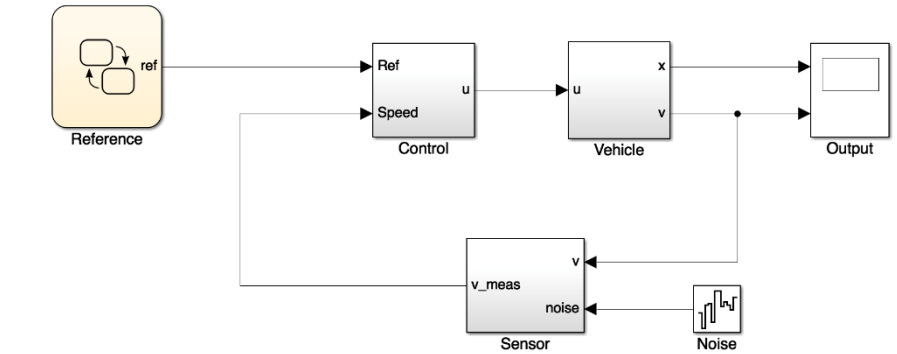
SX MODELING LANGUAGE



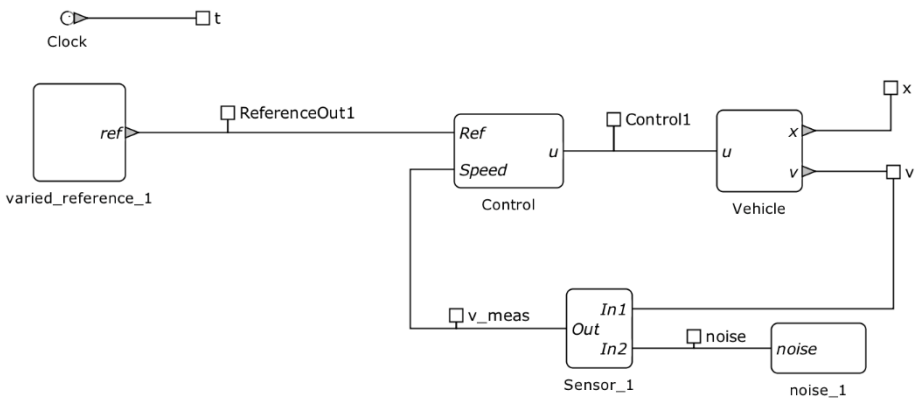
- **Network components**
  - connect base or network components
  - signals = shared variables
  - synchronization = shared events
- graphic representation
  - similar to StateCharts, Matlab Stateflow
- XML

MODEL IMPORT: SIMULINK-STATEFLOW

Translation tool SL2SX for subset of Matlab-Simulink and Stateflow (Minopoli et al., 2016)



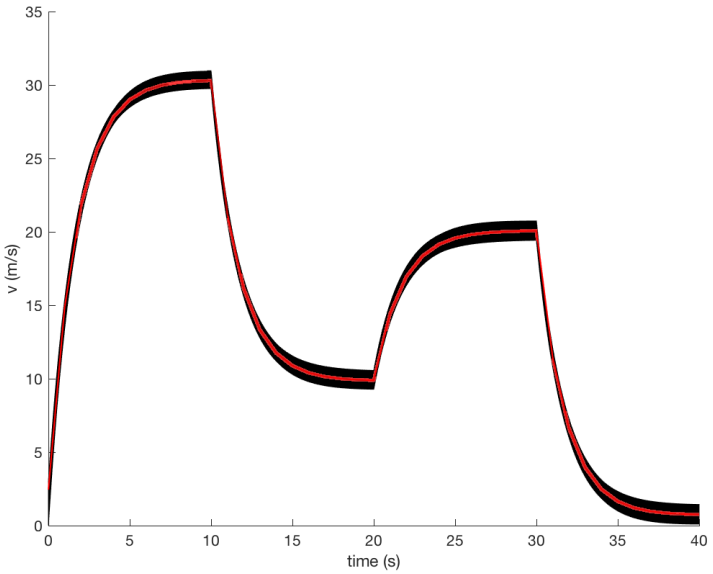
Matlab-Simulink

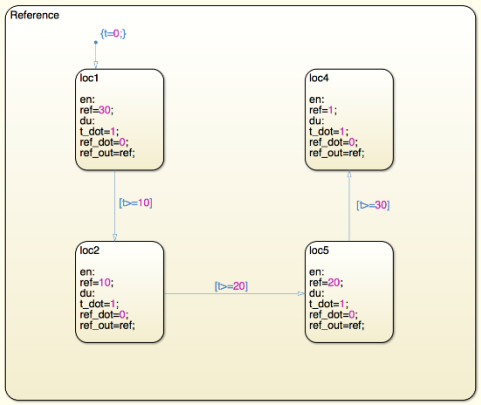


SX

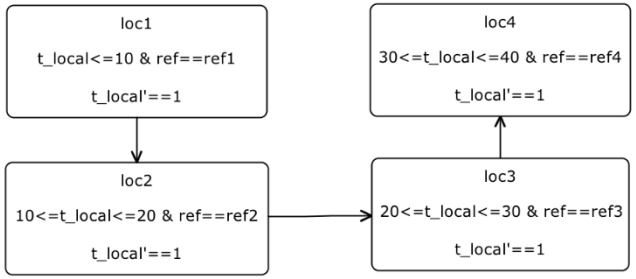
MODEL IMPORT: SIMULINK-STATEFLOW

Translation tool SL2SX for subset of Matlab-Simulink and Stateflow (Minopoli et al., 2016)





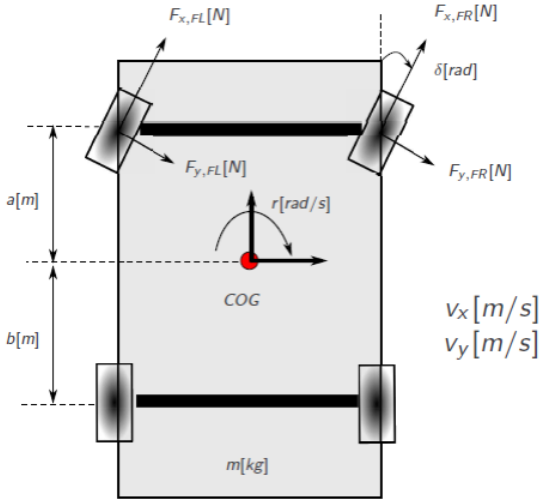
Stateflow



SX

USE CASES

Vehicle modeling for automated driving:

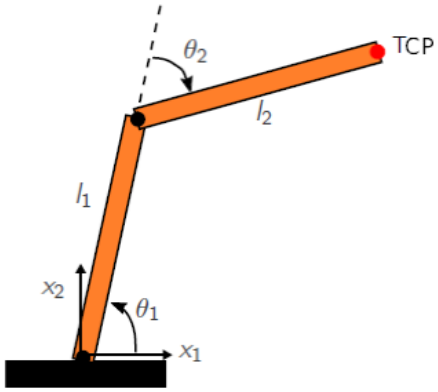


Dynamics in both cases:

- nonlinear dynamics
- different modes
- state / input constraints

automatically possible

Robot modeling for human interaction:



Simplified Models:

- sets of linearized dynamics
- adapt linearizations
- convexification of constraints
- communication of plans

MOBILE ROBOTS IN THE PRESENCE OF HUMANS

- Mobile Robot shall move safely in presence of humans
  - Shared environment
  - Robot has a defined goal location
  - Consider walking humans with unknown goal locations
  - Move safely and efficiently

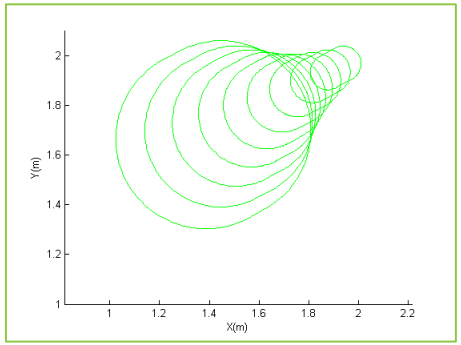
- **Idea**: Verify Passive Safety by Reachability Analysis
  - No collisions when robot is moving
  - Under realistic assumptions of pedestrian behavior

REACHABILITY ANALYSIS

$$\dot{p}_x = v_x$$
$$\dot{p}_y = v_y$$
$$\dot{v}_x = a_x$$
$$\dot{v}_y = a_y$$

$$\mathcal{U}_{ped}^{(a)} = \{(a_x, a_y) \in \mathbb{R} \times \mathbb{R} | a_x^2 + a_y^2 \leq a_{max}^2\}$$

Create Physical Model of Walking Pedestrian

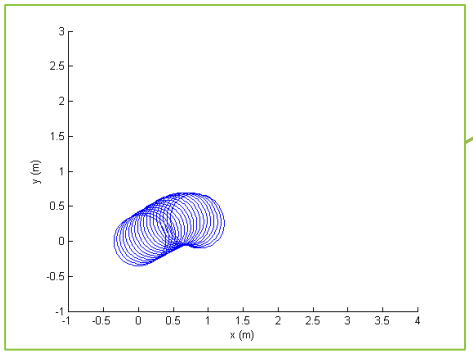


Compute Reachable Sets for each Pedestrian

Check for Possible Collisions

$$\dot{s_1} = u_1 \cos s_3$$
$$\dot{s_2} = u_1 \sin s_3$$
$$\dot{s_3} = u_2$$

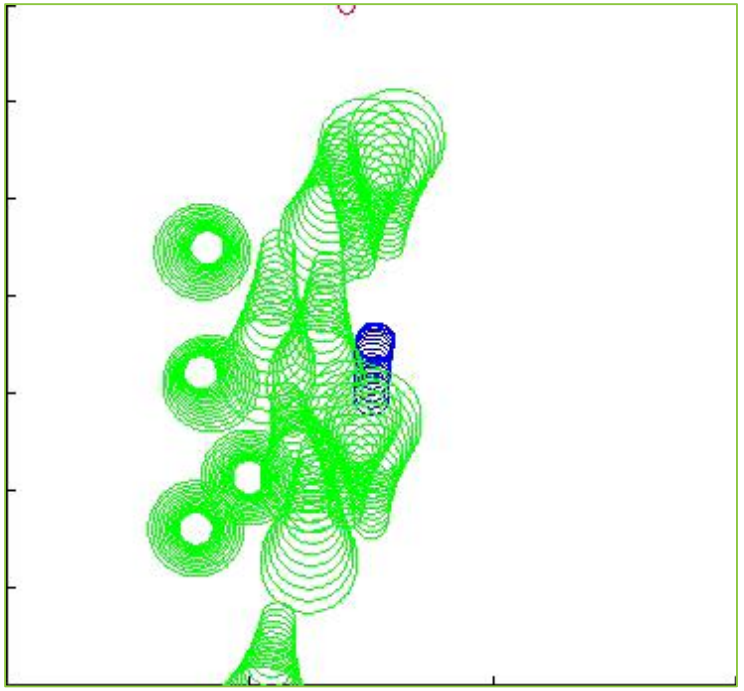
Create Physical Model of Robot



Compute Reachable Sets for Robot

RESULTS

- Implemented using CORA in MATLAB (online)
- **Successfully checked conformance of pedestrian model** against labeled video data
  - Only 16 errors in ~16.000 test cases (resulting from violated assumptions) for unbounded velocity
  - 71 errors for velocity bounded to 2.0 m/s (several running persons)



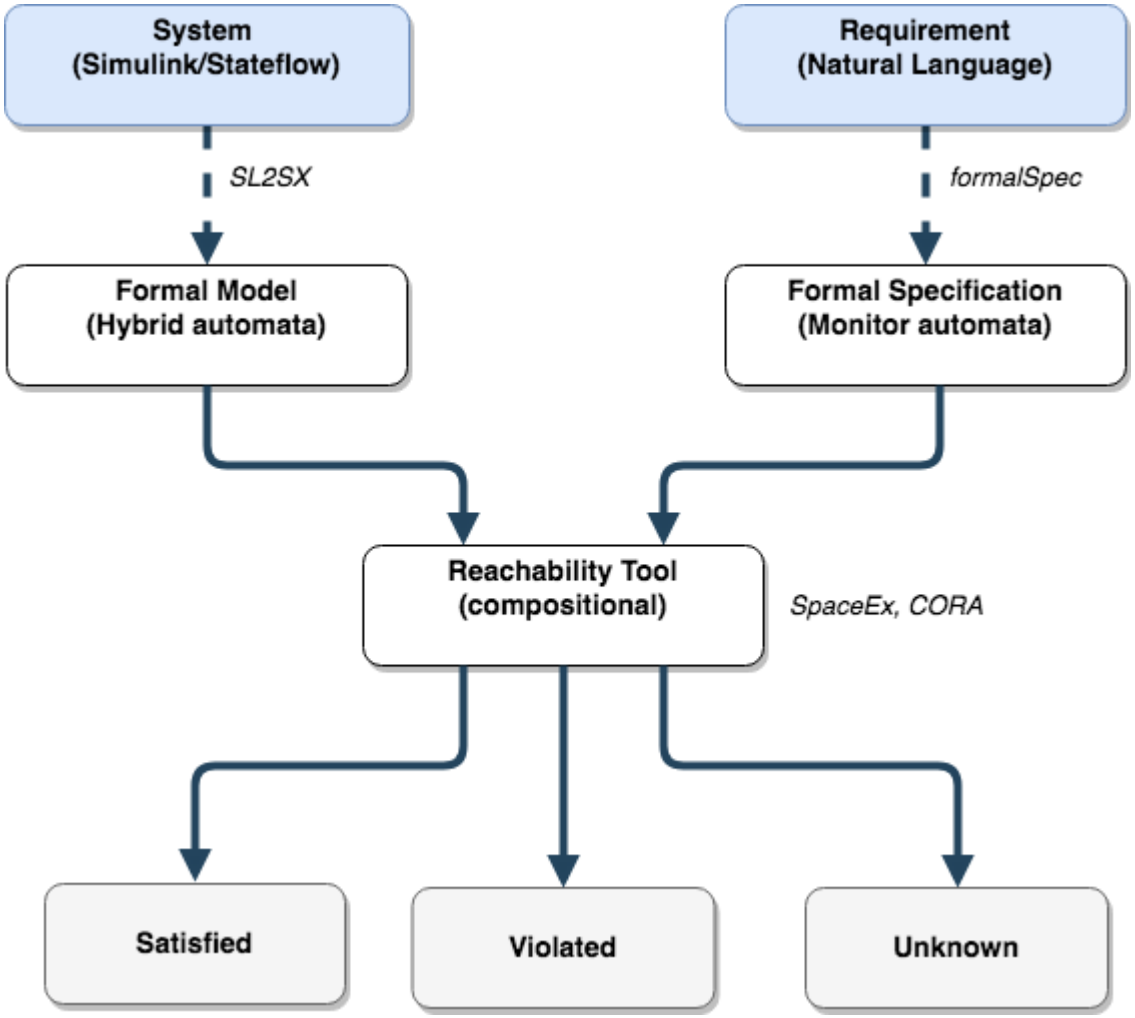
Reachability Analysis - Verification

SPACEEX AND CORA

- **SpaceEx**: Verification for Piecewise Affine Dynamics
  - Univ. Grenoble Alpes (G. Frehse)
  - widely recognized as the most scalable tool (500 variables)
  - mature tool infrastructure (model editor, GUI, virtual machine)
  - model file format is quasi-standard, 3rd party conversion tools, to/from other tools
  - C++ / java

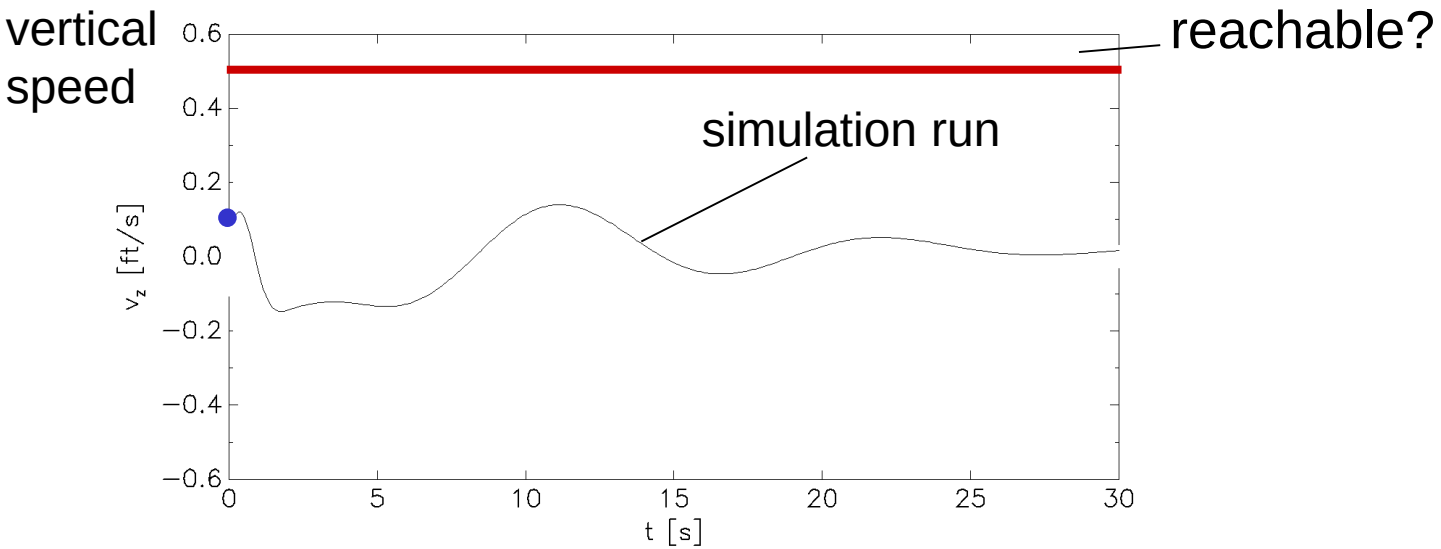
- **CORA**: Verification for Nonlinear Dynamics
  - TU Munich (M. Althoff)
  - tool with most case studies for nonlinear dynamics
  - sophisticated algorithms for uncertain parameters, switches, dynamics etc.
  - based on special geometric objects (zonotopes and extensions)
  - Matlab-toolbox

FORMAL VERIFICATION IN UNCOVERCPS



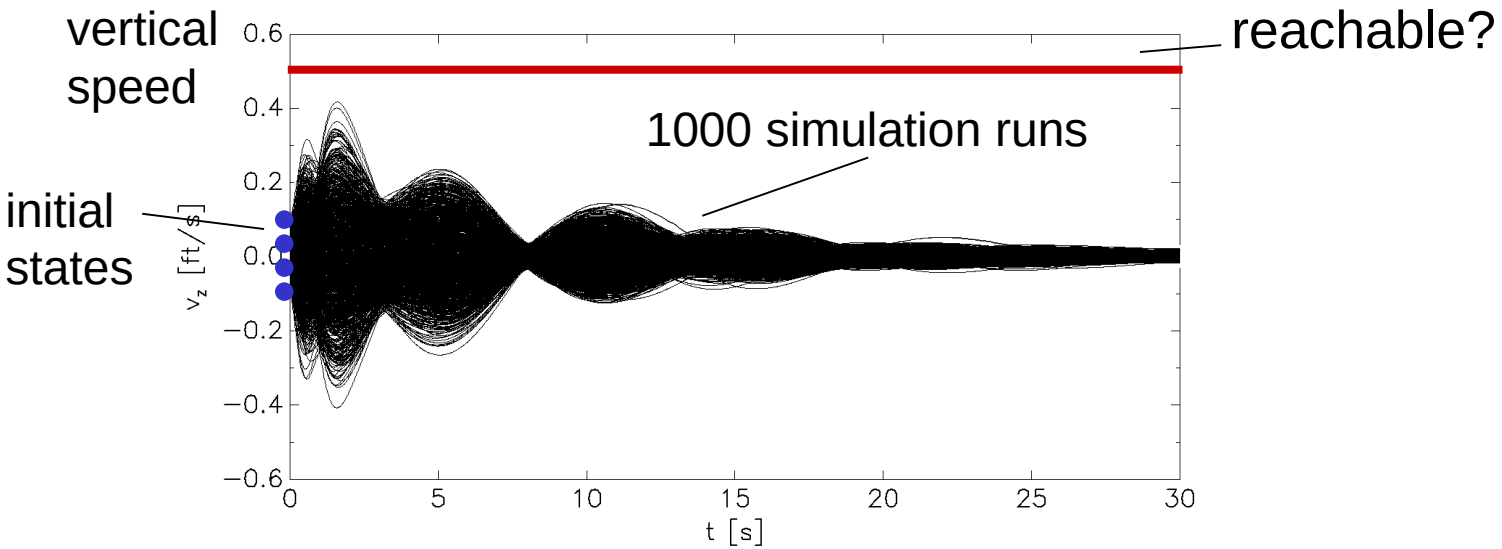
SIMULATION VS REACHABILITY

- **Simulation**
  - **single** behavior



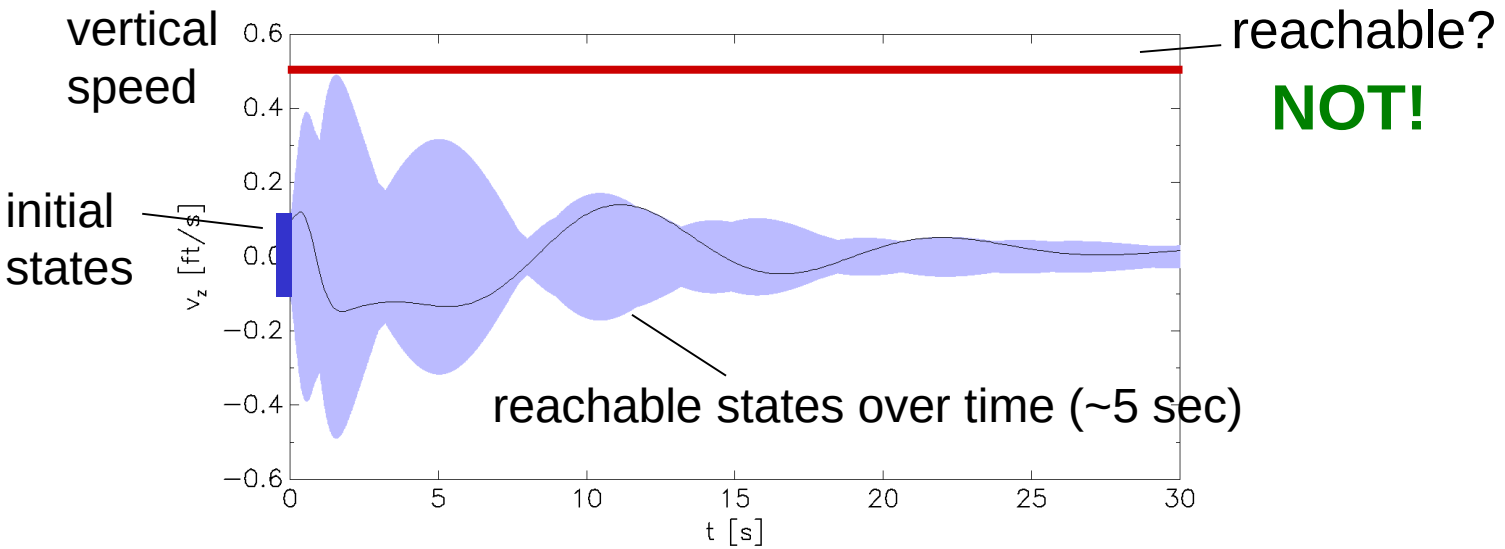
SIMULATION VS REACHABILITY

- **Simulation**
  - **single** behavior



SIMULATION VS REACHABILITY

- **Simulation**
  - **single** behavior
- **Reachability**
  - cover of **all** behaviors



SIMULATION VS REACHABILITY

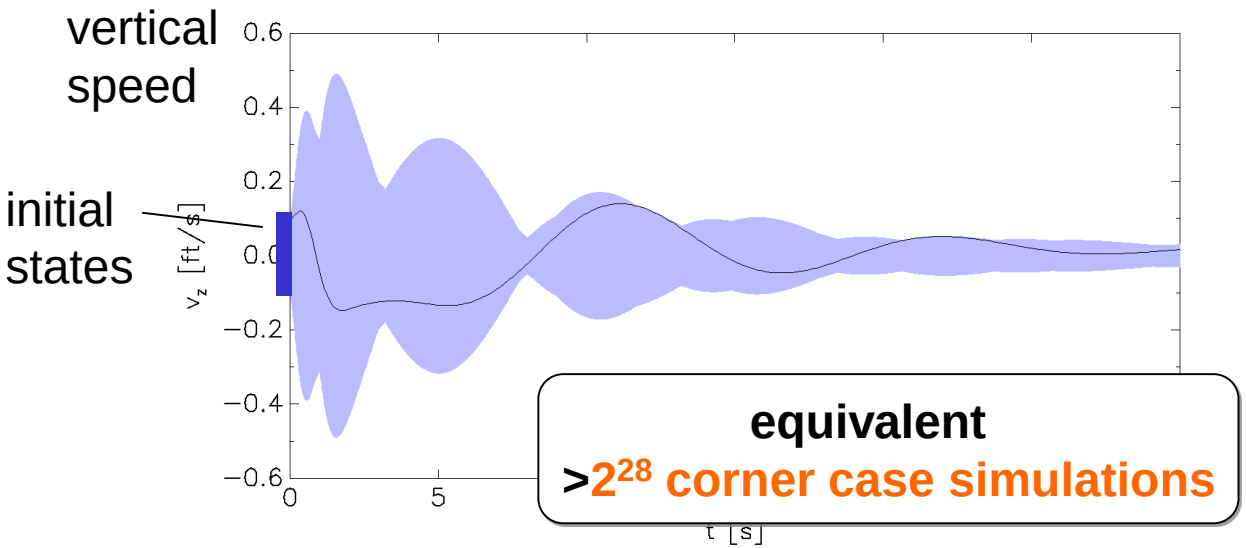
- **Simulation**
  - deterministic

    resolve nondet. using Monte Carlo etc.
  - scalable for nonlinear dyn.

- **Reachability**
  - **nondeterministic**

    continuous disturbances...

    implementation tolerances...
  - scalable for linear dynamics



SIMULATION VS REACHABILITY

- corner case simulation: check all extreme points
  - n variables, T time steps
  - initial set given by intervals = 2^n vertices
  - inputs given by intervals = 2^n vertices

$$2^n (2^n)^T \text{ trajectories}$$

SIMULATION VS REACHABILITY

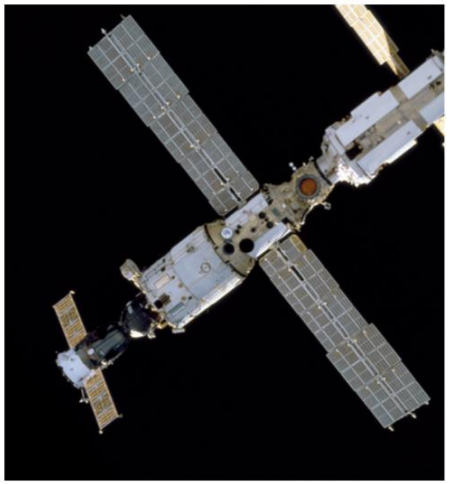
- corner case simulation: check all extreme points
  - n variables, T time steps
  - initial set given by intervals = 2^n vertices
  - inputs given by intervals = 2^n vertices

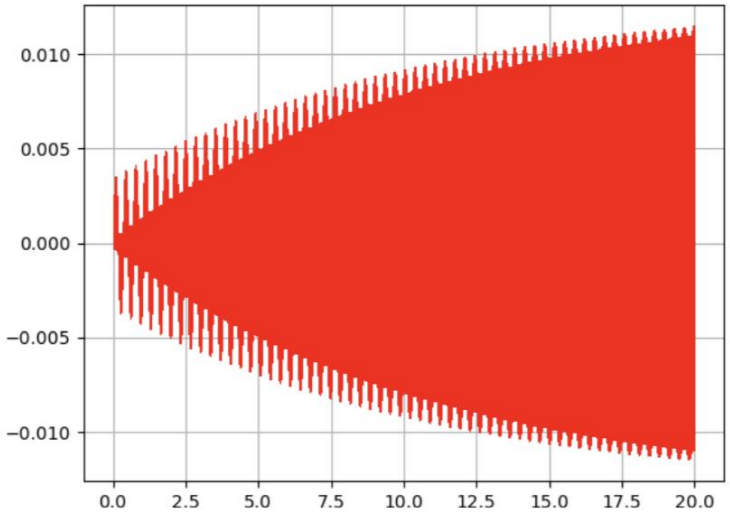
$$\mathbf{2^n\,(2^n)^T}\ \textbf{trajectories}$$

- template reachability (interval enclosure):

$$\mathbf{T}\ O(\mathbf{n^3})\ \textit{operations}$$

EXAMPLE: INTERNATIONAL SPACE STATION





flexible body dynamics of Russian module of ISS[1]

270 variables, 3 nondet. inputs

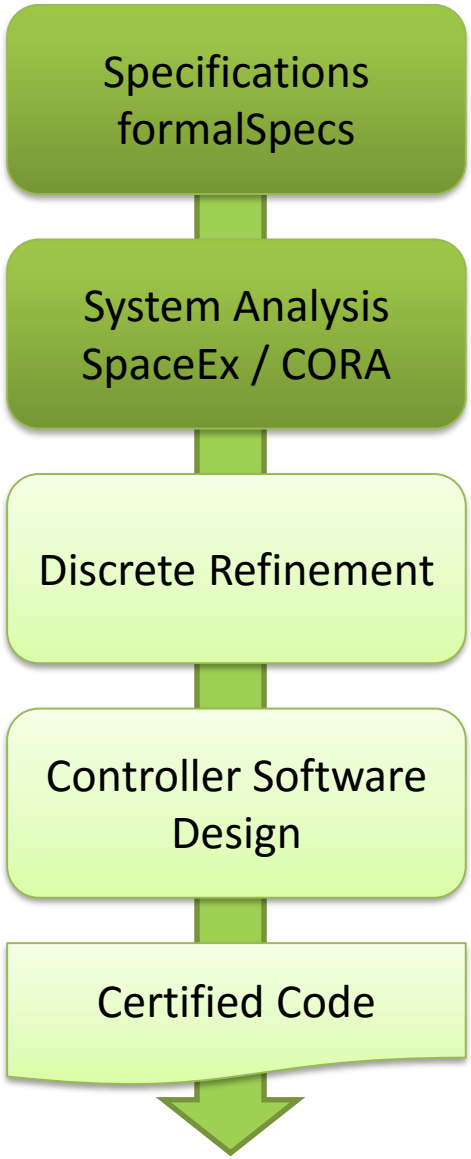
<1h for reach set up to $T = 20$

[1] Y. Chahlaoui and P. Van Dooren, "Benchmark examples for model reduction of linear time-invariant dynamical systems," in *Dimension Reduction of Large-Scale Systems*, Springer, 2005, pp. 379–392.

Implementation

CONTROL DESIGN & IMPLEMENTATION

- The process has the following steps:
  - Specifications
  - Analyses of interactions of an abstract controller with the environment
  - Refinement of the controller and its environment for simulation
  - Final design of the controller and automatic certified code generation



MODEL ABSTRACTIONS FOR CONTROL

Semi-formal specifications

- After t>=0, it is always the case that once x>=0.5 becomes satisfied it holds for less than 0.5 time units.
- Globally, it is always the case that y<ymax holds.
- ...

High-level controller for specification verifications

Hybrid automata

Refined deterministic model with discrete controller and continuous environment

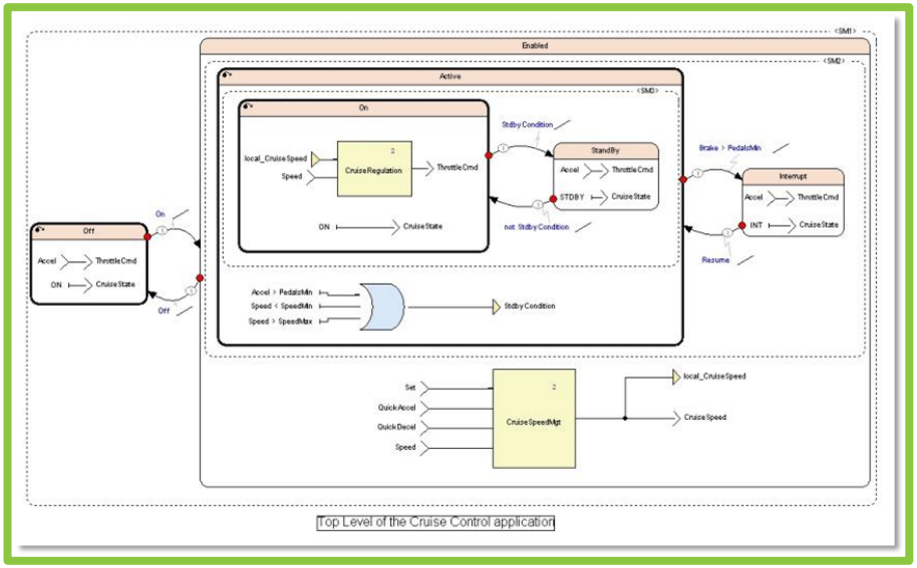
Scade Hybrid

Controller discrete specification for coding

Scade

Code

SCADE HYBRID



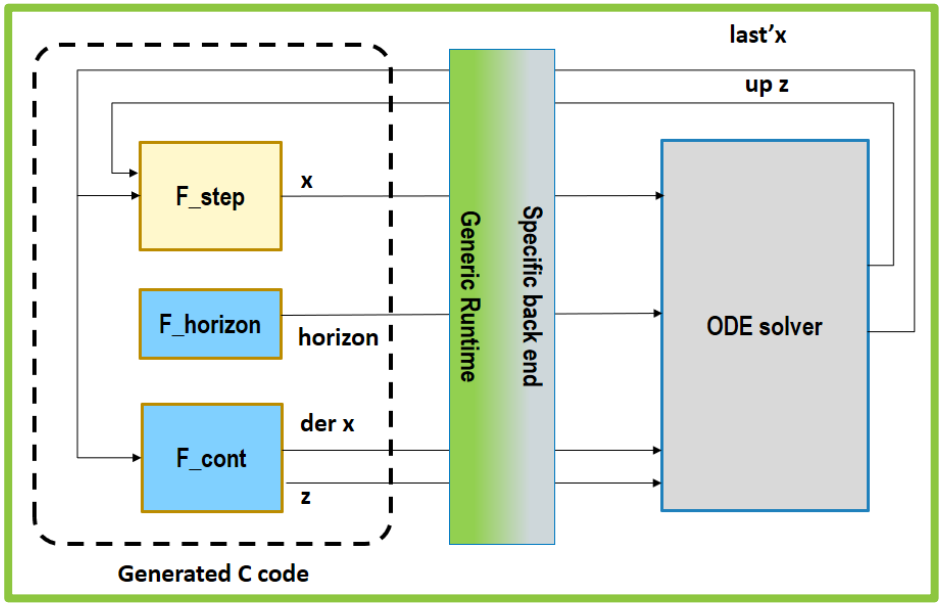
- SCADE tool:
  - an industrial environment dedicated to high-integrity applications development
  - used in various domains: A&D, automotive, rail transport, industry, …
  - Industrial Simulink – Stafeflow import : from control-laws studies to safe implementation
- Scade language describes discrete-time algorithms
  - Control-oriented input formalism
  - **Certified code generation**: ISO 26262, DO-178C, EN 50128, …

SCADE HYBRID

- **Scade Hybrid is an extension of the Scade language**
  - Notion of derivation equation: **der** u = *expr*
  - *Zero-crossing* events
  - Well-defined semantics

- **Closed-loop simulation**
  - Description of physics ;
  - Improving understanding of discrete/continuous interactions;

- **Solver-independent** code generation integrating:
  - Code from discrete controller
  - Code related to continuous behavior



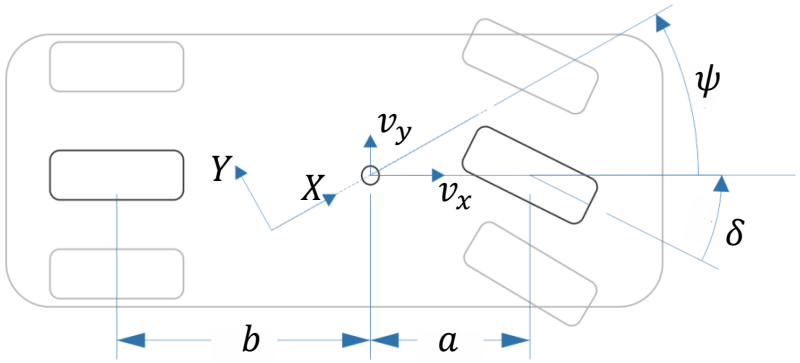
Conclusion

EXAMPLE: VEHICLE MODEL

Vehicle
No collision?

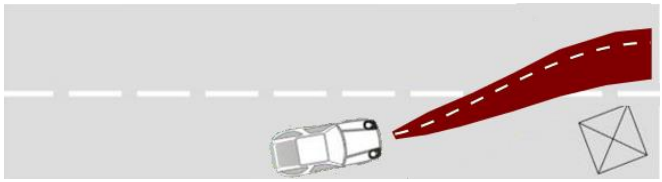
Modeling &
Identification

Conformance Testing



Verification and control model

Frontloading



Formal: No collision

Abstraction and refinement e.g.
trajectory uncertainty due to steering
uncertainty or road conditions

SUMMARY

- The UnCoVerCPS toolchain brings **new capabilities for CPS** design
- Systems can be specified using **semi-automatic formalization** of requirements
- **Conformance Testing** validates the environment models
- **Network of CPS** can be modelized, analyzed and refined
- **Reachability analyses** can be performed to determined proper trajectories
- **Safe controllers** can be derived and implemented using certified code generation.