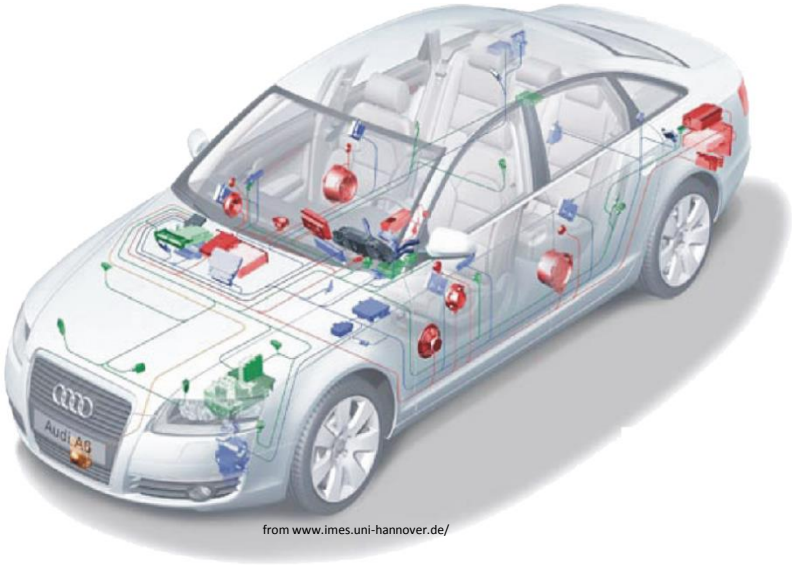
Co-simulation

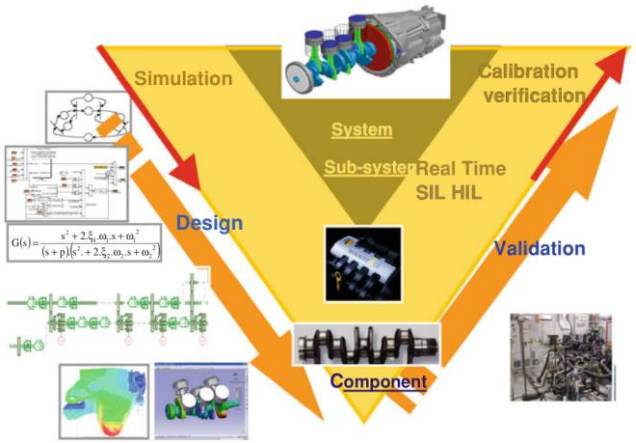
Cláudio Gomes

Complexity meets Markets

- Increasing Complexity
  - Interacting heterogeneous components
- Competitive Market
- Concurrent Development
  - Late Integration Problems
  - Contracts
- Independently Developed Sub-systems
  - Specialized Teams
  - External Suppliers
  - IP Protection
- “Holy Grail”:
  - Integration at **every** stage of development



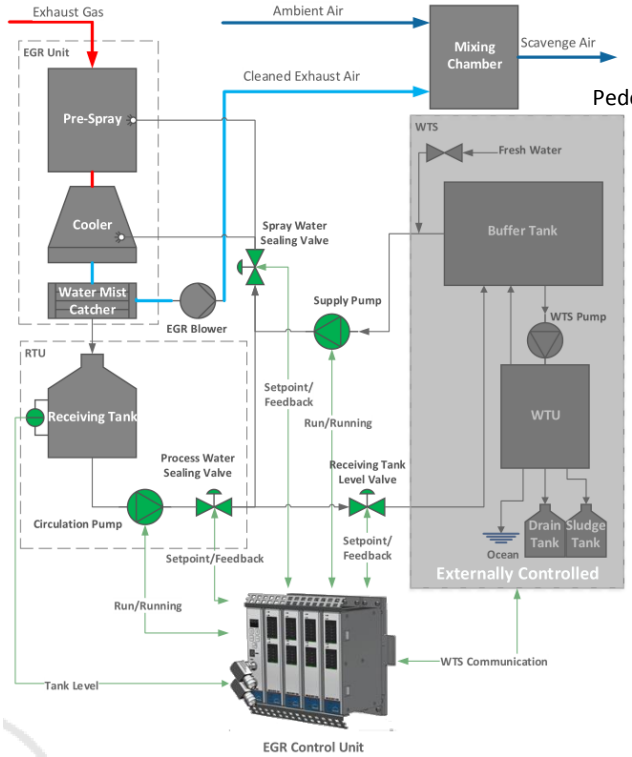
from www.imes.uni-hannover.de/



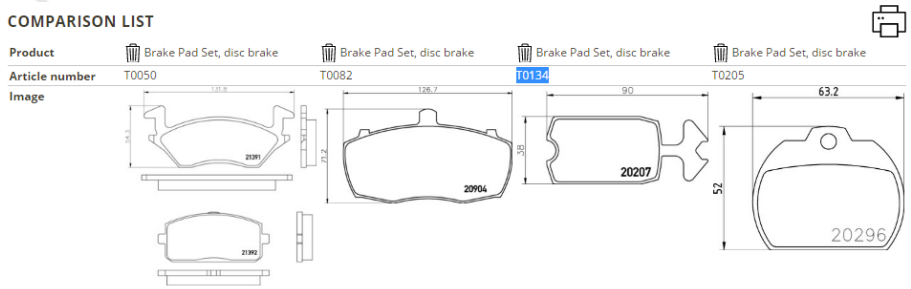
from http://doi.org/10.1007/s00366-012-0286-6

M&S is not Enough

- Domain Specific Tools/Solvers
  - E.g. discrete event solvers, symplectic numerical solvers;
  - <u>Difficult to build one big model</u>
- Suppliers want to protect IP
  - No detailed models provided
  - Catalogs may lack necessary information (e.g., op. temperature range, wear&tear)
  - <u>Hard to get models</u>
- Black box inductive models
  - Lookup tables,
  - Fitted experimental data.
  - <u>There are no models!</u>



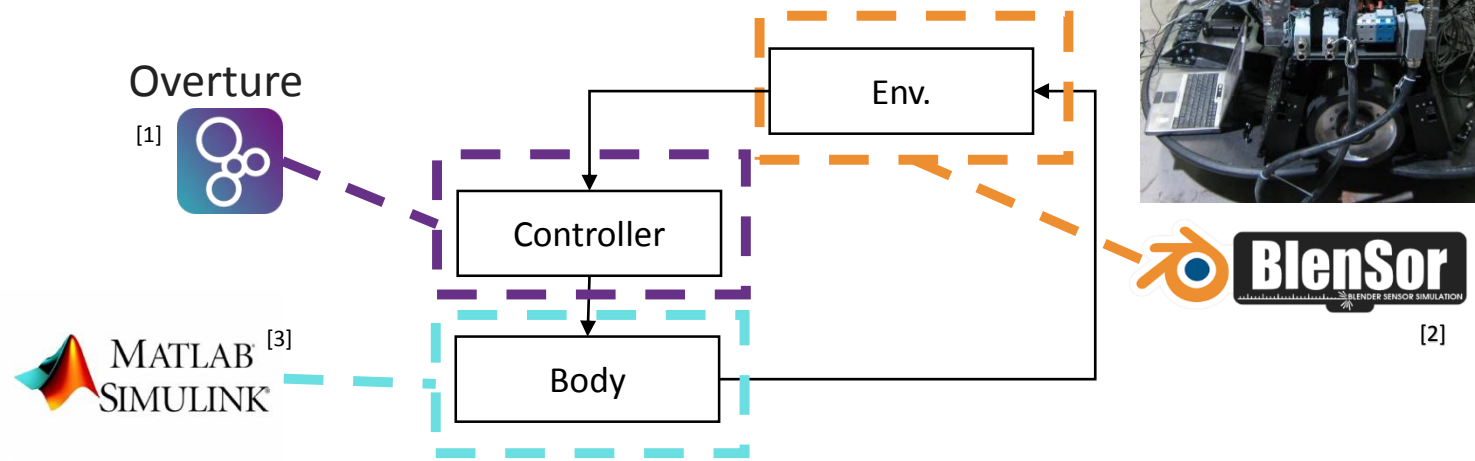
Pedersen, N., Lausdahl, et. al (2017). Distributed Co-Simulation of Embedded Control Software with Exhaust Gas Recirculation Water Handling System using INTO-CPS. In *7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications* (pp. 73–82).

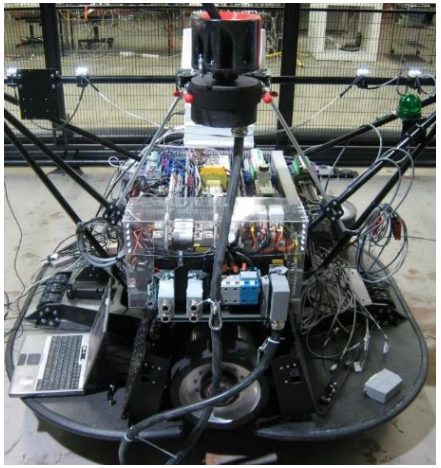


COMPARISON LIST

Product	Brake Pad Set, disc brake	Brake Pad Set, disc brake	Brake Pad Set, disc brake	Brake Pad Set, disc brake
Article number	T0050	T0082	T0134	T0205
Image				
Brake System	Akebono	Lockheed	Citroen	Lucas
Axle	Front Axle	Front Axle	Front Axle	Front Axle
Width [mm]	97.5 - 131.8	126.7	90	63.2
Height [mm]	44 - 54.3	71.2	38	52
usage number	• 21391 • 21392 • 7107D180	• 20904 • 796D61	20207	• 20296 • 7000D71 • 7022D88
Thickness [mm]	13.5	18	13	16.2
Wear Warning Contact	excl. wear warning contact	excl. wear warning contact	excl. wear warning contact	excl. wear warning contact
replacement part	No	No	No	No
Country Version	• Great Britain • Italy	• Great Britain • Italy	• Great Britain • Italy	• Great Britain • Italy

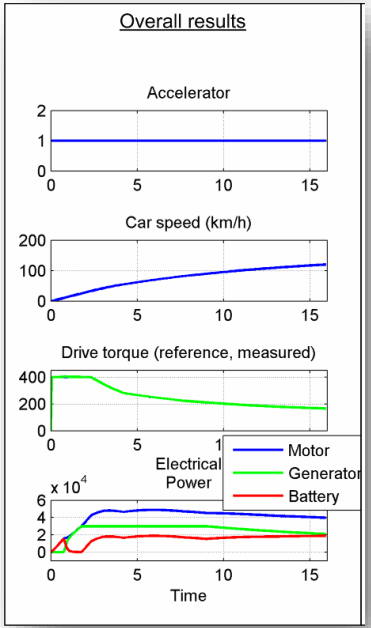
Co-simulation





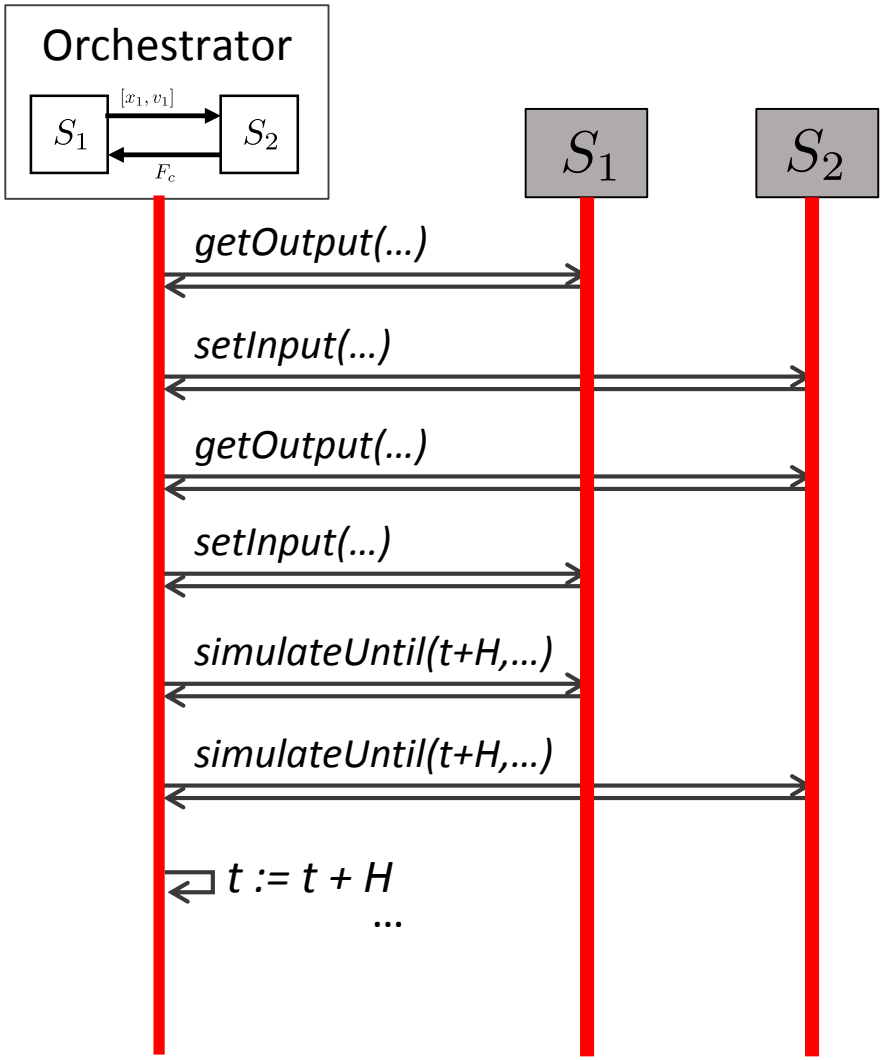
[4]

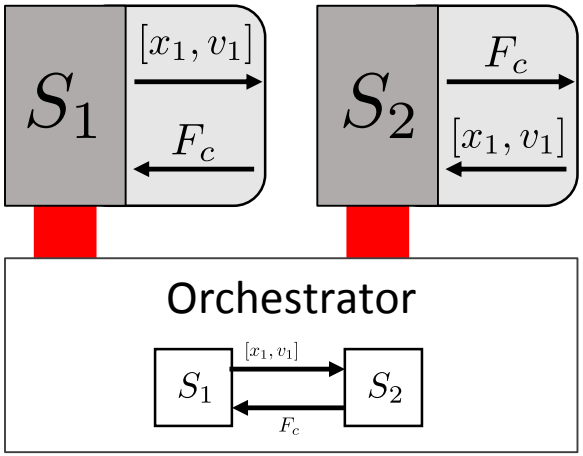
A technique to combine simulators.

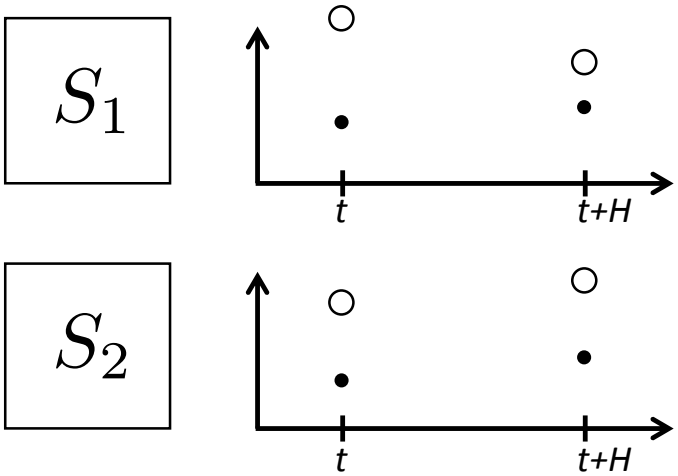


[1] http://overturetool.org/
[2] http://www.blensor.org/
[3] https://nl.mathworks.com/products/simulink.html
[4] Case Study by Kristof Berx , Davy Maes, and Klaas Gadeyne, from Flanders Make

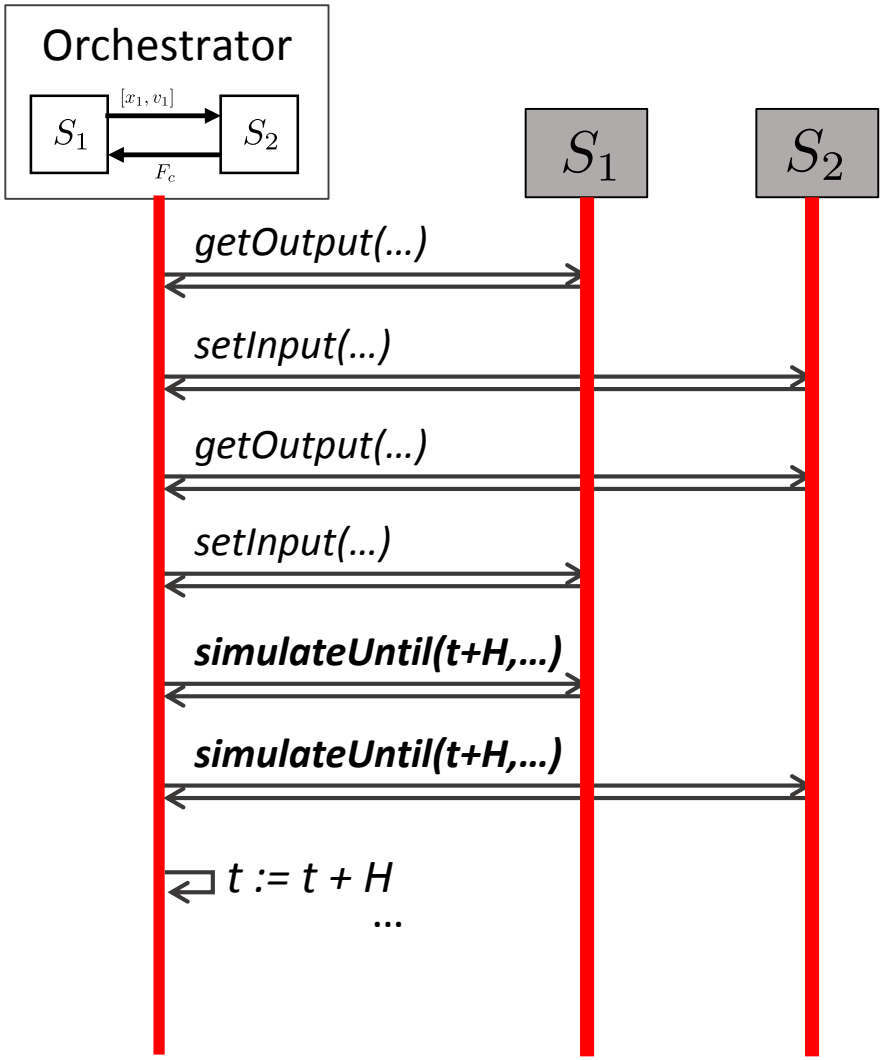
Orchestration

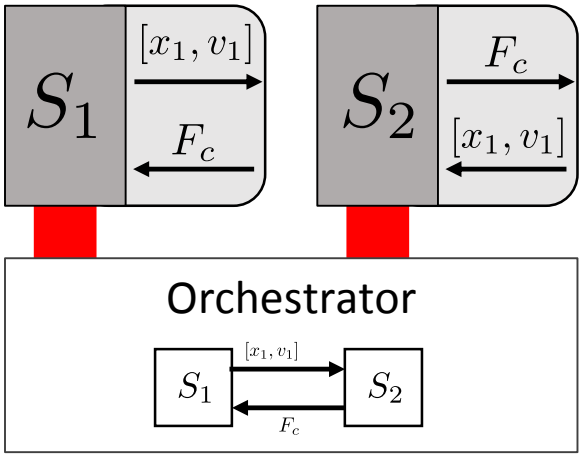


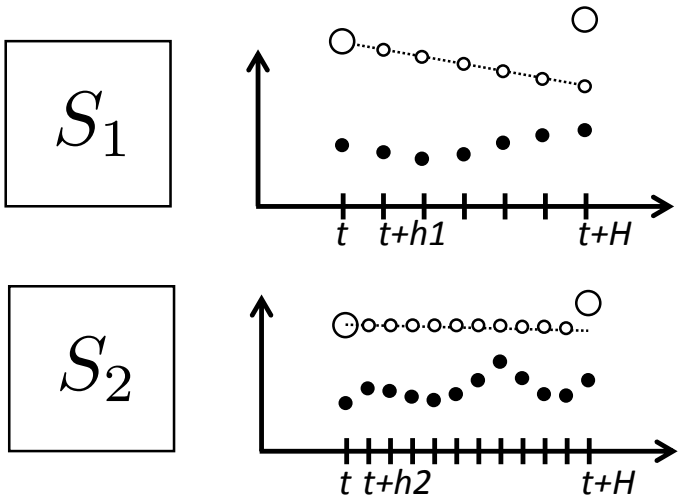




Internal Behavior







The Context

Historical Overview

Casper Thule, David Broman,

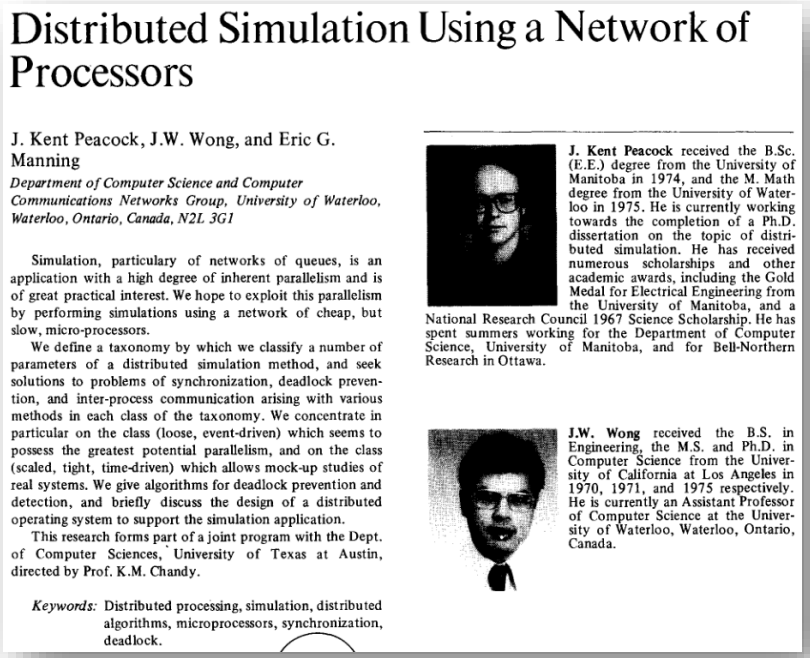
Peter Gorm Larsen, Hans Vangheluwe

Gomes, C. (2017). Keynote on Co-simulation: Open Challenges. Workshop on Formal Co-Simulation of Cyber-Physical Systems. Trento, Italy

Gomes, C., Thule, C., Broman, D., Larsen, P. G., & Vangheluwe, H. (2017). *Co-simulation: State of the art*. Retrieved from http://arxiv.org/abs/1702.00686

The Context

First discrete event synchronization algorithms [1]



Distributed Simulation Using a Network of Processors

J. Kent Peacock, J.W. Wong, and Eric G. Manning

Department of Computer Science and Computer Communications Networks Group, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1

Simulation, particulary of networks of queues, is an application with a high degree of inherent parallelism and is of great practical interest. We hope to exploit this parallelism by performing simulations using a network of cheap, but slow, micro-processors.

We define a taxonomy by which we classify a number of parameters of a distributed simulation method, and seek solutions to problems of synchronization, deadlock prevention, and inter-process communication arising with various methods in each class of the taxonomy. We concentrate in particular on the class (loose, event-driven) which seems to possess the greatest potential parallelism, and on the class (scaled, tight, time-driven) which allows mock-up studies of real systems. We give algorithms for deadlock prevention and detection, and briefly discuss the design of a distributed operating system to support the simulation application.

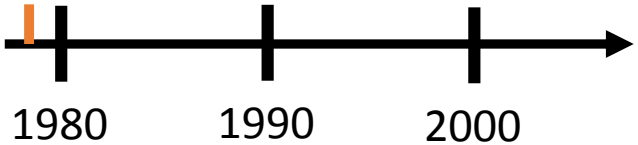
This research forms part of a joint program with the Dept. of Computer Sciences, University of Texas at Austin, directed by Prof. K.M. Chandy.

Keywords: Distributed processing, simulation, distributed algorithms, microprocessors, synchronization, deadlock.

J. Kent Peacock received the B.Sc. (E.E.) degree from the University of Manitoba in 1974, and the M. Math degree from the University of Waterloo in 1975. He is currently working towards the completion of a Ph.D. dissertation on the topic of distributed simulation. He has received numerous scholarships and other academic awards, including the Gold Medal for Electrical Engineering from the University of Manitoba, and a National Research Council 1967 Science Scholarship. He has spent summers working for the Department of Computer Science, University of Manitoba, and for Bell-Northern Research in Ottawa.

J.W. Wong received the B.S. in Engineering, the M.S. and Ph.D. in Computer Science from the University of California at Los Angeles in 1970, 1971, and 1975 respectively. He is currently an Assistant Professor of Computer Science at the University of Waterloo, Waterloo, Ontario, Canada.

[2]

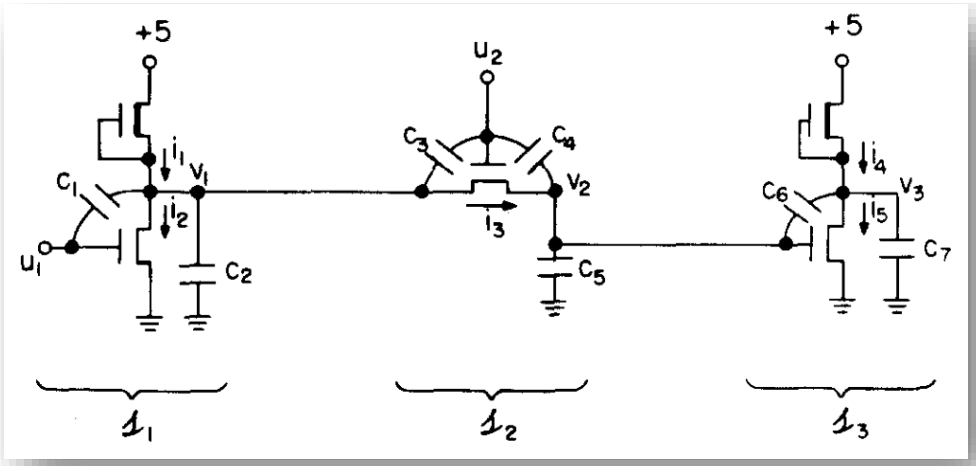


[1] Fujimoto, R. M. (1990). Parallel discrete event simulation. *Communications of the ACM*, *33*(10), 30–53. http://doi.org/10.1145/84537.84545

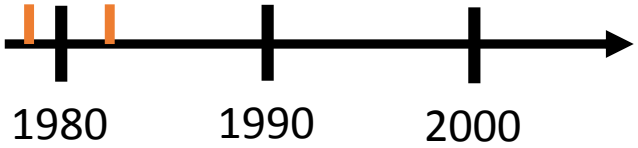
[2] Kent Peacock, J., Wong, J. ., & Manning, E. G. (1979). Distributed simulation using a network of processors. *Computer Networks (1976)*, *3*(1), 44–56. http://doi.org/10.1016/0376-5075(79)90053-9

The Context

- First discrete event synchronization algorithms
  - <u>Decomposition</u> of sparsely coupled systems.[1]



[2]

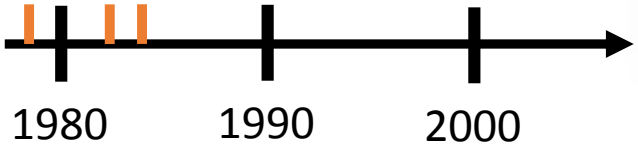


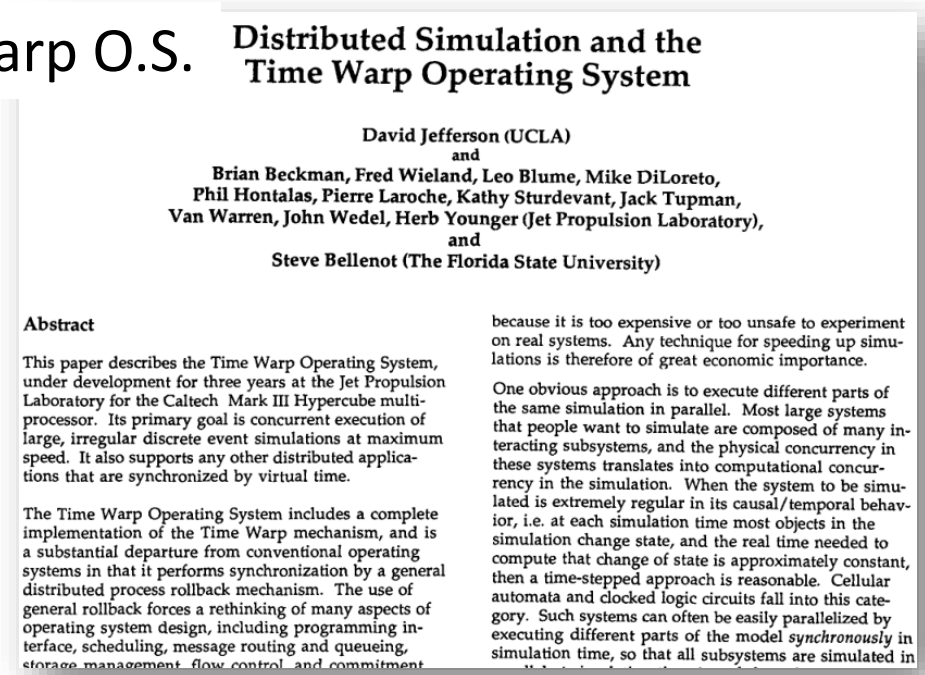
[1] http://www.frobnotz.com/cosim/cosim.html

[2] Newton, A. R., & Sangiovanni-Vincentelli, A. L. (1983). Relaxation-Based Electrical Simulation. *SIAM Journal on Scientific and Statistical Computing*, *4*(3), 485–524. http://doi.org/10.1137/0904036

The Context

- First discrete event synchronization algorithms
  - Decomposition of sparsely coupled systems.
    - <u>Optimistic</u> Time warp O.S.





Distributed Simulation and the Time Warp Operating System

David Jefferson (UCLA)
and
Brian Beckman, Fred Wieland, Leo Blume, Mike DiLoreto,
Phil Hontalas, Pierre Laroche, Kathy Sturdevant, Jack Tupman,
Van Warren, John Wedel, Herb Younger (Jet Propulsion Laboratory),
and
Steve Bellenot (The Florida State University)

Abstract

This paper describes the Time Warp Operating System, under development for three years at the Jet Propulsion Laboratory for the Caltech Mark III Hypercube multiprocessor. Its primary goal is concurrent execution of large, irregular discrete event simulations at maximum speed. It also supports any other distributed applications that are synchronized by virtual time.

The Time Warp Operating System includes a complete implementation of the Time Warp mechanism, and is a substantial departure from conventional operating systems in that it performs synchronization by a general distributed process rollback mechanism. The use of general rollback forces a rethinking of many aspects of operating system design, including programming interface, scheduling, message routing and queueing, storage management, flow control, and commitment

because it is too expensive or too unsafe to experiment on real systems. Any technique for speeding up simulations is therefore of great economic importance.

One obvious approach is to execute different parts of the same simulation in parallel. Most large systems that people want to simulate are composed of many interacting subsystems, and the physical concurrency in these systems translates into computational concurrency in the simulation. When the system to be simulated is extremely regular in its causal/temporal behavior, i.e. at each simulation time most objects in the simulation change state, and the real time needed to compute that change of state is approximately constant, then a time-stepped approach is reasonable. Cellular automata and clocked logic circuits fall into this category. Such systems can often be easily parallelized by executing different parts of the model *synchronously* in simulation time, so that all subsystems are simulated in

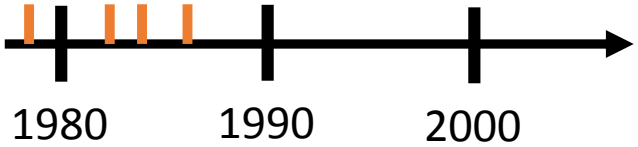
[1]

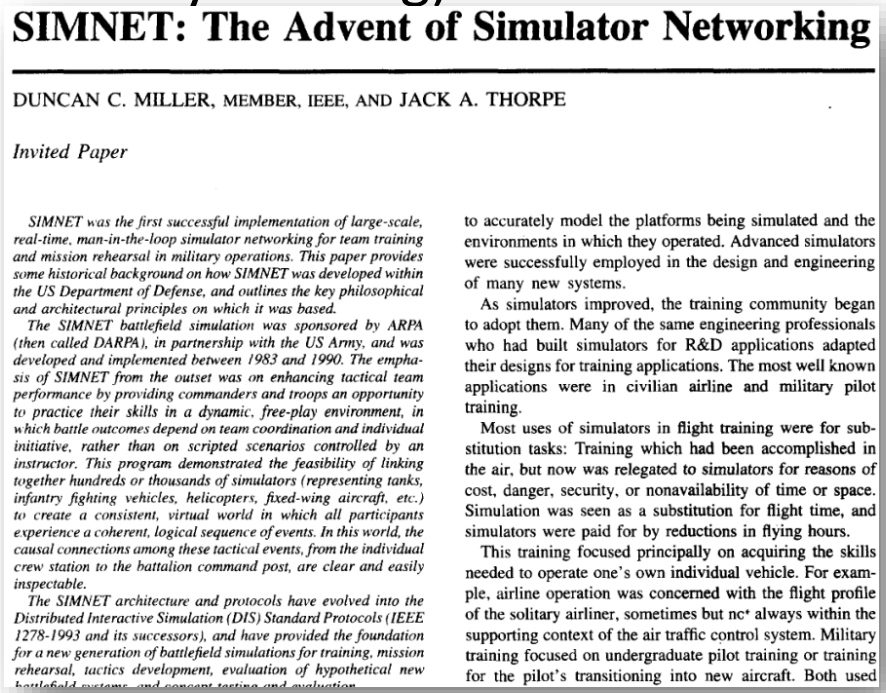
[1] Jefferson, D., Beckman, B., Wieland, F., Blume, L., & Diloreto, M. (1987). Distributed Simulation and the Time Warp Operating System David. *ACM SIGOPS Operating Systems Review*, *21*(5), 77–93. http://doi.org/10.1145/37499.37508

The Context

- First discrete event synchronization algorithms
  - Decomposition of sparsely coupled systems.
    - Optimistic Time warp O.S.
      - SIMNET(Real-time Military Training)

[1] Miller, D. C., & Thorpe, J. A. (1995). SIMNET: the advent of simulator networking. *Proceedings of the IEEE*, *83*(8), 1114–1123. http://doi.org/10.1109/5.400452





SIMNET: The Advent of Simulator Networking

DUNCAN C. MILLER, MEMBER, IEEE, AND JACK A. THORPE

Invited Paper

SIMNET was the first successful implementation of large-scale, real-time, man-in-the-loop simulator networking for team training and mission rehearsal in military operations. This paper provides some historical background on how SIMNET was developed within the US Department of Defense, and outlines the key philosophical and architectural principles on which it was based.

The SIMNET battlefield simulation was sponsored by ARPA (then called DARPA), in partnership with the US Army, and was developed and implemented between 1983 and 1990. The emphasis of SIMNET from the outset was on enhancing tactical team performance by providing commanders and troops an opportunity to practice their skills in a dynamic, free-play environment, in which battle outcomes depend on team coordination and individual initiative, rather than on scripted scenarios controlled by an instructor. This program demonstrated the feasibility of linking together hundreds or thousands of simulators (representing tanks, infantry fighting vehicles, helicopters, fixed-wing aircraft, etc.) to create a consistent, virtual world in which all participants experience a coherent, logical sequence of events. In this world, the causal connections among these tactical events, from the individual crew station to the battalion command post, are clear and easily inspectable.

The SIMNET architecture and protocols have evolved into the Distributed Interactive Simulation (DIS) Standard Protocols (IEEE 1278-1993 and its successors), and have provided the foundation for a new generation of battlefield simulations for training, mission rehearsal, tactics development, evaluation of hypothetical new

to accurately model the platforms being simulated and the environments in which they operated. Advanced simulators were successfully employed in the design and engineering of many new systems.

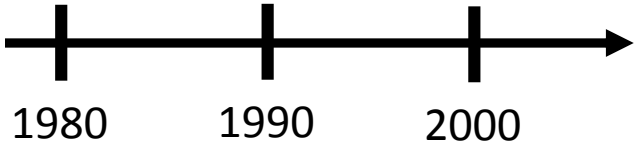
As simulators improved, the training community began to adopt them. Many of the same engineering professionals who had built simulators for R&D applications adapted their designs for training applications. The most well known applications were in civilian airline and military pilot training.

Most uses of simulators in flight training were for substitution tasks: Training which had been accomplished in the air, but now was relegated to simulators for reasons of cost, danger, security, or nonavailability of time or space. Simulation was seen as a substitution for flight time, and simulators were paid for by reductions in flying hours.

This training focused principally on acquiring the skills needed to operate one's own individual vehicle. For example, airline operation was concerned with the flight profile of the solitary airliner, sometimes but nc+ always within the supporting context of the air traffic control system. Military training focused on undergraduate pilot training or training for the pilot's transitioning into new aircraft. Both used

[1]

The Context



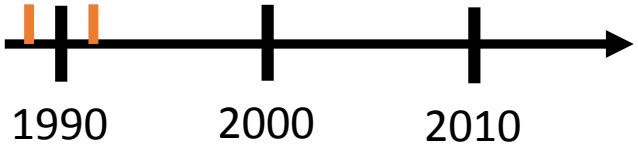
The Context

SIMNET(Real-time Military Training)

SW/HW co-simulation.

> "Software developers were often left to develop code for months with severely limited ability to test [...].
> Painful integration efforts came late in the design cycle, and minor miscommunication became major design flaws."
>
> -- Anne Powell and Shawn Lin

[1]



[1] http://www.frobnotz.com/cosim/cosim.html

The Context

[1] Otter, M., & Elmqvist, H. (1995). The DSblock model interface for exchanging model components. In *Proceedings of the Eurosim'95 Simulation Congress* (pp. 505–510).

SIMNET(Military Training)

SW/HW co-simulation.

DSBlock Standard

[1]

```
  parameters |      | initial
             |      | conditions
             v      v
           I---------I
U(T) ----->I DSblock I-----> Y(T)
           I---------I
              ^    .
              |    .
              |    .........> W(T), X(T), Z(T)
              |
              | data from file or database
```

DSblock: A neutral description of dynamic systems

Version 3.3

Otter

en, April 1994

cs and System Dynamics
ffenhofen
essling
er@dlr.de

n (CACSD) packages should be OPEN for
s well as export. In the following a
proposed to define parameterized,

1990 2000 2010

"A DSblock may serve as neutral 'model bus' to exchange nonlinear models between different modelling and simulation environments."

The Context

- SIMNET(Military Training)
  - SW/HW co-simulation.
    - DSBlock Standard
      - MPM&S Research agenda

[1] Vangheluwe, H. L., Vansteenkiste, G. C., & Kerckhoffs, E. J. H. (1996). Simulation for the Future : Progress of the Esprit Basic Research Working Group 8467. In *Proceedings of the 1996 European Simulation Symposium* (pp. XXIX--XXXIV). Genoa: Society for Computer Simulation International.

[1]

Simulation for the Future: Progress of the Esprit Basic Research Working Group 8467

Hans L. Vangheluwe, Ghislain C. Vansteenkiste
Applied Mathematics, Biometrics and Process Control
University of Gent
Coupure 653
B-9000 Gent
Belgium

Eugène J.H. Kerckhoffs *
Applied Mathematics and Informatics
Delft University of Technology
Julianalaan 132
NL-2628 BL Delft
The Netherlands

tel: +32 (9) 264 59 37
fax: +32 (9) 223 49 41

Abstract

The Esprit Basic Research Working Grou
lation for the Future: new concepts, tools and applications",

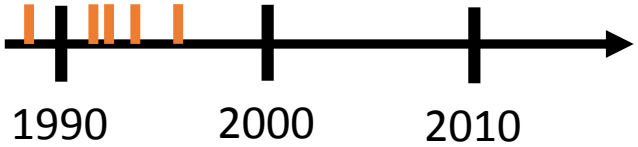
than currently used and exploit this as a basis

"A communication protocol which defines interaction between simulators in a tool and vendor independent manner."

1990

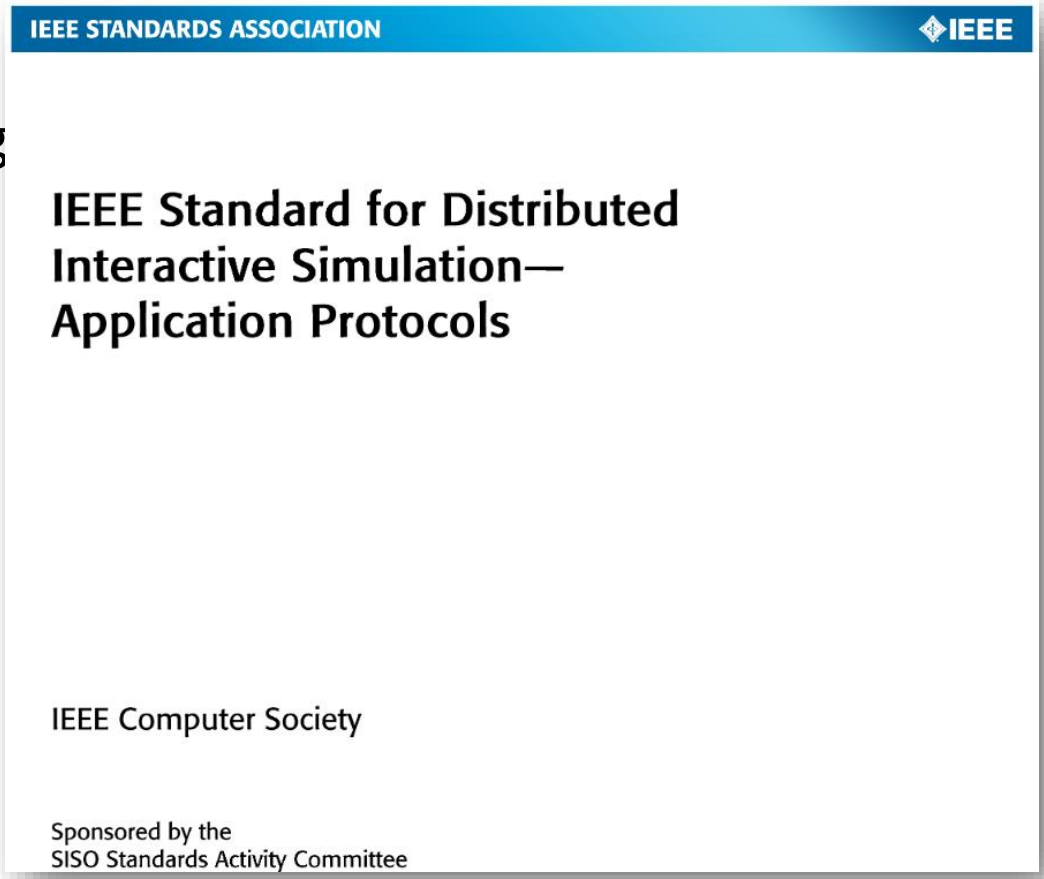
The Context

- SIMNET(Military Training)
  - SW/HW co-simulation.
  - DSBlock Standard
  - MPM&S Research ag
  - DIS IEEE Standard



[1] IEEE Standard for Distributed Interactive Simulation--Application Protocols. (2012). *IEEE Std 1278.1-2012 (Revision of IEEE Std 1278.1-1995)*. http://doi.org/10.1109/IEEESTD.2012.6387564

[1]


IEEE STANDARDS ASSOCIATION IEEE

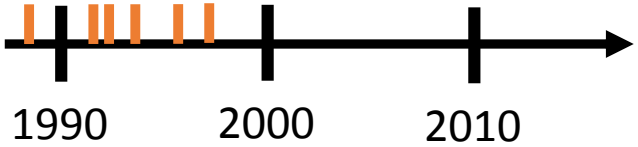
IEEE Standard for Distributed Interactive Simulation—Application Protocols

IEEE Computer Society

Sponsored by the
SISO Standards Activity Committee

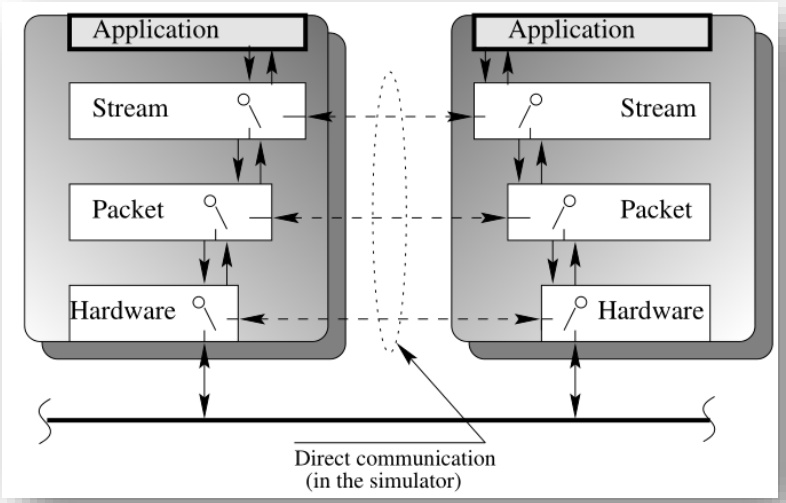
The Context

- SIMNET(Military Training)
  - SW/HW co-simulation.
    - DSBlock Standard
      - MPM&S Research agenda
        - DIS IEEE Standard
          - <u>Multi-abstraction</u>



[1] Hines, K., & Borriello, G. (1997). Selective focus as a means of improving geographically distributed embedded system co-simulation. In *8th IEEE International Workshop on Rapid System Prototyping, 1997* (pp. 58–62). http://doi.org/10.1109/IWRSP.1997.618825

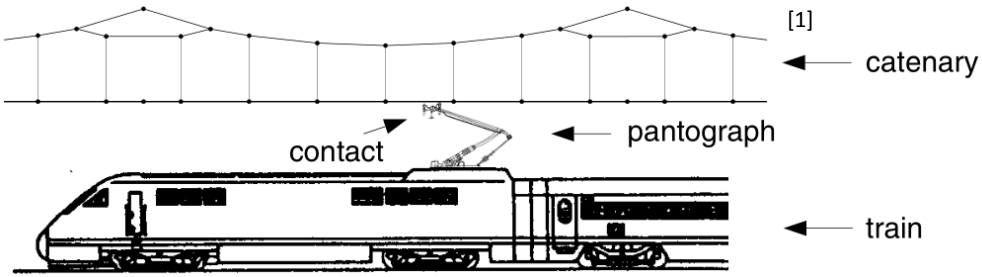
[1]



The Context

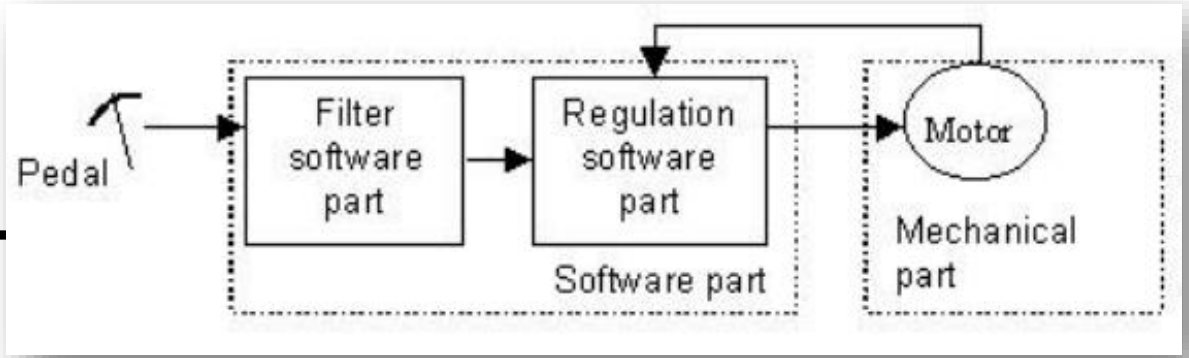
- SIMNET(Military Training)
  - SW/HW co-simulation.
    - DSBlock Standard
      - MPM&S Research agenda
        - DIS IEEE Standard
          - Multi-abstraction
            - Automotive, Railway, HVAC reported <u>applications</u>

1990 2000 2010



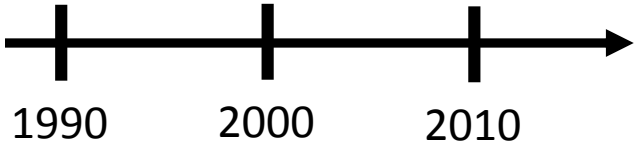
[1] Arnold, M., & Günther, M. (2001). Preconditioned Dynamic Iteration for Coupled Differential-Algebraic Systems. *BIT Numerical Mathematics*, *41*(1), 1–25. http://doi.org/10.1023/A:1021909032551

[2]



[2] Le Marrec, P., Valderrama, C. A., Hessel, F., Jerraya, A. A., Attia, M., & Cayrol, O. (1998). Hardware, software and mechanical cosimulation for automotive applications. In *9th International Workshop on Rapid System Prototyping* (pp. 202–206). http://doi.org/10.1109/IWRSP.1998.676692

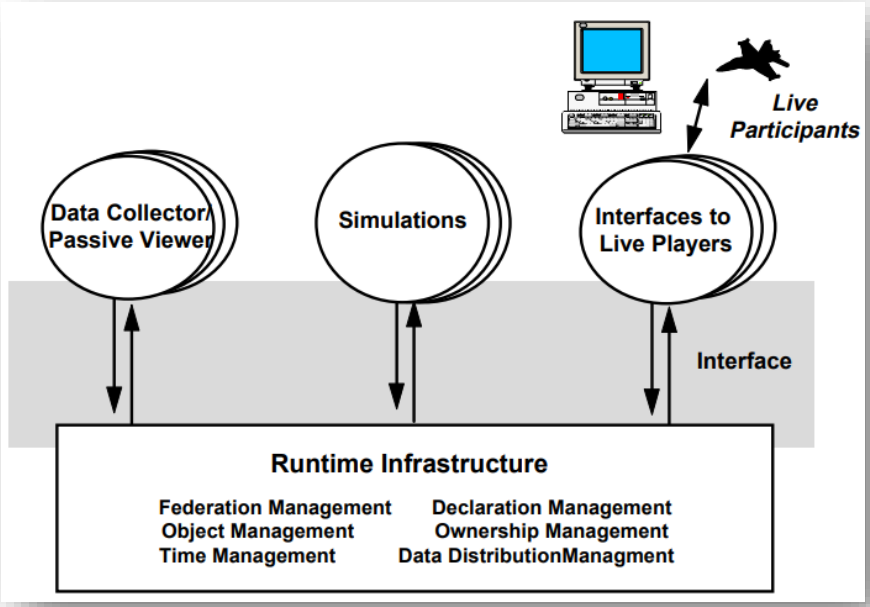
The Context

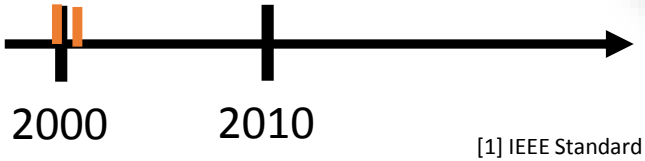


The Context

- Automotive, Railway, HVAC reported applications
- HLA IEEE Standard

[1]



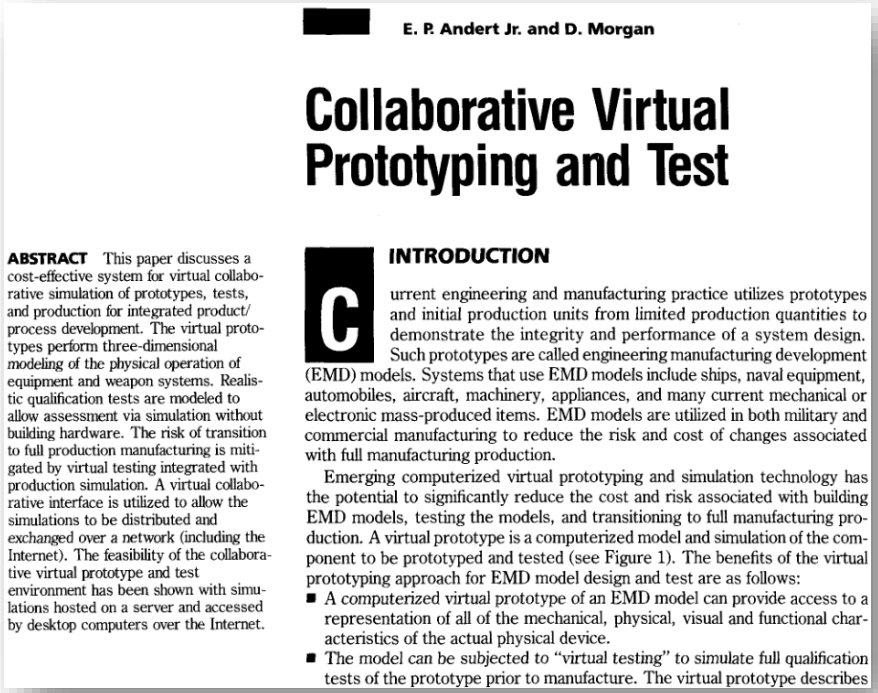


[1] IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) - Federate Interface Specification. (2010). IEEE Standard 1516-2010. Retrieved from https://standards.ieee.org/findstds/standard/1516-2010.html

The Context

- Automotive, Railway, HVAC reported applications
- HLA IEEE Standard
- DMU Concept

[1]



E. P. Andert Jr. and D. Morgan

Collaborative Virtual Prototyping and Test

ABSTRACT This paper discusses a cost-effective system for virtual collaborative simulation of prototypes, tests, and production for integrated product/process development. The virtual prototypes perform three-dimensional modeling of the physical operation of equipment and weapon systems. Realistic qualification tests are modeled to allow assessment via simulation without building hardware. The risk of transition to full production manufacturing is mitigated by virtual testing integrated with production simulation. A virtual collaborative interface is utilized to allow the simulations to be distributed and exchanged over a network (including the Internet). The feasibility of the collaborative virtual prototype and test environment has been shown with simulations hosted on a server and accessed by desktop computers over the Internet.

INTRODUCTION

Current engineering and manufacturing practice utilizes prototypes and initial production units from limited production quantities to demonstrate the integrity and performance of a system design. Such prototypes are called engineering manufacturing development (EMD) models. Systems that use EMD models include ships, naval equipment, automobiles, aircraft, machinery, appliances, and many current mechanical or electronic mass-produced items. EMD models are utilized in both military and commercial manufacturing to reduce the risk and cost of changes associated with full manufacturing production.

Emerging computerized virtual prototyping and simulation technology has the potential to significantly reduce the cost and risk associated with building EMD models, testing the models, and transitioning to full manufacturing production. A virtual prototype is a computerized model and simulation of the component to be prototyped and tested (see Figure 1). The benefits of the virtual prototyping approach for EMD model design and test are as follows:

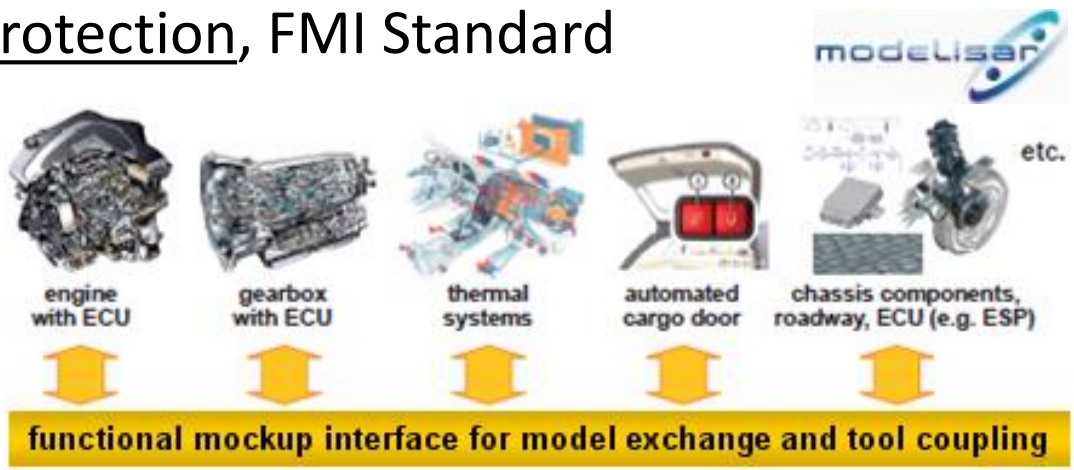
- A computerized virtual prototype of an EMD model can provide access to a representation of all of the mechanical, physical, visual and functional characteristics of the actual physical device.
- The model can be subjected to "virtual testing" to simulate full qualification tests of the prototype prior to manufacture. The virtual prototype describes

2000 2010

[1] Andert Jr., E. P., & Morgan, D. (1998). Collaborative Virtual Prototyping and Test. *Naval Engineers Journal*, *110*(6), 17–23. http://doi.org/10.1111/j.1559-3584.1998.tb02962.x

The Context

- Automotive, Railway, HVAC reported applications
- HLA IEEE Standard
- DMU Concept
  - IP Protection, FMI Standard



[1]

[1] www.fmi-standard.org

2000 2010

The Context

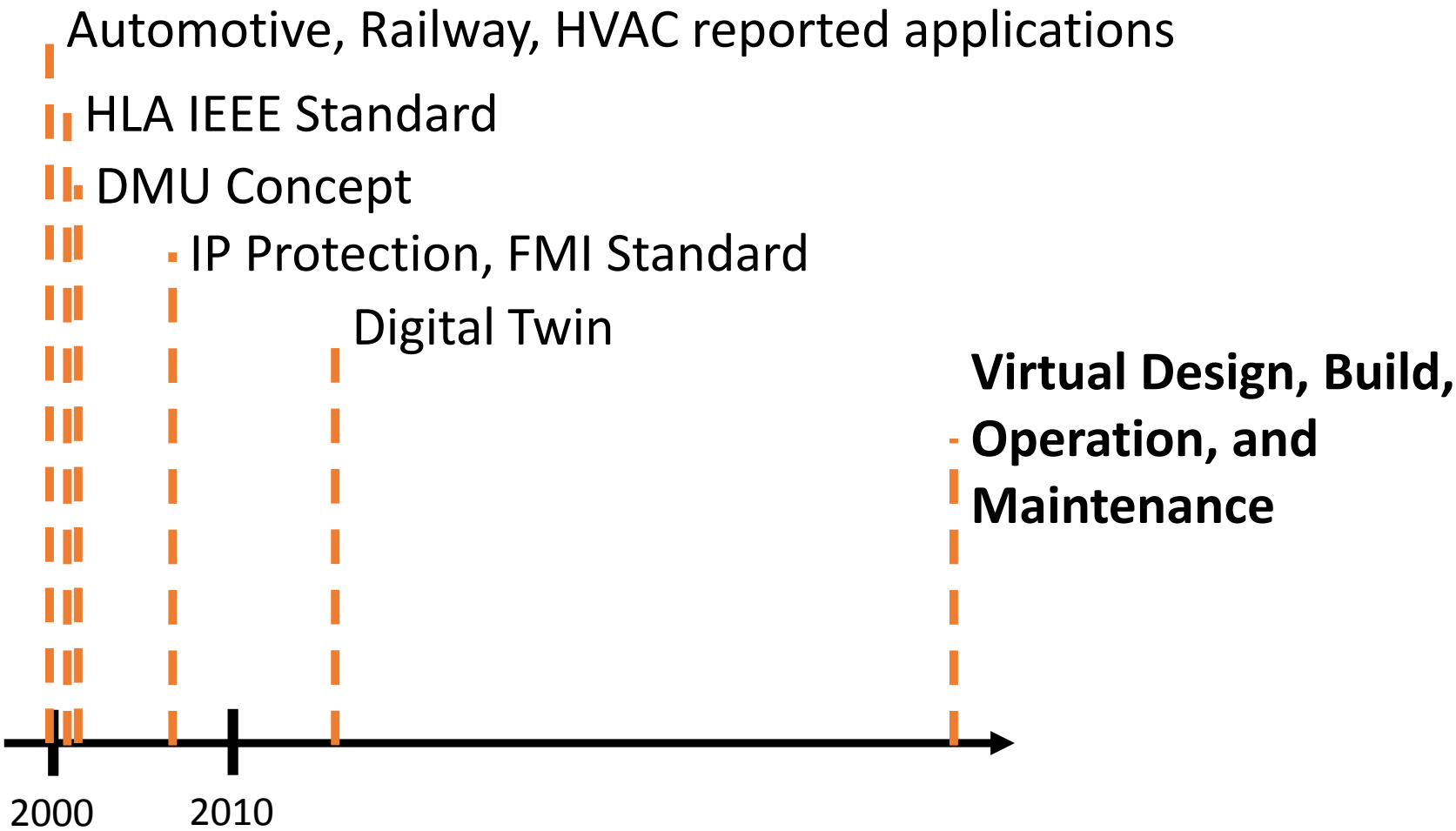
- Automotive, Railway, HVAC reported applications
- HLA IEEE Standard
- DMU Concept
  - IP Protection, FMI Standard
    - Digital Twin

2000 2010

[1]

[1] https://siemensplm.i.lithium.com/t5/image/serverpage/image-id/31800iB94D19235331E799?v=1.0

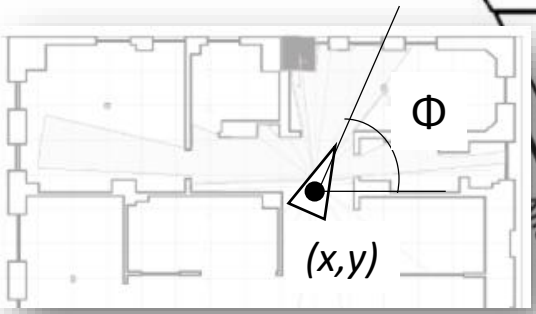
The Context

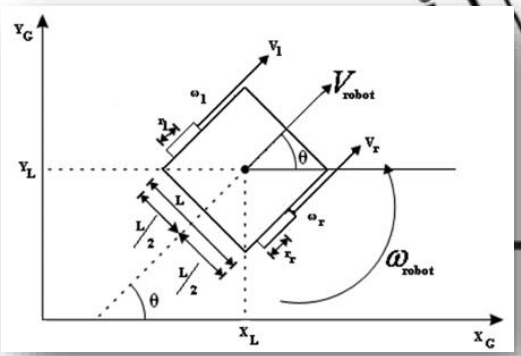

Automotive, Railway, HVAC reported applications
HLA IEEE Standard
DMU Concept
IP Protection, FMI Standard
Digital Twin
Virtual Design, Build, Operation, and Maintenance
2000
2010

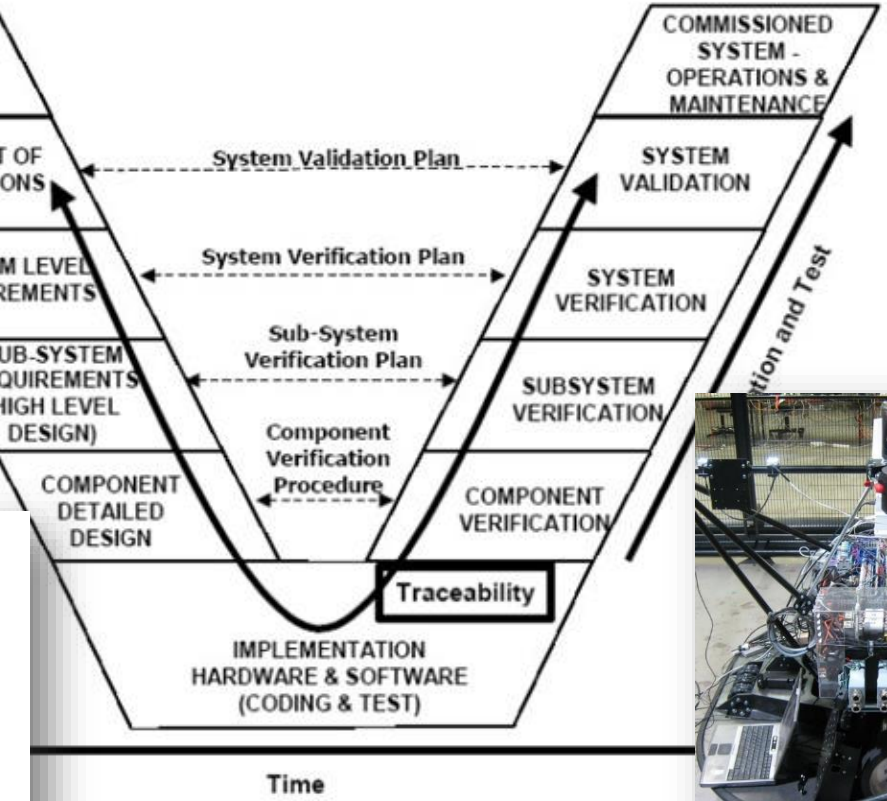
The Vision

Virtual Design, Build, Operation, and Maintenance

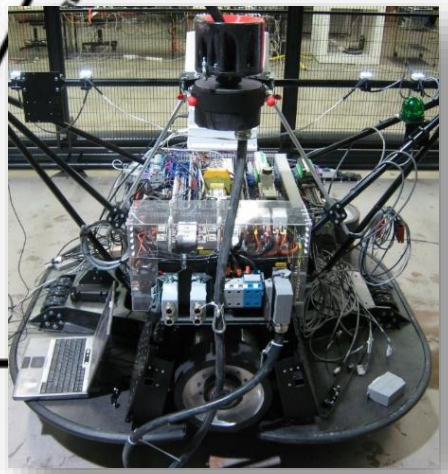
Virtual Design





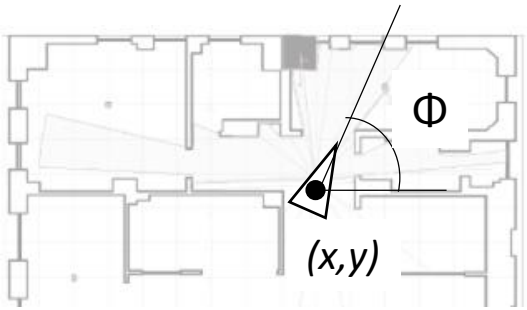


[2]

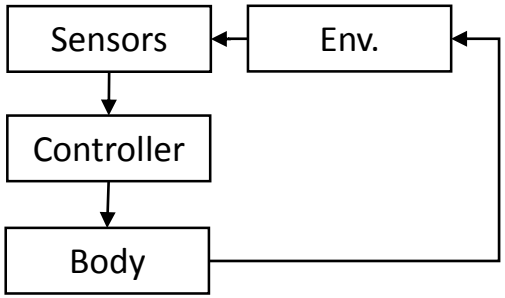


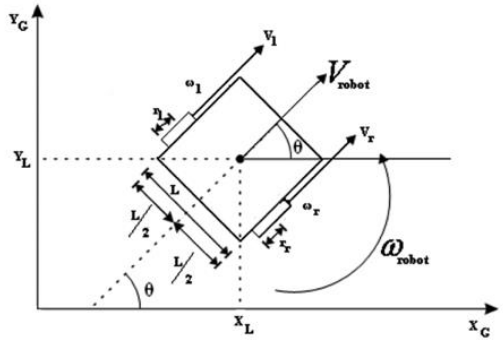
[2] Case Study by Kristof Berx , Davy Maes, and Klaas Gadeyne, from Flanders Make

Virtual Design

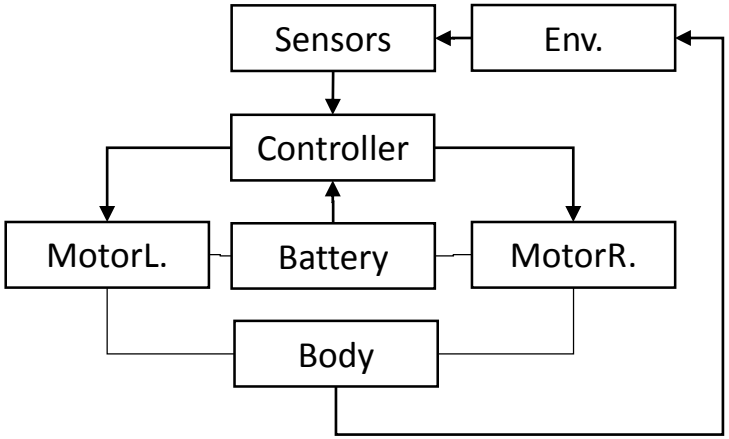


Frequent full system evaluation



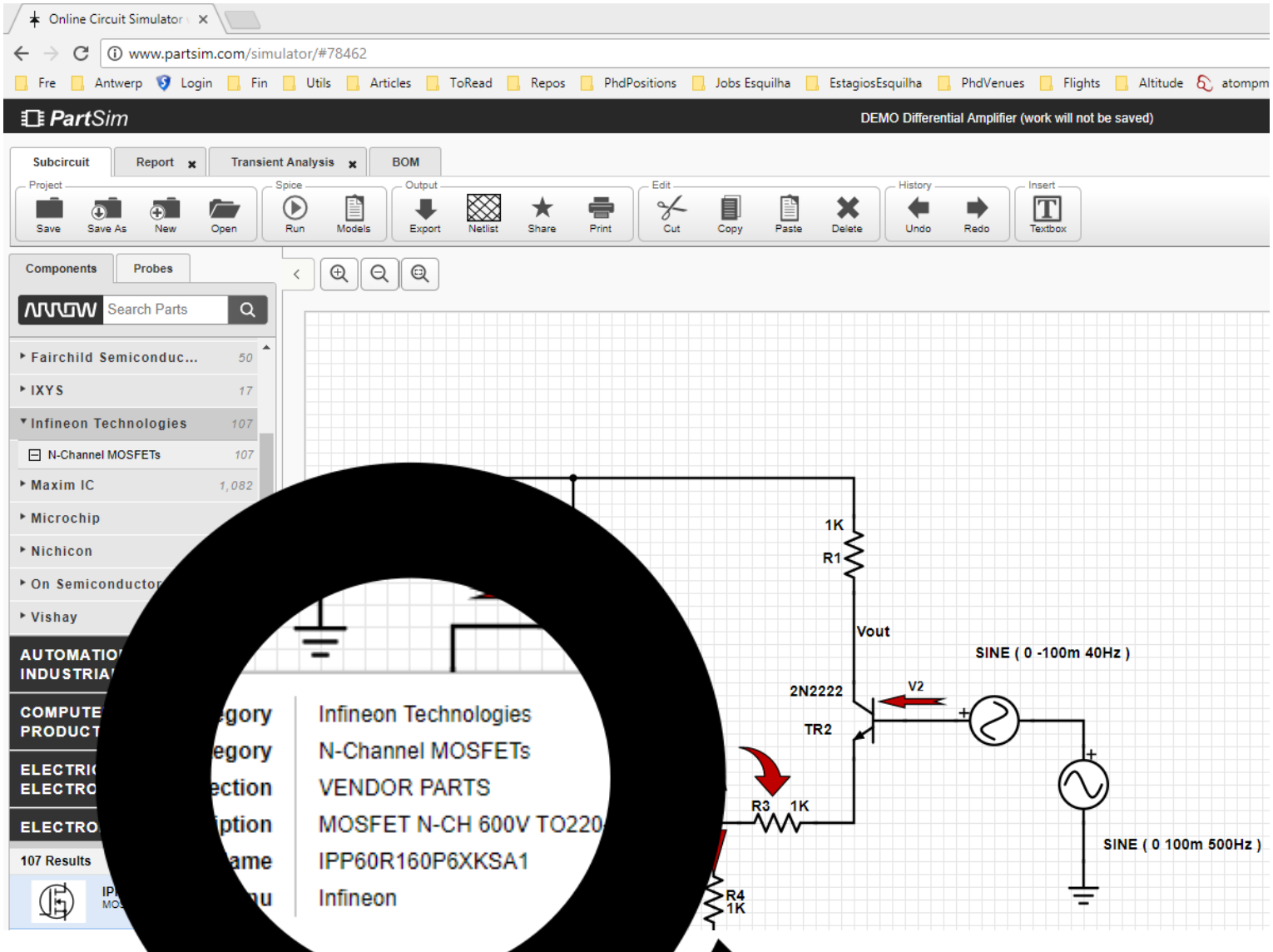


…at multiple levels of refinement

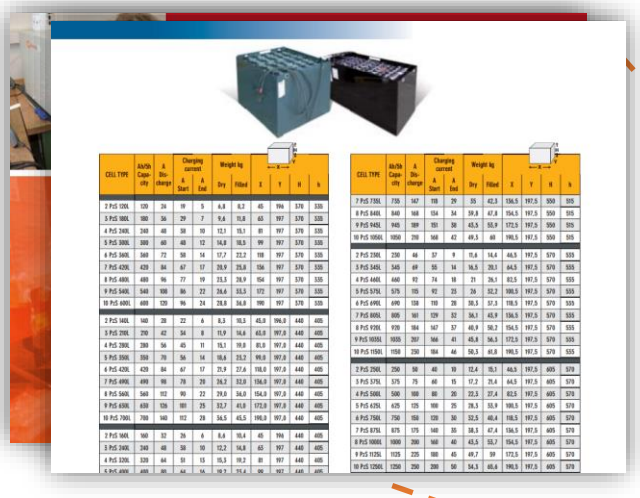


Virtual Build

[1] http://www.partsim.com/



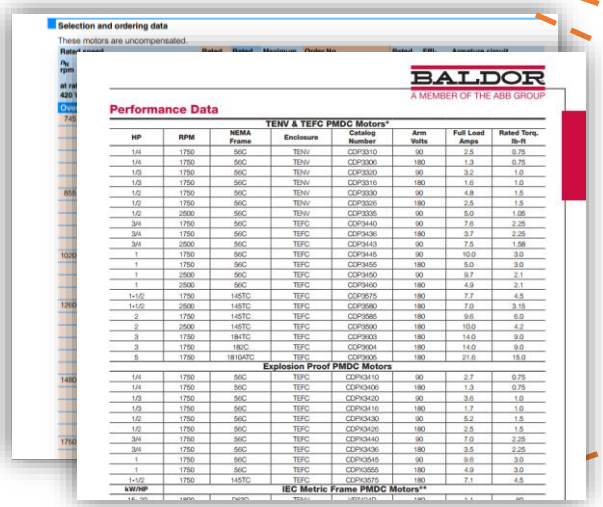
Virtual Build

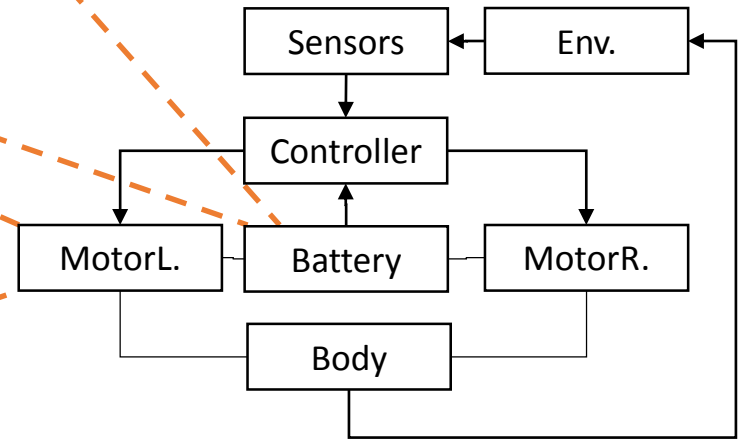


Frequent full system evaluation

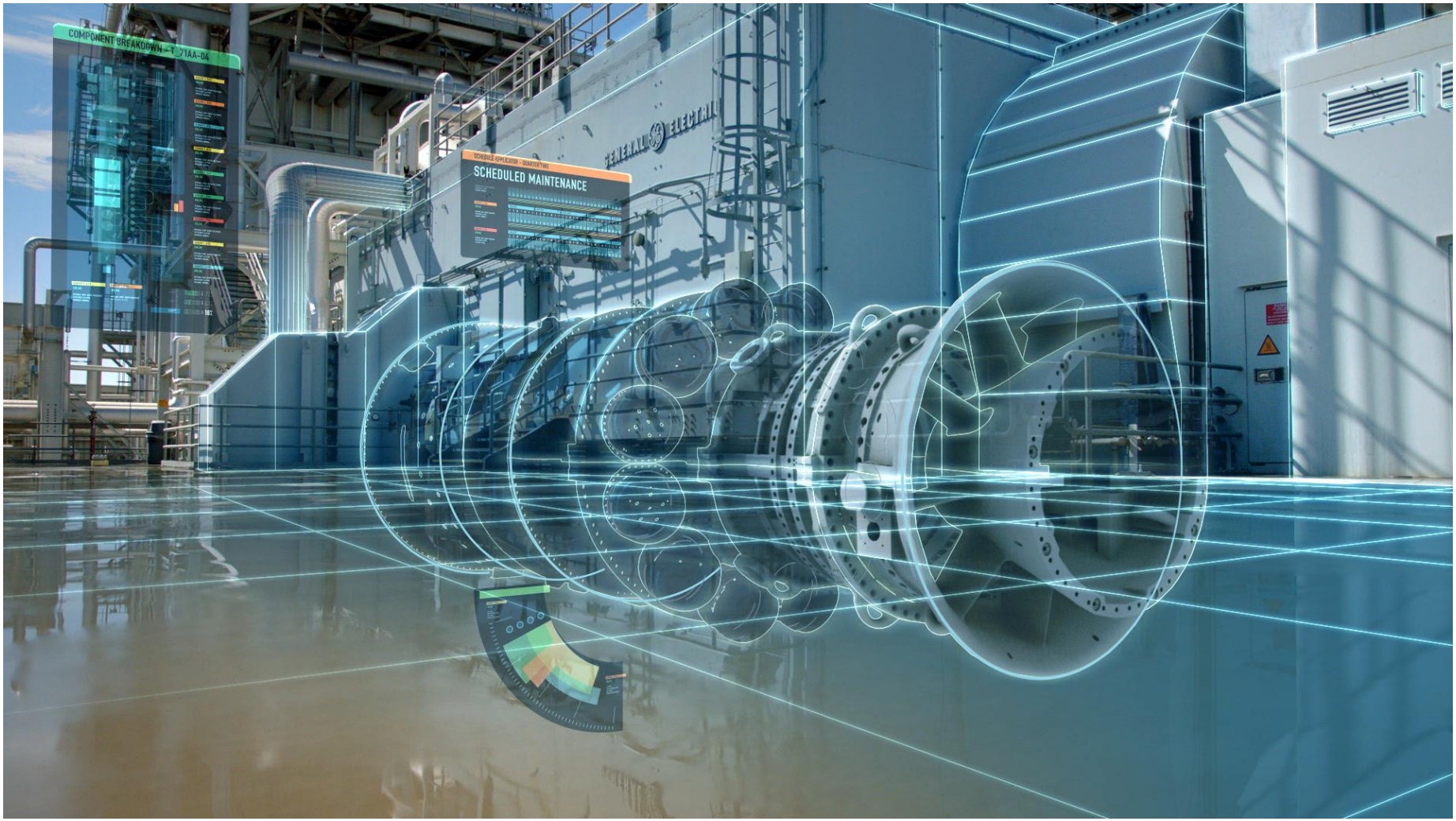
…at any level of refinement

…between multiple suppliers



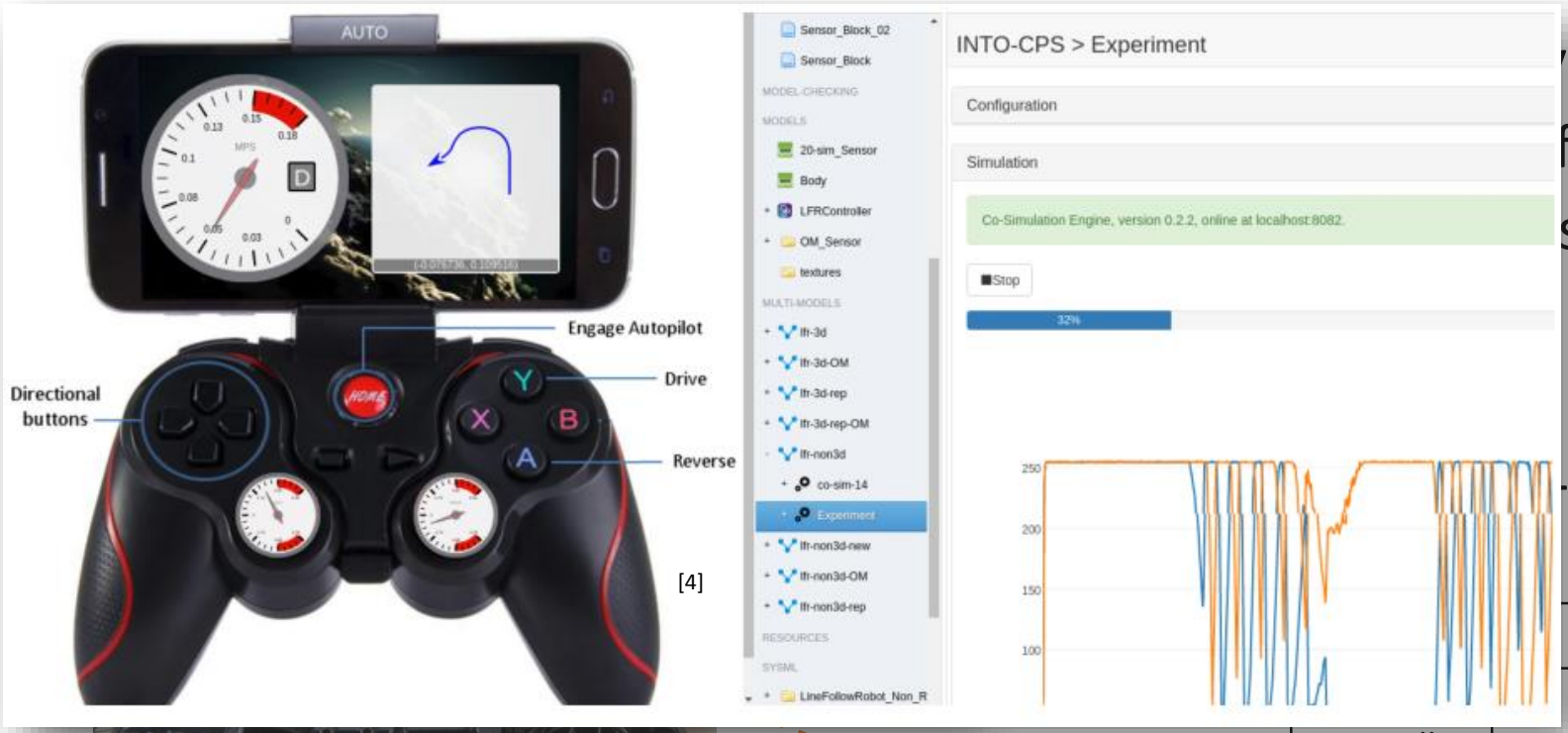


Virtual Operation

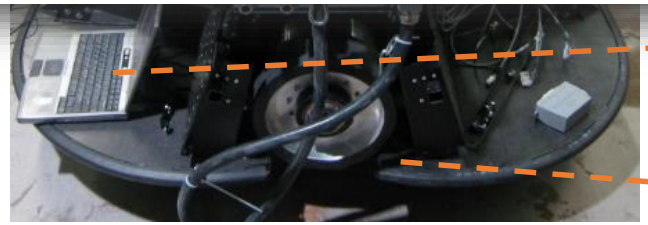


[1] http://www.power-technology.com/features/featuredigital-twin-the-living-breathing-and-failing-virtual-power-plant-5742648/

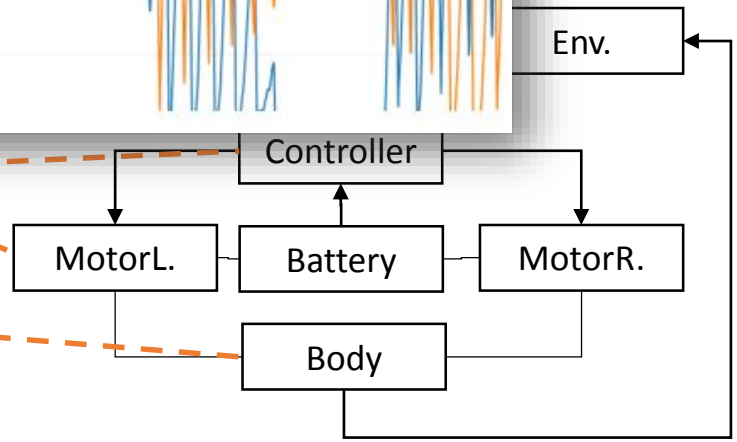
Virtual Operation



[4]



[2]



valuation

finement

suppliers

-the-loop

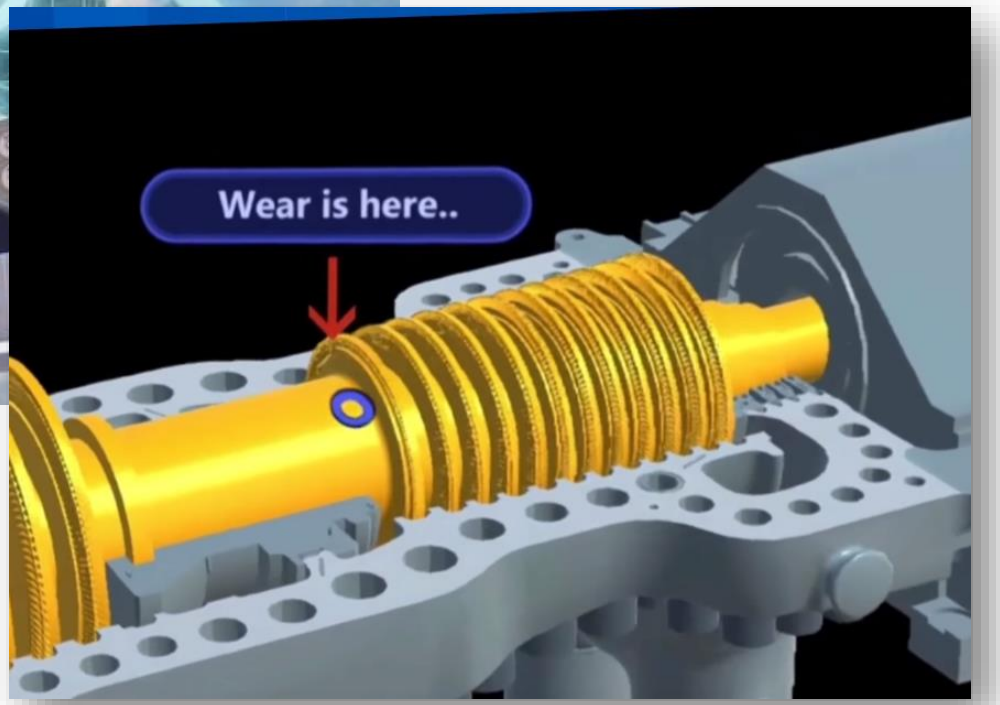
[2] Case Study by Kristof Berx , Davy Maes, and Klaas Gadeyne, from Flanders Make

[3] https://www.virtalis.com/blogs/2013/09/06/germanys-dlr-turns-to-virtalis-for-cockpit-simulator-upgrade-4/

[4] Palmieri, M., Bernardeschi, C., & Masci, P. (2017). Co-simulation of semi-autonomous systems : the Line Follower Robot case study. In *1st Workshop on Formal Co-Simulation.* Trento, Italy.

Virtual Maintenance

[1]



[2]

[1] https://www.fool.com/investing/2016/09/18/what-is-the-industrial-internet.aspx
[2] https://www.youtube.com/watch?v=2dCz3oL2rTw

The Baby Steps: Semantic adaptation

Our contributions

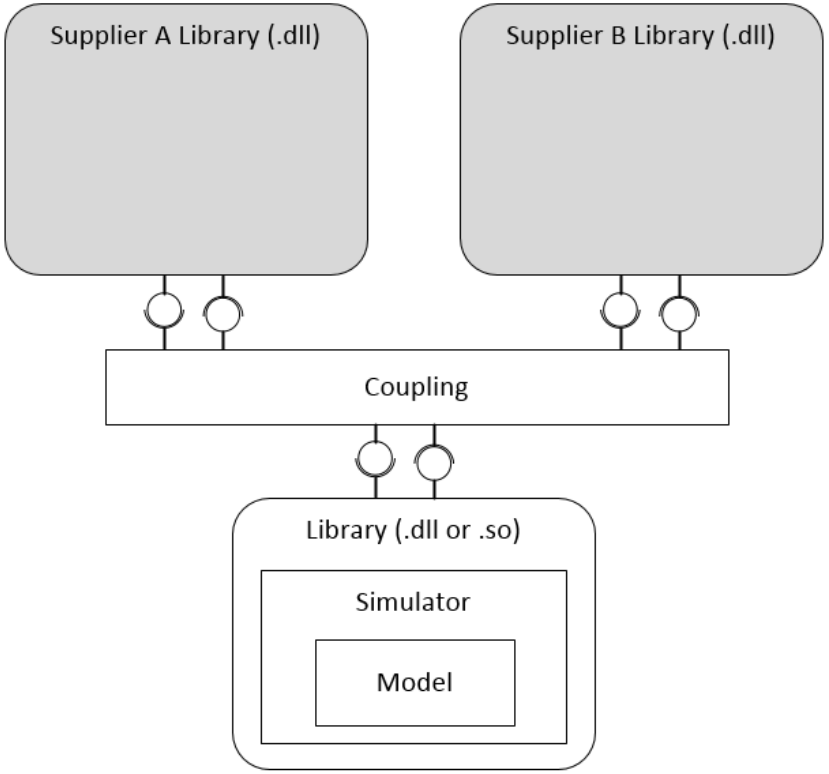
Bart Meyers, Joachim Denil, Casper Thule, Kenneth Lausdahl

Peter Gorm Larsen, Hans Vangheluwe, Paul De Meulenaere

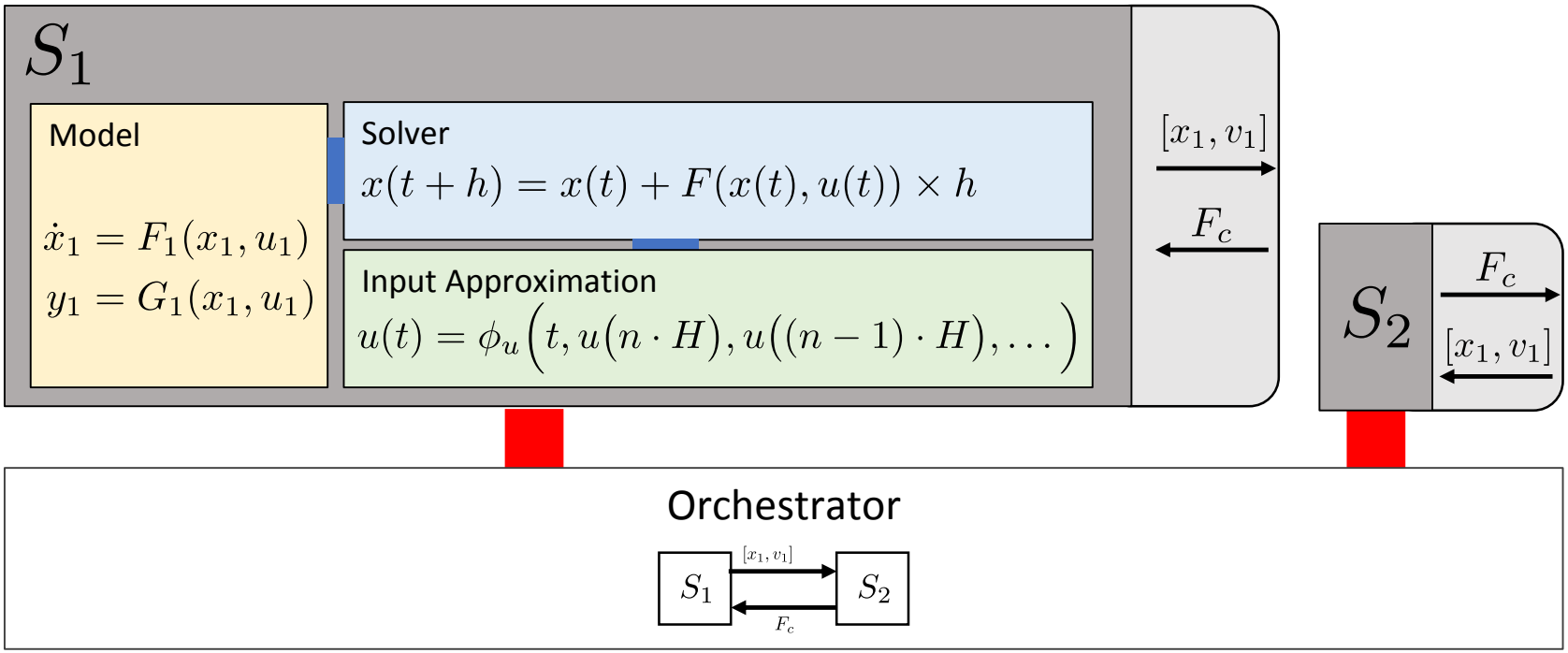
(2018) Semantic Adaptation for FMI Co-simulation with Hierarchical Simulators, in SIMULATION. To appear.

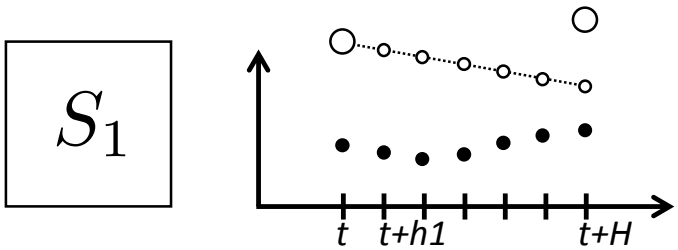
Functional Mock-up Interface Standard

- Standard interaction with any simulator
- A Functional Mockup Unit is a zip-file (.fmu) consisting of
  - C Library (.dll or .so)
  - XML file (metadata)
- Every simulator is a black box.
  - Executed locally but can communicate with a remote server
- The coupling (master algorithm) must be provided

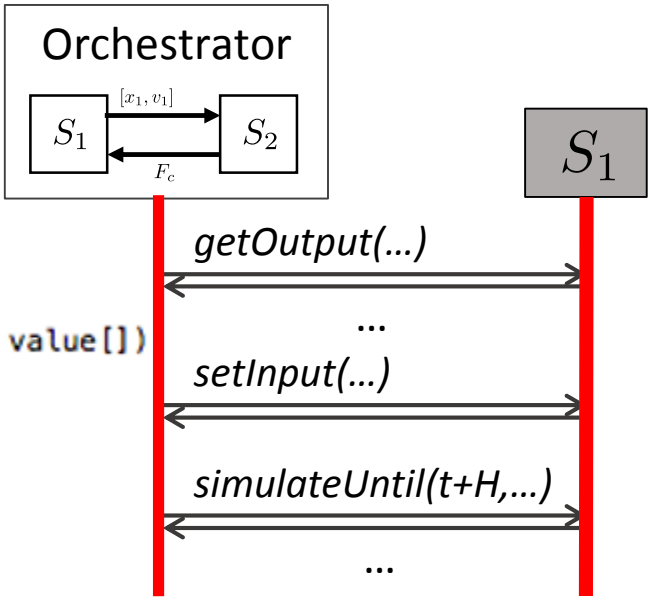


FMU (Conceptual) Internals





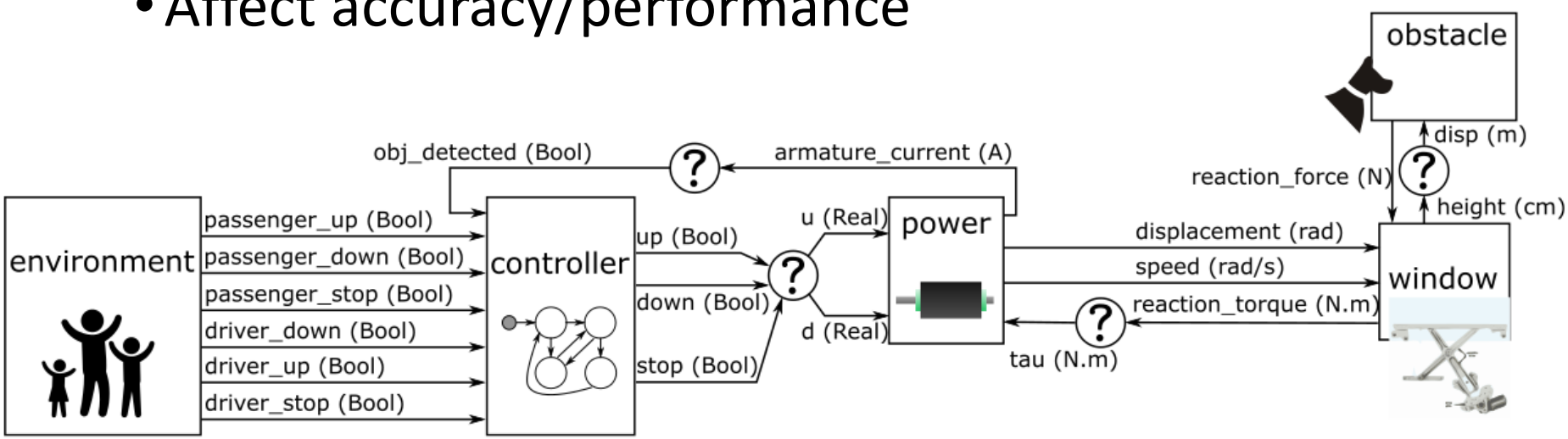
FMU Example




```
fmi2Status fmi2GetReal(fmi2Component fc,
{                const fmi2ValueReference vr[], size_t nvr, fmi2Real value[])
    FMUInstance* comp = (FMUInstance *)fc;
    int i;
    for (i = 0; i < nvr; i++)
    {
        value[i] = comp->r[(vr[i])];
    }
    return fmi2OK;
}

fmi2Status fmi2DoStep(fmi2Component fc , fmi2Real currentCommPoint, fmi2Real commStepSize, fmi2Boolean
    noPrevFMUState)
{
    FMUInstance* fi = (FMUInstance *)fc;
    fmi2Status simStatus = fmi2OK;
    printf("%s in fmiDoStep()\n",fi->instanceName);
    fi->currentTime = currentCommPoint + commStepSize;
    printf("Motor_in: %f\n", fi->r[_motor_in]);
    printf("slave CBD_PART2 now at time: %f\n", fi->currentTime);

    fi->r[_position] =  fi->r[_position] + fi->r[_velocity] * commStepSize;
    fi->r[_velocity] = fi->r[_velocity] + fi->r[_acceleration_after_friction] * commStepSize;
    fi->r[_friction] = fi->r[_velocity] * 5.81;
    fi->r[_motor_acceleration] = fi->r[_motor_in] * 40;
    fi->r[_acceleration_after_friction] = fi->r[_motor_acceleration] - fi->r[_friction];

    return simStatus;
}
```

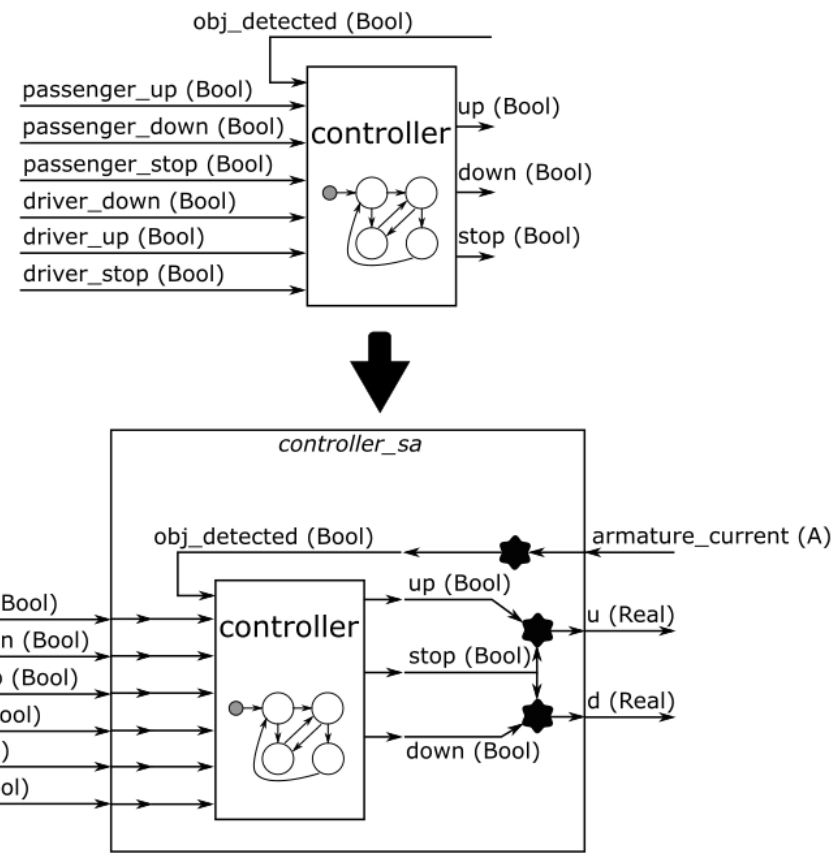
Motivation for Semantic Adaptation

- Quick and sound way of adapting the behaviour of an interconnected set of FMUs
  - Unit conversion
  - Interaction protocol modification
    - Time triggered vs Event triggered execution
  - Affect accuracy/performance

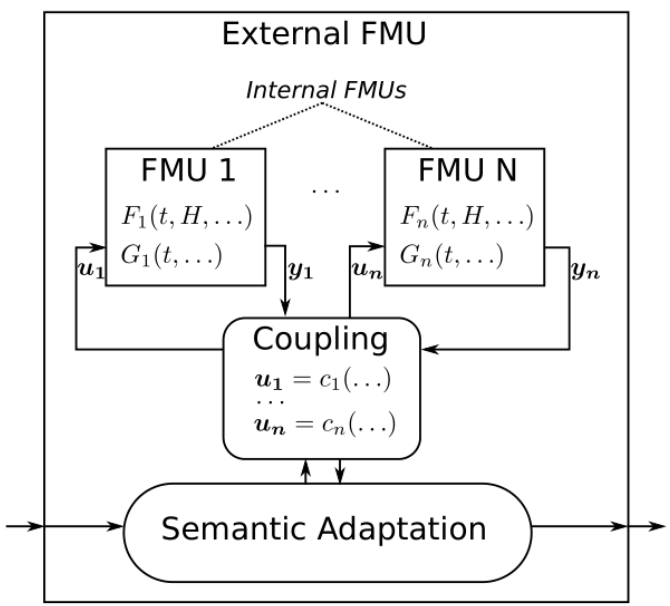


Semantic Adaptation

- Actions by which the **behavior** of an original set of interconnected FMUs is **altered**, following the **transparency** and **modularity** principles.

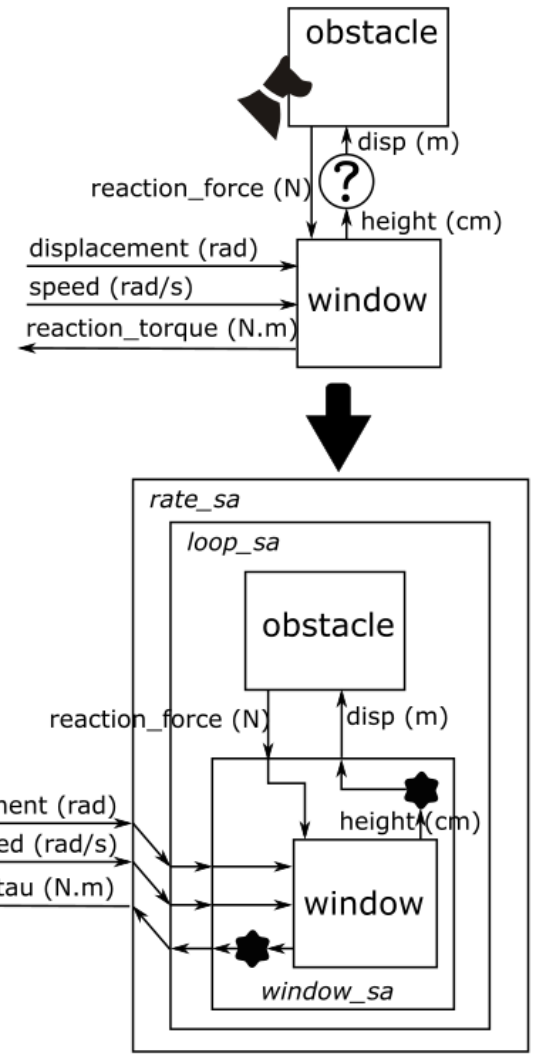


How?

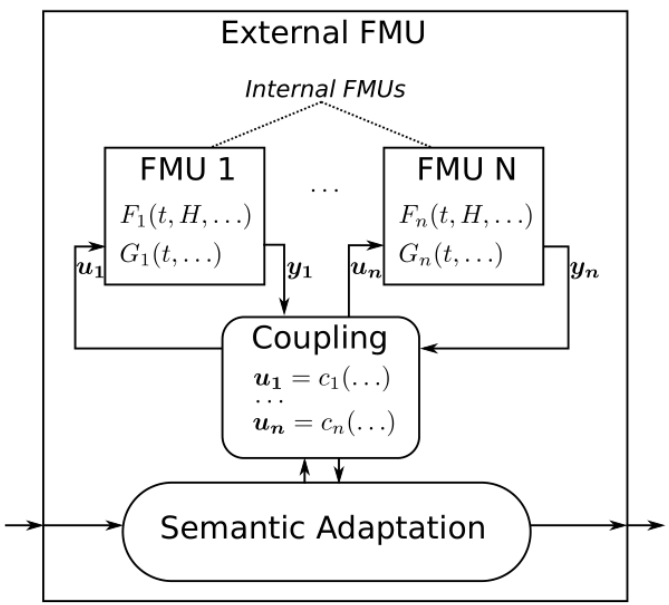


Semantic Adaptation

- Actions by which the **behavior** of an original set of interconnected FMUs is **altered**, following the **transparency** and **modularity** principles.

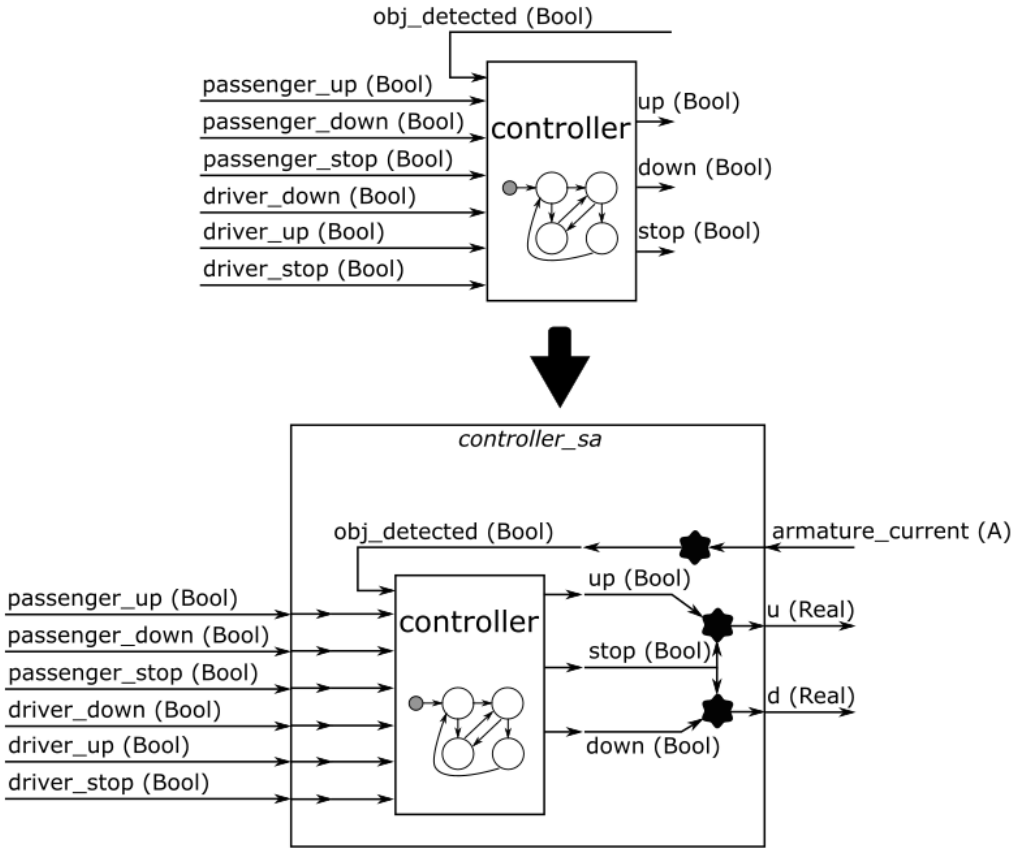


How?



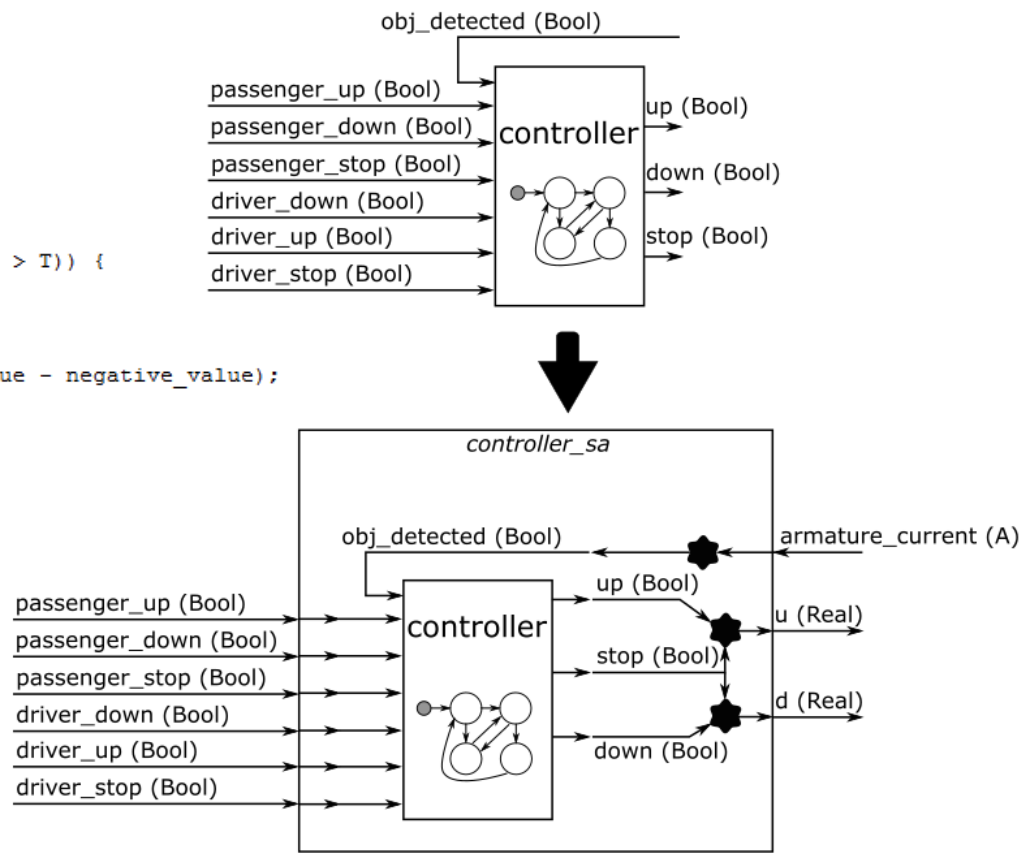
A DSL for Semantic Adaptation

```
semantic adaptation reactive moore ControllerSA controller_sa
at "./path/to/ControllerSA.fmu"

    for inner fmu Controller ctrl
    at "./path/to/LazySA.fmu"
    with input ports obj_detected, passenger_up, passenger_d
    with output ports up, down, stop

input ports armature_current -> ctrl.obj_detected,
            passenger_up -> ctrl.passenger_up,
            passenger_down -> ctrl.passenger_down,
            passenger_stop -> ctrl.passenger_stop,
            driver_up -> ctrl.driver_up,
            driver_down -> ctrl.driver_down,
            driver_stop -> ctrl.driver_stop

output ports    u, d
```

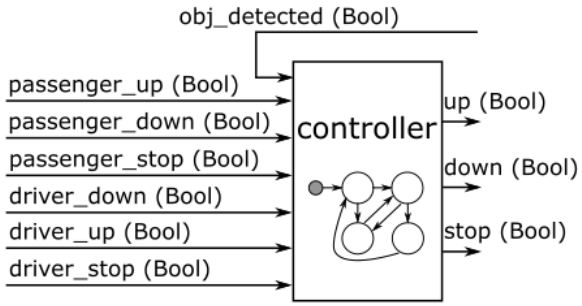


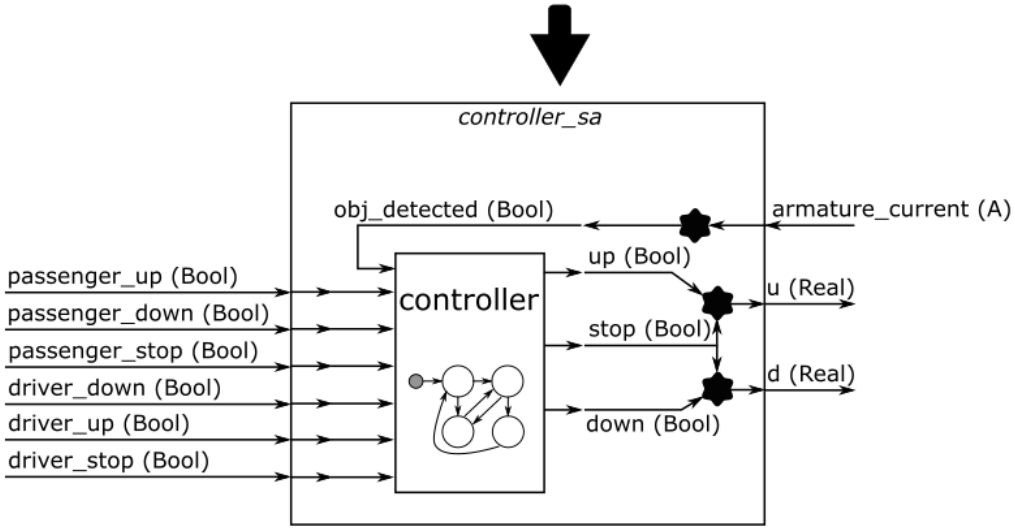
A DSL for Semantic Adaptation

```
control rules {
    var step_size := H;
    var aux_obj_detected := false;
    var crossedTooFar := false;
    if ((not is_close(p_v, T, RTOL, ATOL) and p_v < T)
                and (not is_close(f_v, T, RTOL, ATOL) and f_v > T)) {
        crossedTooFar := true;
        var negative_value := p_v - T;
        var positive_value := f_v - T;
        step_size := (H * (- negative_value)) / (positive_value - negative_value);
    } else {
        if ((not is_close(p_v, T, RTOL, ATOL) and p_v < T)
                    and is_close(f_v, T, RTOL, ATOL )) {
            c := true;
        }
    }

    if (not crossedTooFar){
        step_size := do_step(ctrl, t, H);
    }

    if (is_close(step_size, H, RTOL, ATOL)) {
        p_v := f_v;
    }
    return step_size;
}
```



A DSL for Semantic Adaptation






```
in var  f_v := INIT_V;
in rules {
    true -> {
        f_v := controller_sa.armature_current;
    } --> {
        ctrl.obj_detected := c;
    };
}

out rules {
    ctrl.up -> { } --> {controller_sa.u := 1.0; };
    not ctrl.up -> { } --> {controller_sa.u := 0.0; };

    ctrl.down -> { } --> {controller_sa.d := 1.0; };
    not ctrl.down -> { } --> {controller_sa.d := 0.0; };

    ctrl.stop -> { } --> {controller_sa.u := 0.0 ; controller_sa.d := 0.0; };
}
```

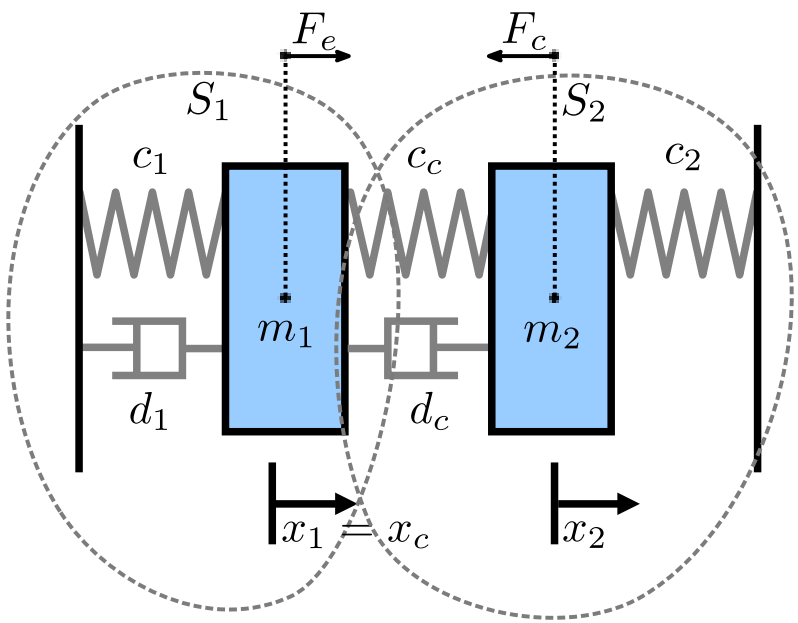
The Baby Steps: Stability of Adaptive Co-simulation

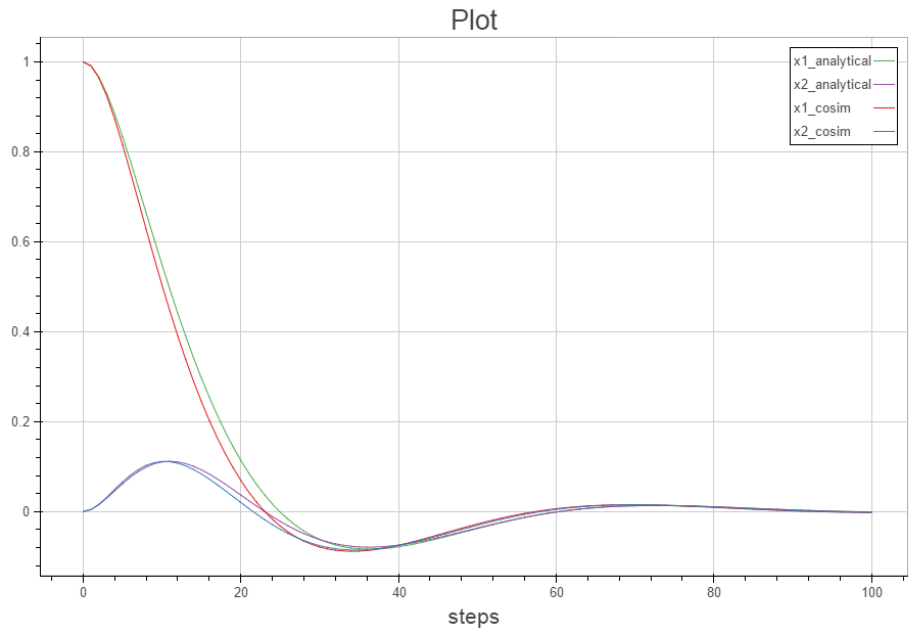
Our contributions

Benoît Legat, Raphael Jungers, Hans Vangheluwe

(2017). Stable Adaptive Co-simulation : A Switched Systems Approach. In IUTAM Symposium on Co-Simulation and Solver Coupling (p. to appear). Darmstadt, Germany.

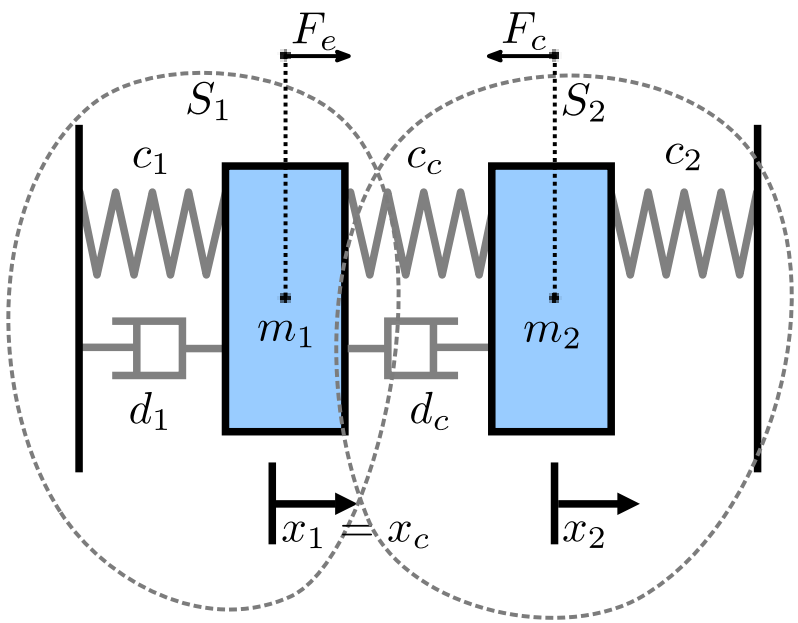
Running Example – Original System

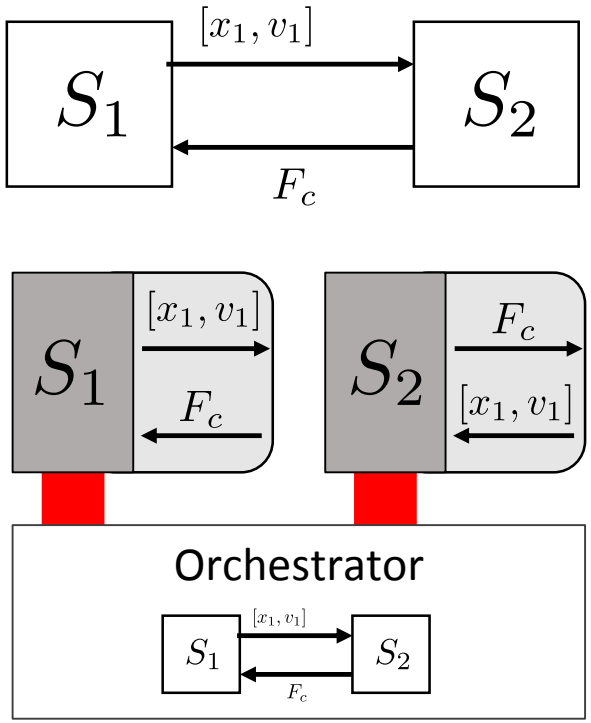




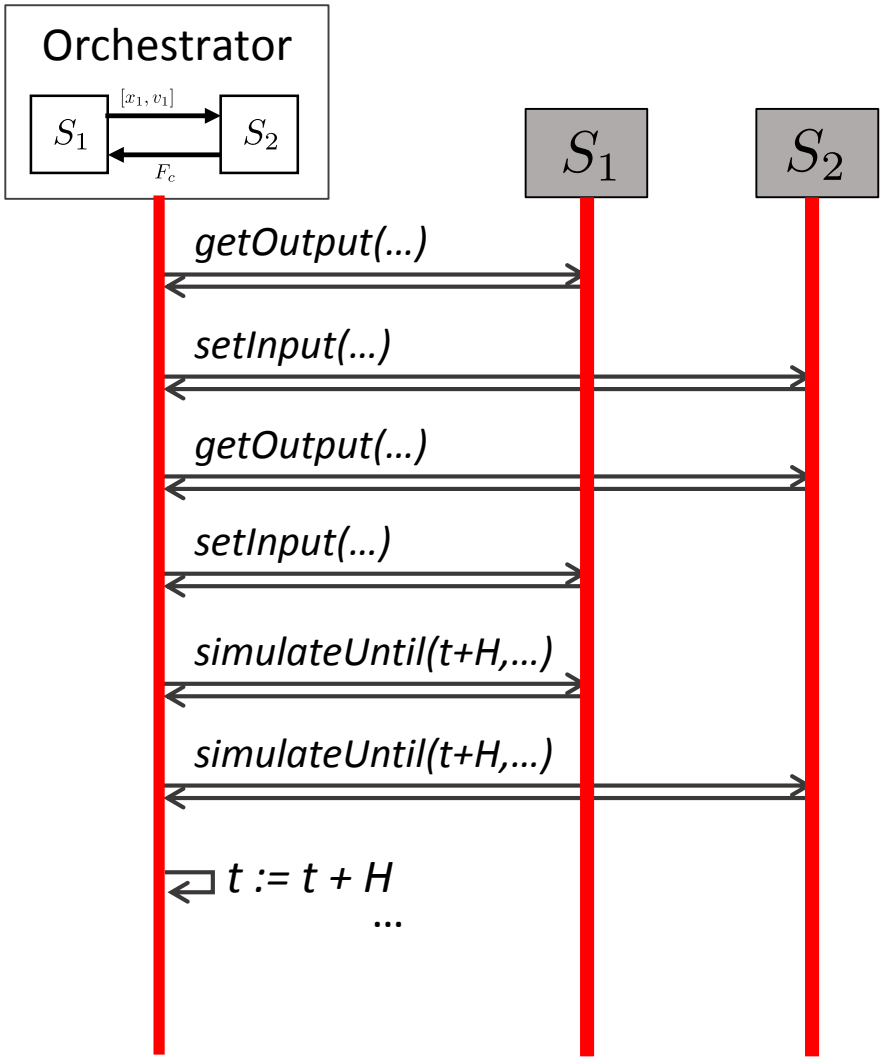
Busch, M. (2016). Continuous approximation techniques for co-simulation methods: Analysis of numerical stability and local error. *ZAMM - Journal of Applied Mathematics and Mechanics*, *96*(9), 1061–1081. http://doi.org/10.1002/zamm.201500196

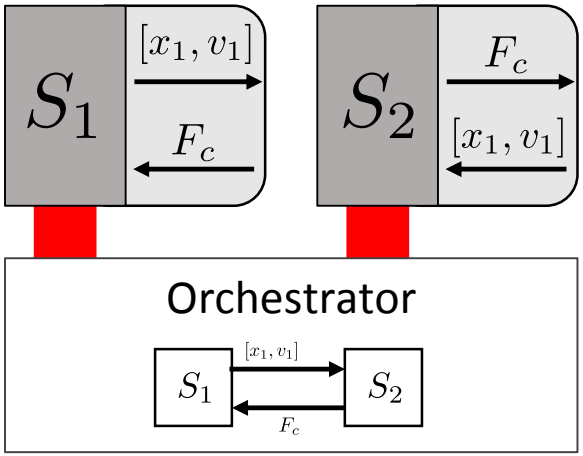
Running Example – Co-simulation

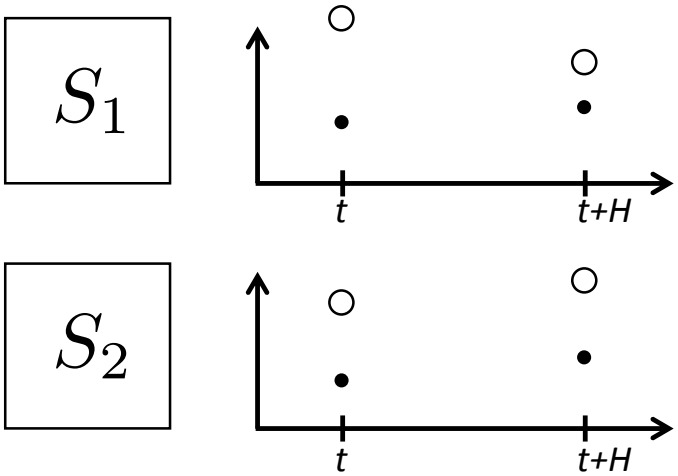




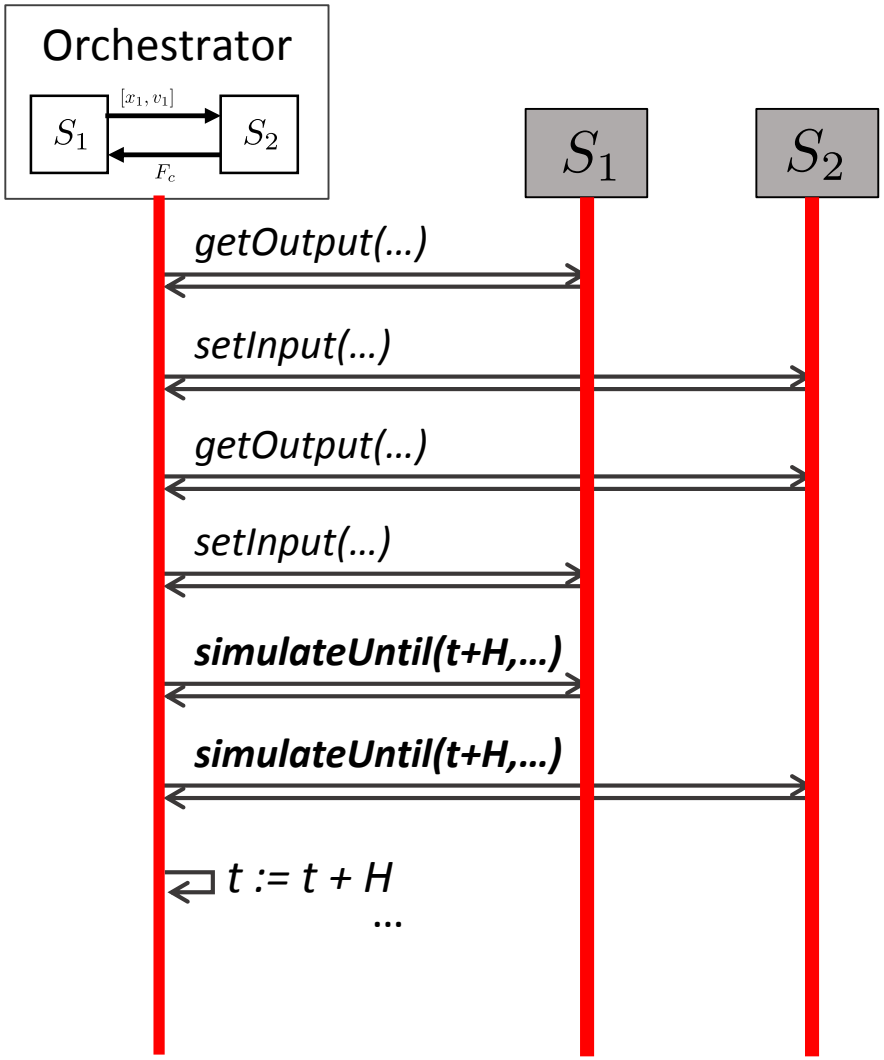
Orchestration

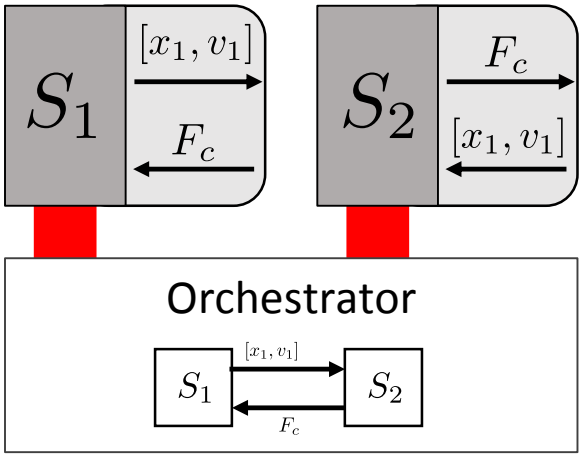


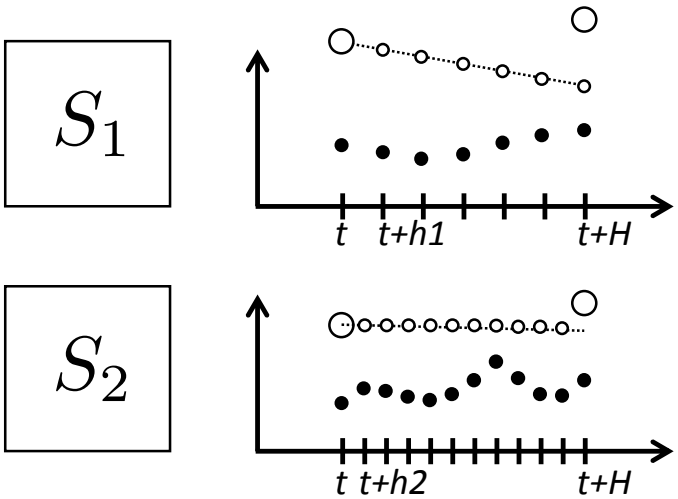




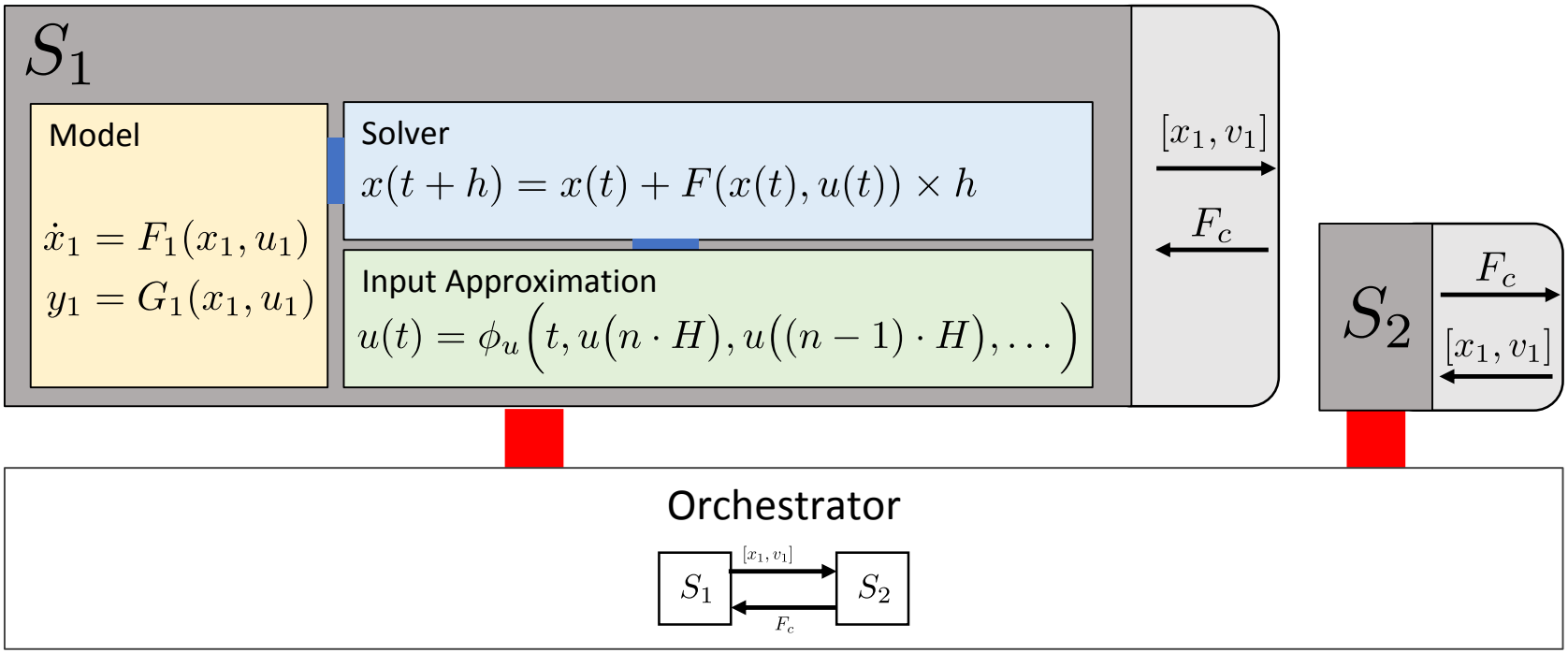
Internal Behavior



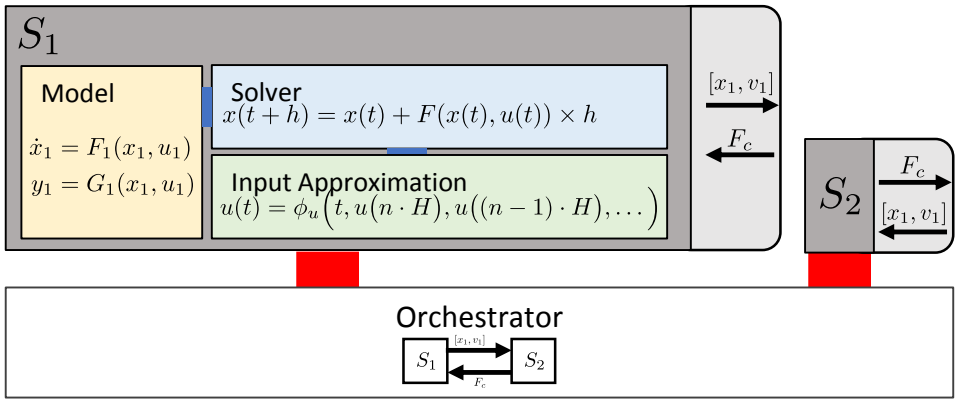




Simulator Internals



Orchestration Space – Inputs



Busch, M., & Schweizer, B. (2011). Stability of Co-Simulation Methods Using Hermite and Lagrange Approximation Techniques. In *ECCOMAS Thematic Conference on Multibody Dynamics* (pp. 1–10). Brussels, Belgium.

Stettinger, G., Horn, M., Benedikt, M., & Zehetner, J. (2014). Model-based coupling approach for non-iterative real-time co-simulation. In *2014 European Control Conference (ECC)* (pp. 2084–2089). http://doi.org/10.1109/ECC.2014.6862242

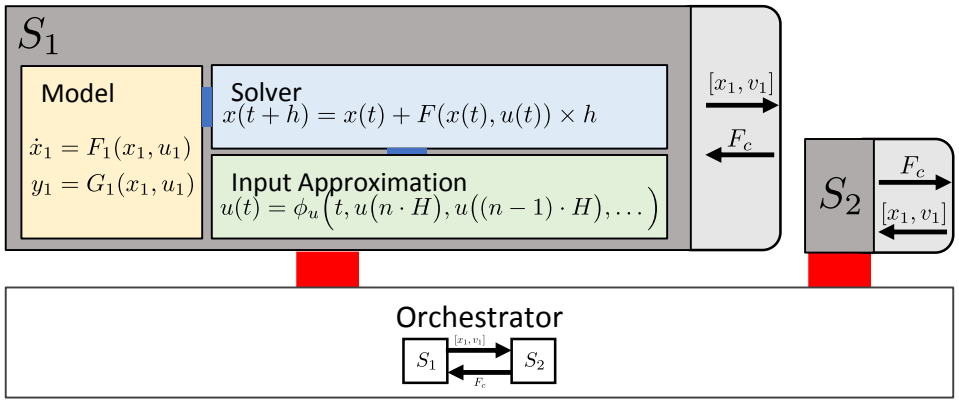
Input Approximations

Extrapolation	×	Polynomial °0	×	C^0 Continuous	×	…
Interpolation		Polynomial °1		C^1 Continuous		
		…		…		
		Context-aware				
		Model ID'ed				

Ben Khaled-El Feki, A., Duval, L., Faure, C., Simon, D., & Ben Gaid, M. (2017). CHOPtrey: contextual online polynomial extrapolation for enhanced multi-core co-simulation of complex systems. *SIMULATION*, 3754971668402. http://doi.org/10.1177/0037549716684026

Burden, R. L., & Faires, J. D. (2010). *Numerical Analysis* (9th ed.). Cengage Learning.

Orchestration Space – Solvers



Numerical Solvers

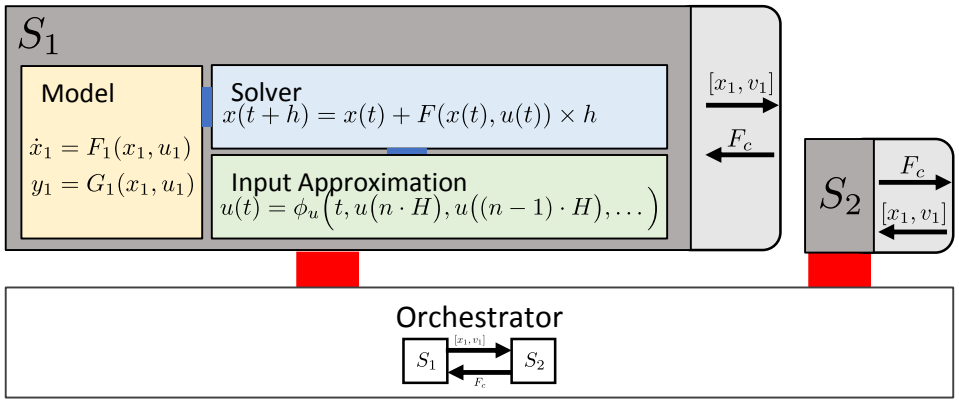
Parallel	×	Order 0	×	Implicit	×	Step size	×	…
Sequential		Order 1		Semi-Explicit				
		…		Explicit				

Hairer, E., & Wanner, G. (1996). *Solving ordinary differential equations II: Stiff and differential-algebraic problems*.

Orchestration Space – Synchronization

Arnold, M., Clauß, C., & Schierz, T. (2014). Error Analysis and Error Estimates for Co-simulation in FMI for Model Exchange and Co-Simulation v2.0. In S. Schöps, A. Bartel, M. Günther, W. E. J. ter Maten, & C. P. Müller (Eds.), *Progress in Differential-Algebraic Equations* (pp. 107–125). Berlin, Heidelberg: Springer Berlin Heidelberg. http://doi.org/10.1007/978-3-662-44926-4_6

Schweizer, B., & Lu, D. (2015). Predictor/corrector co-simulation approaches for solver coupling with algebraic constraints. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift Für Angewandte Mathematik Und Mechanik*, *95*(9), 911–938. http://doi.org/10.1002/zamm.201300191

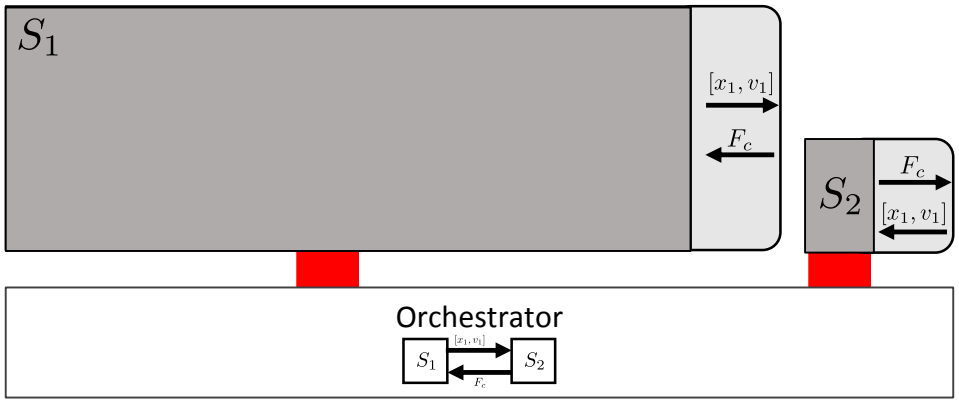


Orchestration Algorithms

Parallel	×	Jacobi	×	Implicit	×	Step size	×	…
Sequential		Gauss-Seidel		Semi-Explicit				
		…		Explicit				

Gomes, C., Thule, C., Broman, D., Larsen, P. G., & Vangheluwe, H. (2017). *Co-simulation: State of the art*. Retrieved from http://arxiv.org/abs/1702.00686

Capability Interaction



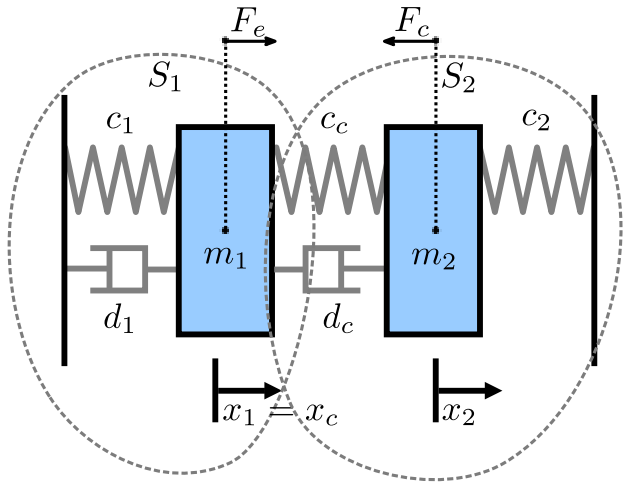
Input Approximations

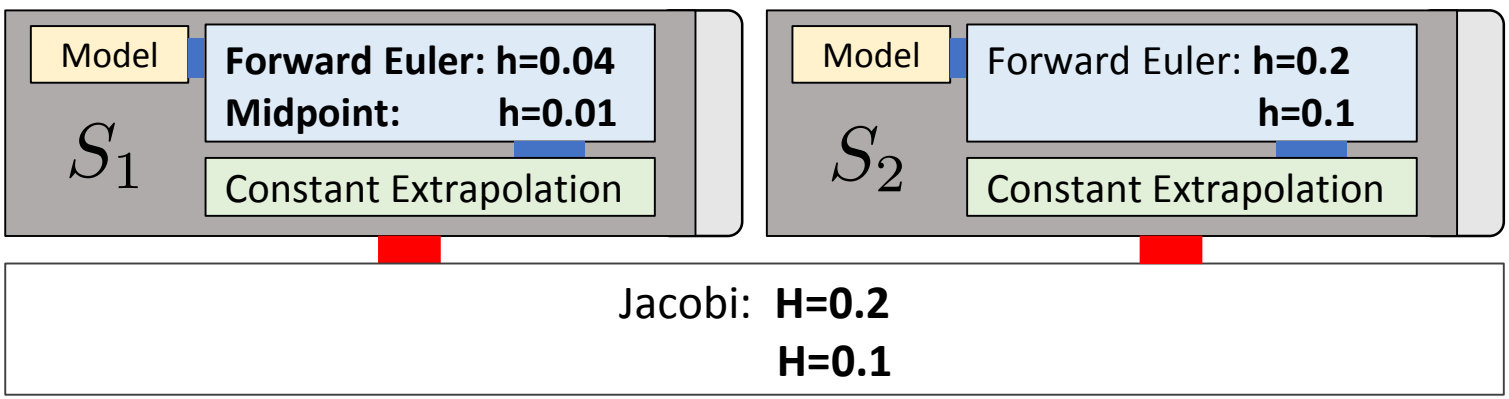
Extrapolation	×	Polynomial °0	×	C^0 Continuous	×	…
Interpolation		Polynomial °1		…		
		…				

Orchestration Algorithms

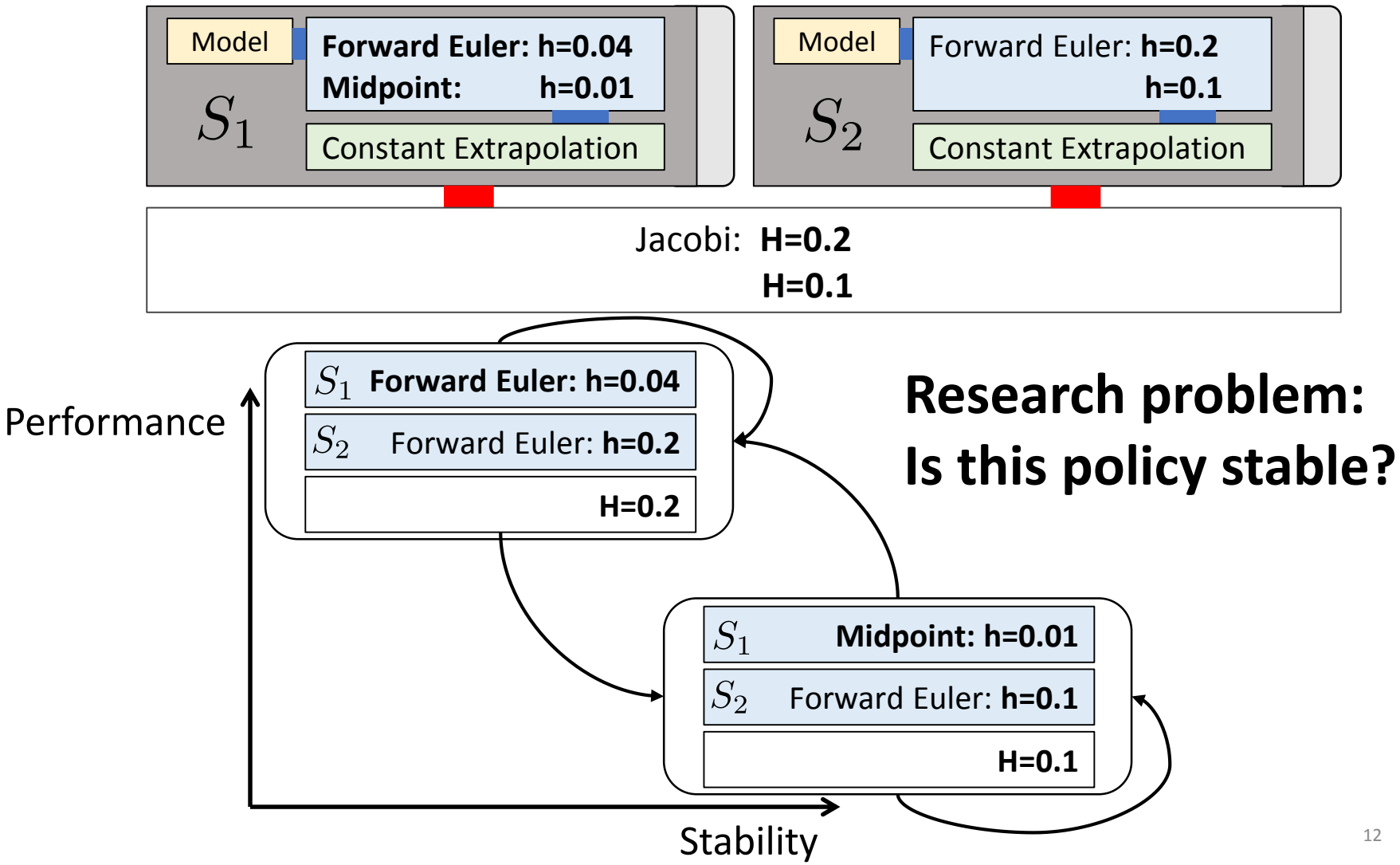
Parallel	×	Jacobi	×	Implicit	×	Step size	×	…
Sequential		Gauss-Seidel		Semi-Explicit				
		…		Explicit				

Adaptive Orchestration





Adaptive Orchestration



Research problem:
Is this policy stable?

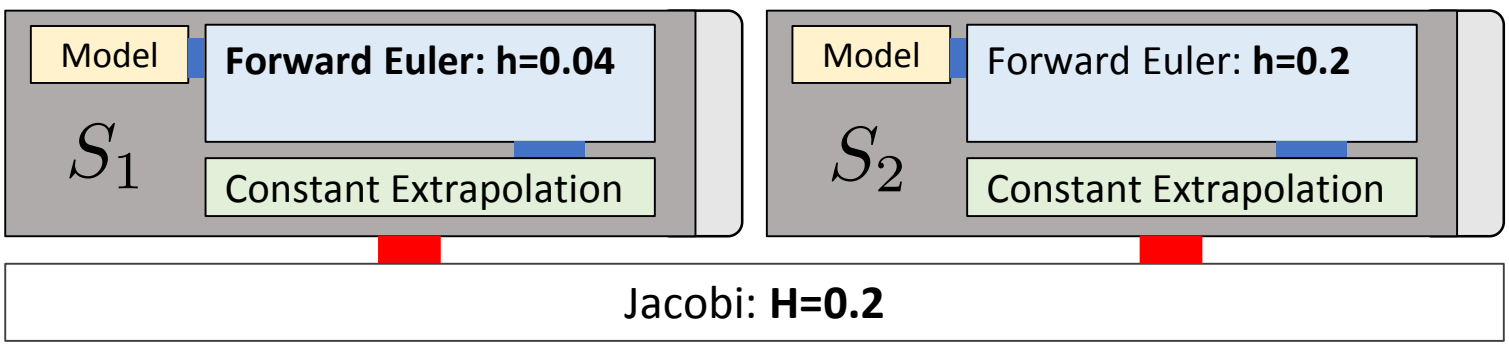
Non-adaptive Stability Analysis

$$\begin{bmatrix} x_1^{(n+1)} \\ v_1^{(n+1)} \end{bmatrix} = A_1^{k_1} \begin{bmatrix} x_1^{(n)} \\ v_1^{(n)} \end{bmatrix} + \left(\sum_{m=0}^{k_1-1} A_1^m B_1 \right) u_1^{(n)}$$

$$\begin{bmatrix} x_2^{(n+1)} \\ v_2^{(n+1)} \end{bmatrix} = A_2^{k_2} \begin{bmatrix} x_2^{(n)} \\ v_2^{(n)} \end{bmatrix} + \left(\sum_{m=0}^{k_2-1} A_2^m B_2 \right) u_2^{(n)}$$

$$y_1^{(n)} = \begin{bmatrix} x_1^{(n)} \\ v_1^{(n)} \end{bmatrix}$$

$$y_2^{(n)} = \begin{bmatrix} c_k & d_k \end{bmatrix} \begin{bmatrix} x_2^{(n)} \\ v_2^{(n)} \end{bmatrix} + \begin{bmatrix} -c_k & -d_k \end{bmatrix} u_2^{(n)}$$



$$\begin{bmatrix} x_1^{(n+1)} \\ v_1^{(n+1)} \\ x_2^{(n+1)} \\ v_2^{(n+1)} \end{bmatrix} = \mathbf{A}_{H=0.2, S_2=\langle FE, 0.04\rangle, S_2=\langle FE, 0.2\rangle} \begin{bmatrix} x_1^{(n)} \\ v_1^{(n)} \\ x_2^{(n)} \\ v_2^{(n)} \end{bmatrix}$$

Busch, M., & Schweizer, B. (2010). Numerical stability and accuracy of different co-simulation techniques: analytical investigations based on a 2-DOF test model. In *1st Joint International Conference on Multibody System Dynamics* (pp. 25–27).

Stability definition

$$\mathbf{x^{(n+1)}} = \mathbf{A}\mathbf{x^{(n)}}$$

Non-adaptive:

$$\forall \mathbf{x^{(0)}} \quad \lim_{k\to\infty} \left\| \underbrace{AA\ldots A}_{k\ \ times} \mathbf{x^{(0)}} \right\| = 0$$

Adaptive:

$$\mathbf{x^{(n+1)}} = \begin{cases} \mathbf{A}_1\mathbf{x^{(n)}} \\ or \quad \mathbf{A}_2\mathbf{x^{(n)}} \\ \ldots \\ or \quad \mathbf{A}_N\mathbf{x^{(n)}} \end{cases} \Leftrightarrow \begin{array}{l} \mathbf{x^{(n+1)}} \in \left\{ \mathbf{A}\mathbf{x^{(n)}} | \mathbf{A} \in \Sigma \right\} \\ \Sigma = \{\mathbf{A}_1, \ldots, \mathbf{A}_N\} \end{array}$$

$$\forall \mathbf{x^{(0)}} \quad \lim_{k\to\infty} \left\| \underbrace{\mathbf{A}_1\mathbf{A}_2\ldots\mathbf{A}_k}_{k\ \ times} \mathbf{x^{(0)}} \right\| = 0$$

for any permutation $(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k) \in Perm(\Sigma)$

Non-adaptive Stability Analysis

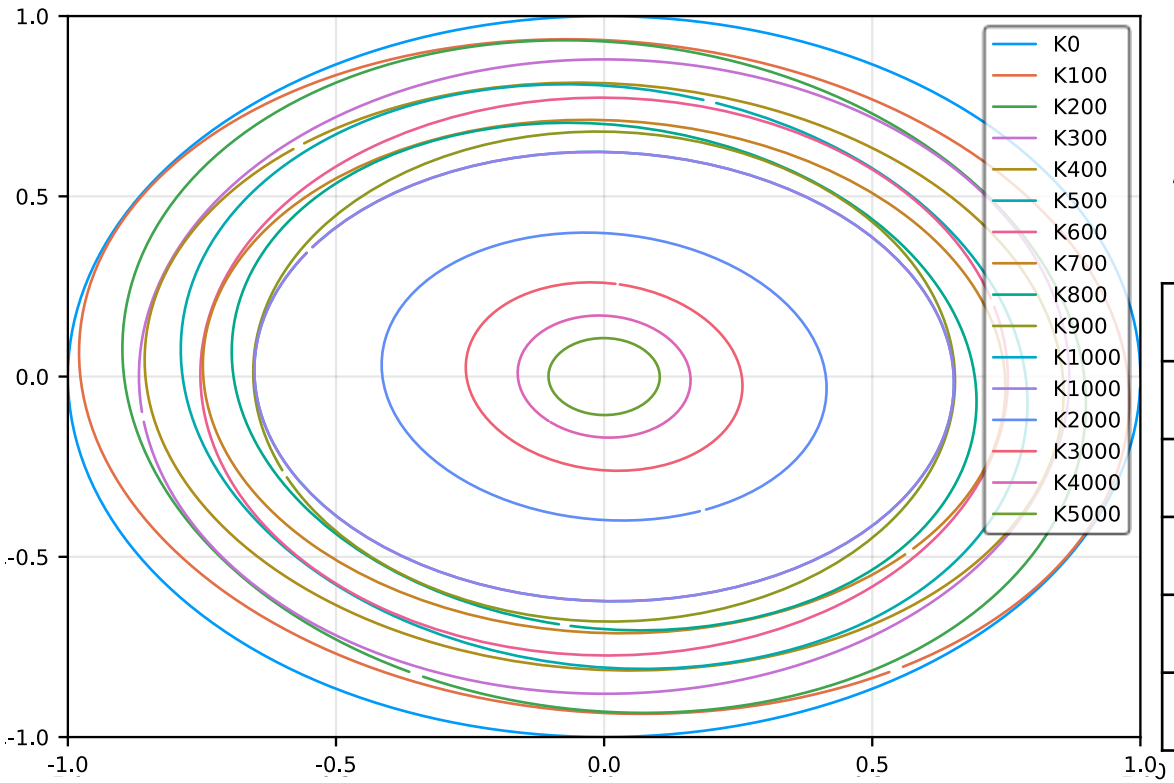
$$\forall \mathbf{x}^{(\mathbf{0})}$$

$$\mathbf{x}^{(\mathbf{n+1})} = \mathbf{A}\mathbf{x}^{(\mathbf{n})}$$

$$\lim_{k\to\infty} \left\| \underbrace{AA\ldots A}_{k\ \ times} \mathbf{x}^{(\mathbf{0})} \right\| = 0 \quad \Leftrightarrow \quad \rho\left(\mathbf{A}\right) < 1$$

$$\rho\left(\mathbf{A}\right) = \lim_{k\to\infty} \max_{\|\mathbf{x}\|=1} \left\|\mathbf{A}^k\mathbf{x}\right\|^{1/k}$$

Theys, J. (2005). *Joint Spectral Radius: theory and approximations*. Hamilton Institute (Ireland)



Upper bound:

$$\rho\left(\mathbf{A}\right) < \max_{\|\mathbf{x}\|=1} \left\|\mathbf{A}^k\mathbf{x}\right\|^{1/k}$$

k	upper bound
1	1.000050
10	1.000049
100	0.999971
1000	0.999577
10000	0.999553

Adaptive Cosim Stability Analysis

Stability: $\forall \mathbf{x}^{(\mathbf{0})}$

$$\mathbf{x}^{(\mathbf{n+1})} \in \left\{ \mathbf{A}\mathbf{x}^{(\mathbf{n})} | \mathbf{A} \in \Sigma \right\}$$

$$\lim_{k\to\infty} \left\| \underbrace{\mathbf{A}_1\mathbf{A}_2\ldots\mathbf{A}_k}_{k\ \ times} \mathbf{x}^{(\mathbf{0})} \right\| = 0$$ for any permutation $(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k) \in Perm(\Sigma)$

$$\hat{\rho}_k(\Sigma) = \sup_{(\mathbf{A}_1,\mathbf{A}_2,\ldots,\mathbf{A}_k)\in Perm(\Sigma)} \max_{\|\mathbf{x}\|=1} \|\mathbf{A}_1\mathbf{A}_2\ldots\mathbf{A}_k\mathbf{x}\|$$

$$\rho(\Sigma) = \lim_{k\to\infty} \hat{\rho}_k(\Sigma)^{1/k}$$

Stable? $\Leftrightarrow$ $\rho(\Sigma) < 1$

Upper bound can be computed: $\hat{\rho}(\Sigma) < \hat{\rho}_k(\Sigma)^{1/k}$

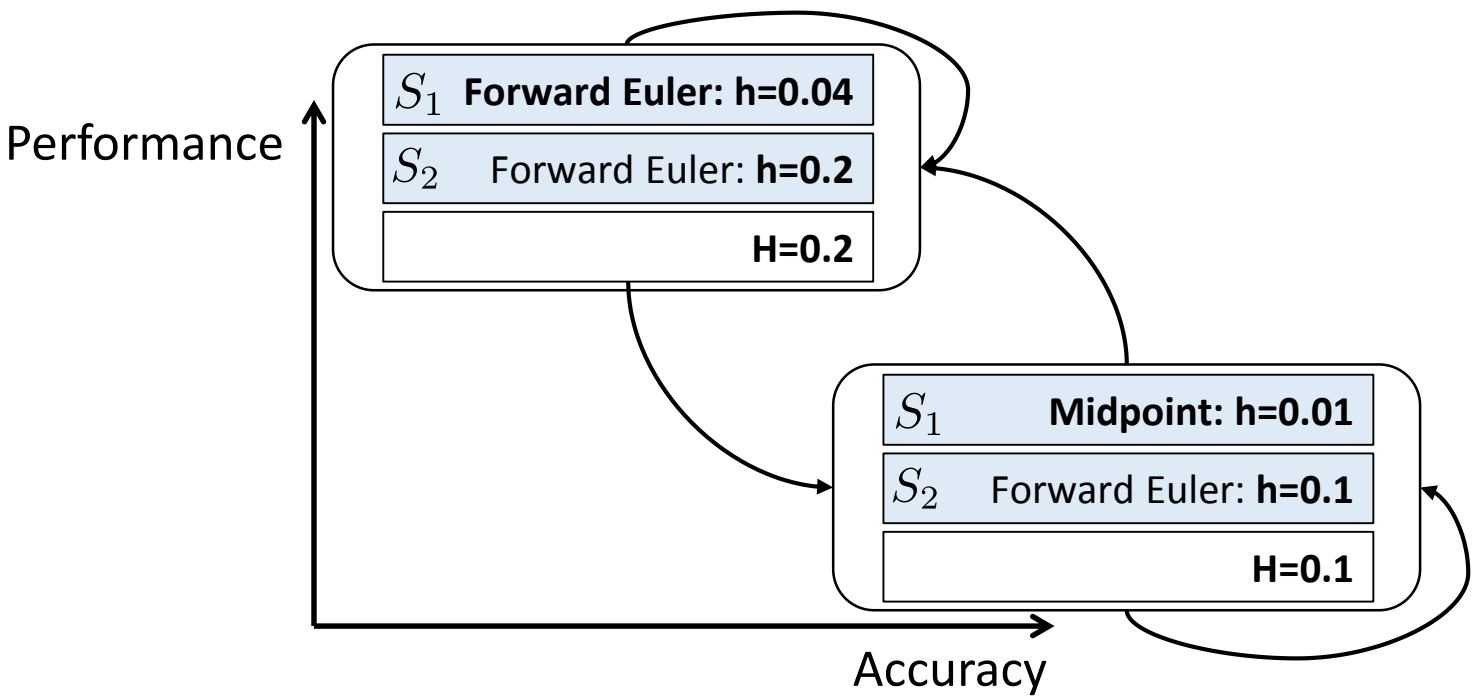
Ahmadi, A. A., Jungers, R. M., Parrilo, P. A., & Roozbehani, M. (2014). Joint Spectral Radius and Path-Complete Graph Lyapunov Functions. *SIAM Journal on Control and Optimization*, *52*(1), 687–717. http://doi.org/10.1137/110855272

Parrilo, P. A., & Jadbabaie, A. (2008). Approximation of the joint spectral radius using sum of squares. *Linear Algebra and Its Applications*, *428*(10), 2385–2402. http://doi.org/10.1016/j.laa.2007.12.027

Jungers, R. (2009). *The joint spectral radius: theory and applications* (Vol. 385). Springer Science & Business Media.

Example

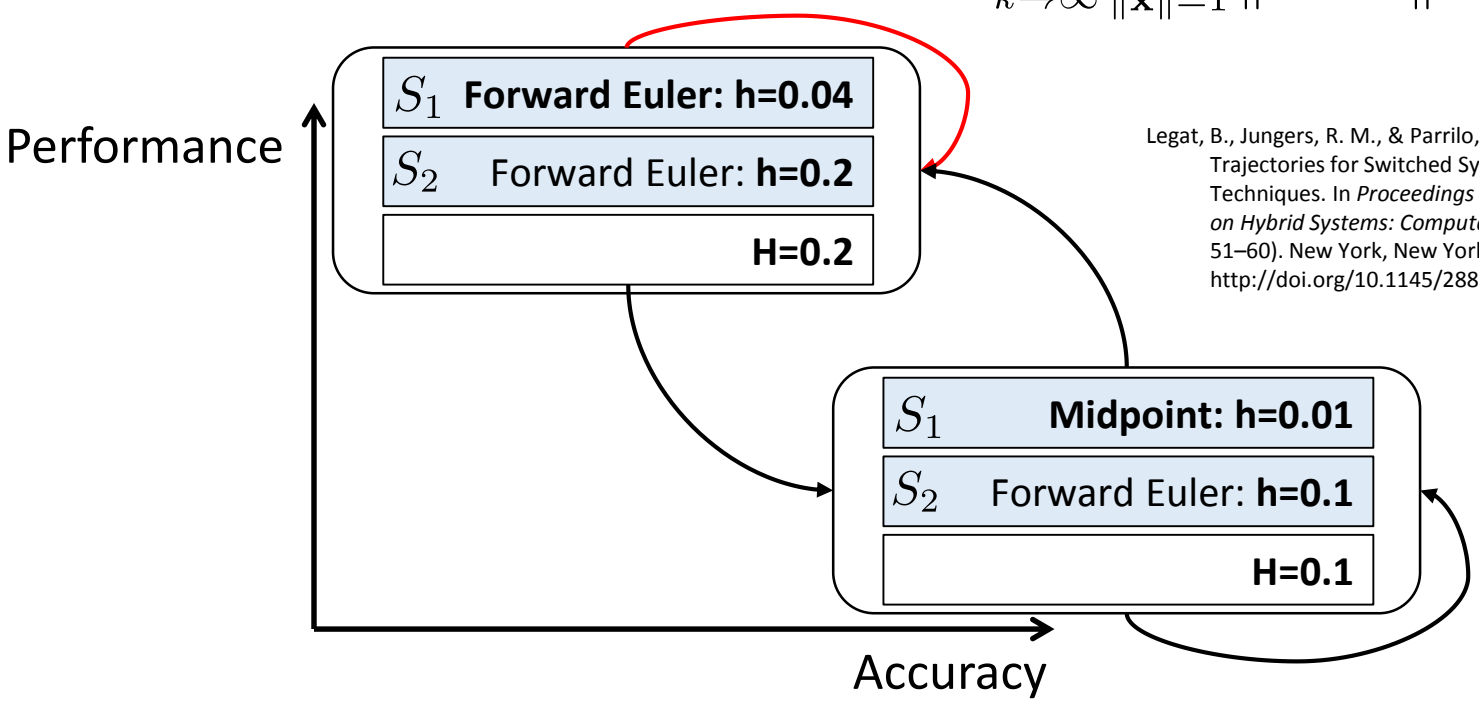
$$\Sigma = \left\{ \underbrace{\mathbf{A}_{H=0.2,S_1=\langle FE,0.04\rangle,S_2=\langle FE,0.2\rangle}}_{\mathbf{A}_{fe}}, \underbrace{\mathbf{A}_{H=0.1,S_1=\langle Mid,0.01\rangle,S_2=\langle FE,0.1\rangle}}_{\mathbf{A}_{mid}} \right\}$$



Example

$$\Sigma = \left\{ \underbrace{\mathbf{A}_{H=0.2,S_1=\langle FE,0.04\rangle,S_2=\langle FE,0.2\rangle}}_{\mathbf{A}_{fe}}, \underbrace{\mathbf{A}_{H=0.1,S_1=\langle Mid,0.01\rangle,S_2=\langle FE,0.1\rangle}}_{\mathbf{A}_{mid}} \right\}$$

$$\lim_{k\to\infty} \max_{\|\mathbf{x}\|=1} \left\| {\mathbf{A}_{fe}}^k \mathbf{x} \right\|^{1/k} = \rho\left(\mathbf{A}_{fe}\right) > 1$$



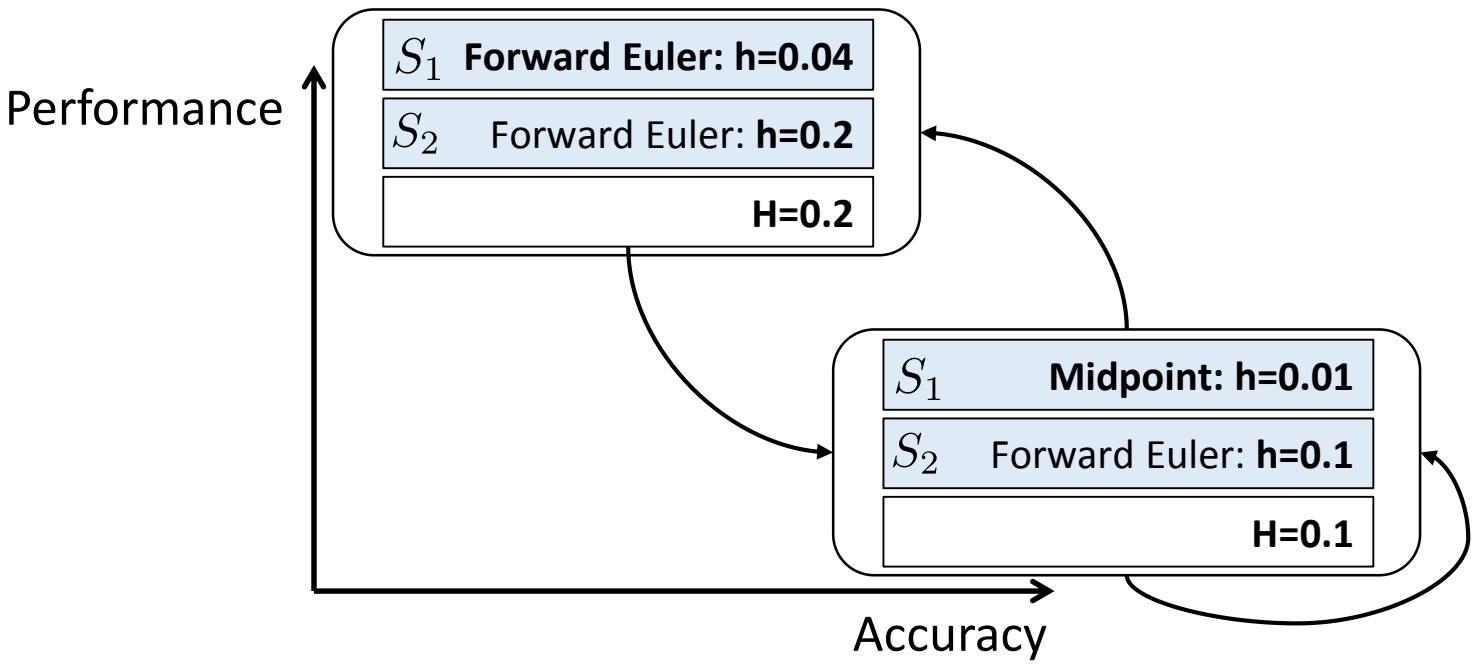
Legat, B., Jungers, R. M., & Parrilo, P. A. (2016). Generating Unstable Trajectories for Switched Systems via Dual Sum-Of-Squares Techniques. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control - HSCC '16* (pp. 51–60). New York, New York, USA: ACM Press. http://doi.org/10.1145/2883817.2883821

Example

$$\Sigma = \{\mathbf{A}_{mid}, \mathbf{A}_{fe}\mathbf{A}_{mid}\}$$

$$\rho(\Sigma) < 0.99986$$

https://github.com/blegat/SwitchOnSafety.jl



Summary of the Approach

1. Capture possible orchestration decisions
   - IP can be protected if solver is embedded

$$\begin{bmatrix} x_1^{(n+1)} \\ v_1^{(n+1)} \end{bmatrix} = A_1^{k_1} \begin{bmatrix} x_1^{(n)} \\ v_1^{(n)} \end{bmatrix} + \left(\sum_{m=0}^{k_1-1} A_1^m B_1 \right) u_1^{(n)}$$

$$y_1^{(n)} = \begin{bmatrix} x_1^{(n)} \\ v_1^{(n)} \end{bmatrix}$$

$$\begin{bmatrix} x_2^{(n+1)} \\ v_2^{(n+1)} \end{bmatrix} = A_2^{k_2} \begin{bmatrix} x_2^{(n)} \\ v_2^{(n)} \end{bmatrix} + \left(\sum_{m=0}^{k_2-1} A_2^m B_2 \right) u_2^{(n)}$$

$$y_2^{(n)} = \begin{bmatrix} c_k & d_k \end{bmatrix} \begin{bmatrix} x_2^{(n)} \\ v_2^{(n)} \end{bmatrix} + \begin{bmatrix} -c_k & -d_k \end{bmatrix} u_2^{(n)}$$

$$\begin{bmatrix} x_1^{(n+1)} \\ v_1^{(n+1)} \\ x_2^{(n+1)} \\ v_2^{(n+1)} \end{bmatrix} = \mathbf{A}_{H=0.2, S_2=\langle FE, 0.04\rangle, S_2=\langle FE, 0.2\rangle} \begin{bmatrix} x_1^{(n)} \\ v_1^{(n)} \\ x_2^{(n)} \\ v_2^{(n)} \end{bmatrix}$$

$$\Sigma = \{\mathbf{A}_{mid}, \mathbf{A}_{fe}\}$$

Summary of the Approach

1. Capture possible orchestration decisions
   - IP can be protected if solver is embedded
2. Check stability of unrestricted adaptive orchestration

$$\Sigma = \{\mathbf{A}_{mid}, \mathbf{A}_{fe}\}$$
$$\rho(\Sigma) < 1$$

Summary of the Approach

1. Capture possible orchestration decisions
   - IP can be protected if solver is embedded
2. Check stability of unrestricted adaptive orchestration
3. Restrict if needed

$$\Sigma = \{\mathbf{A}_{mid}, \mathbf{A}_{fe}\}$$

$$\Downarrow$$

$$\Sigma = \{\mathbf{A}_{mid}, \mathbf{A}_{fe}\mathbf{A}_{mid}\}$$

Limitations

- Scalability
- How to efficiently restrict the orchestration policy?

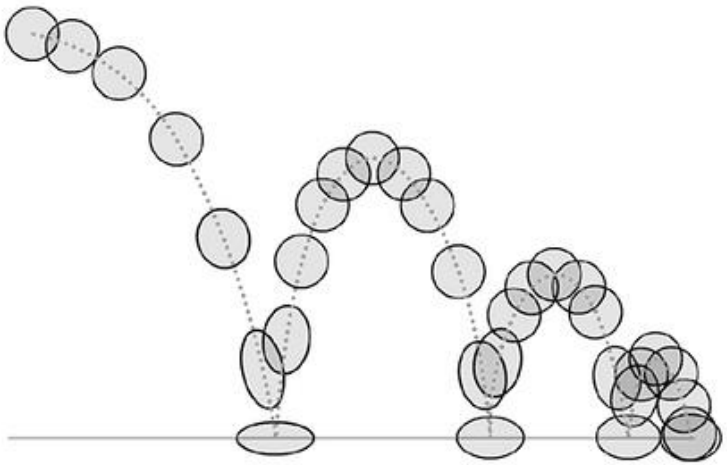
The Baby Steps: Stability of Hybrid Co-simulation

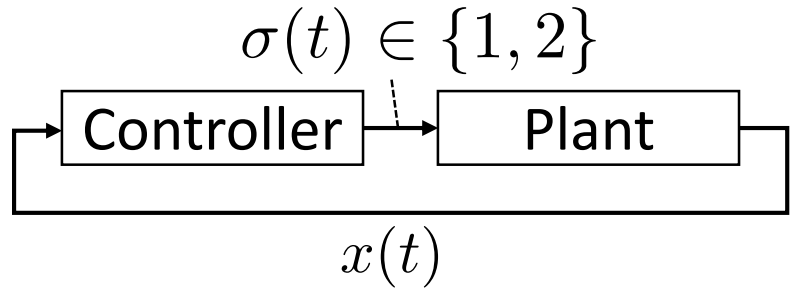
Our contributions

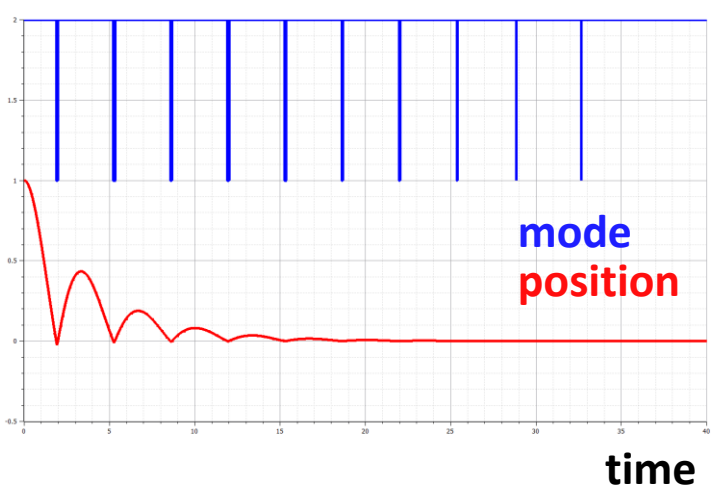
Paschalis Karalis, Eva Navarro-López, Hans Vangheluwe

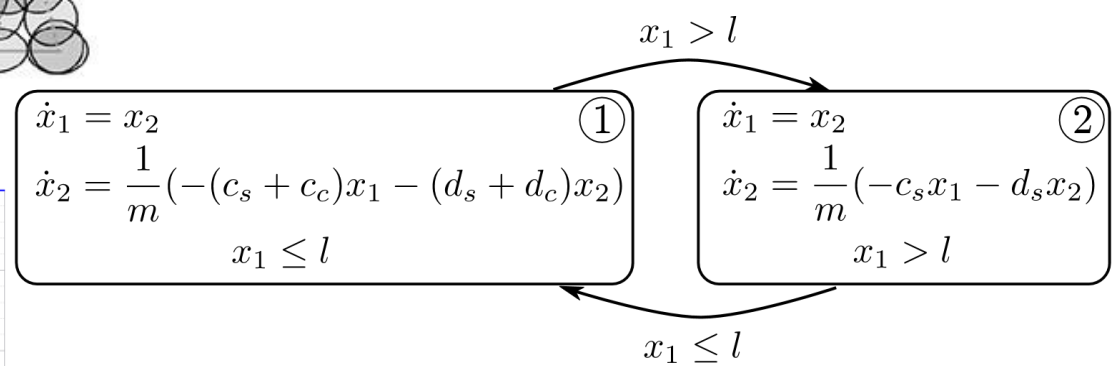
(2017). Approximated Stability Analysis of Bi-Modal Hybrid Co-simulation Scenarios. In *1st Workshop on Formal Co-Simulation of Cyber-Physical Systems* (to appear). Trento, Italy.

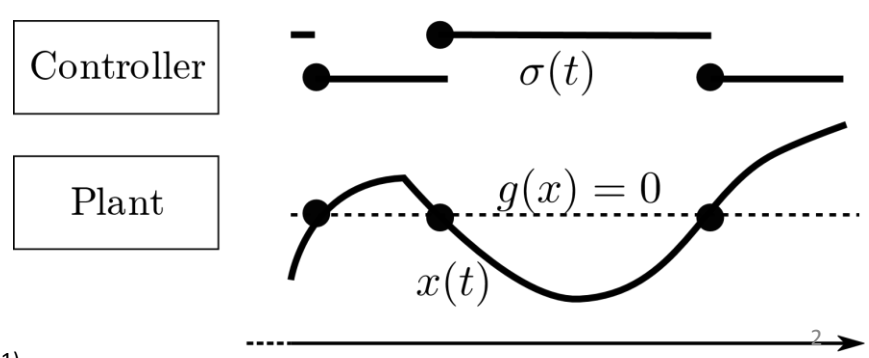
Running Example: Bouncing Ball





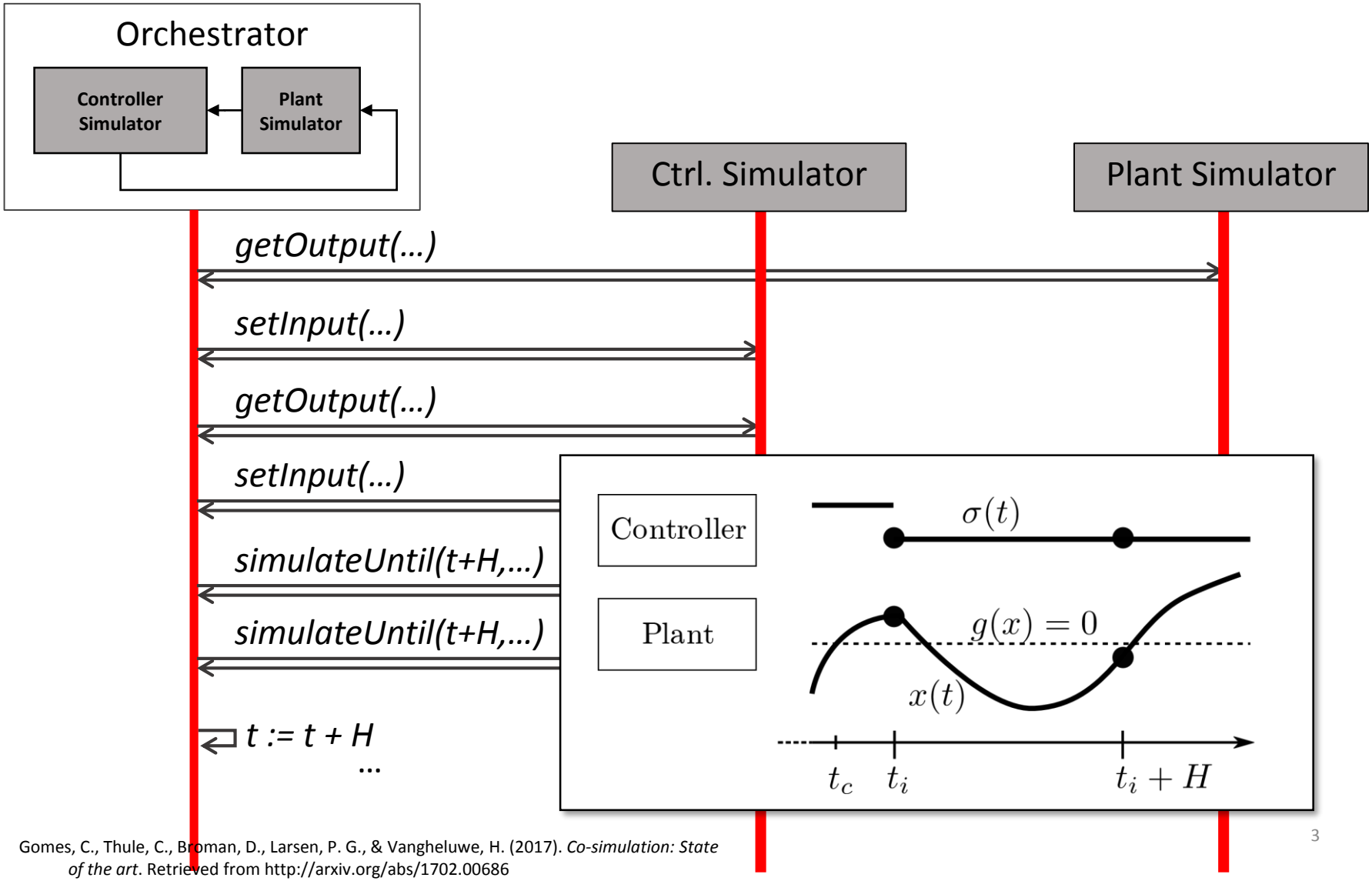






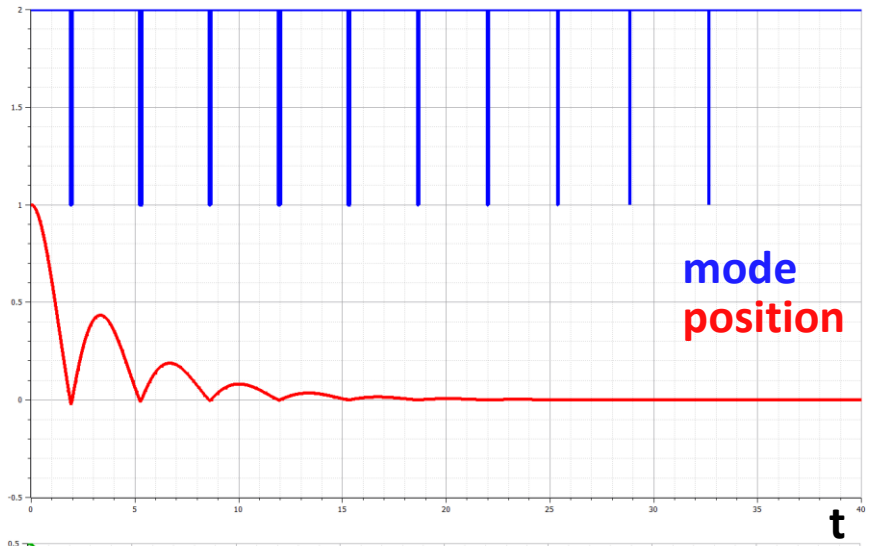
Navarro-López, E. M., & Carter, R. (2011). Hybrid automata: an insight into the discrete abstraction of discontinuous systems. *International Journal of Systems Science*, *42*(11), 1883–1898. http://doi.org/10.1080/00207721.2010.495189

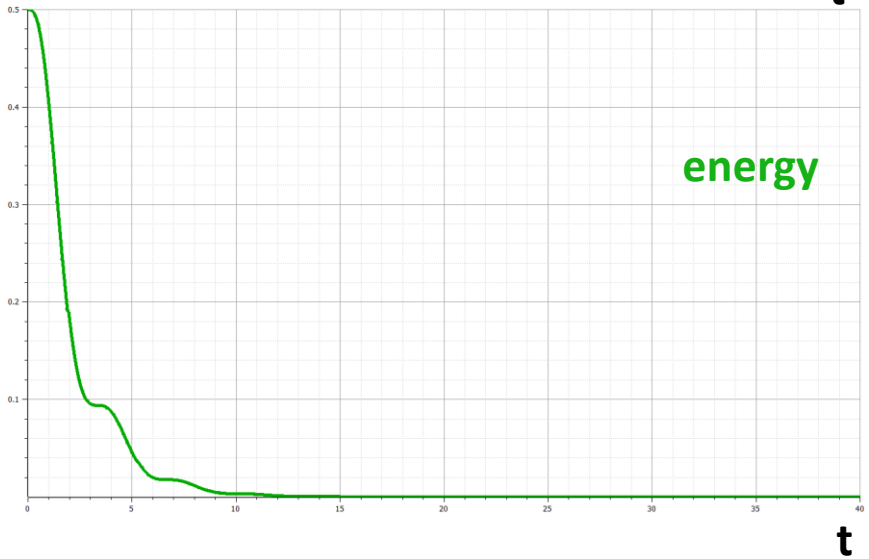
Bouncing ball Co-simulation



Gomes, C., Thule, C., Broman, D., Larsen, P. G., & Vangheluwe, H. (2017). *Co-simulation: State of the art*. Retrieved from http://arxiv.org/abs/1702.00686

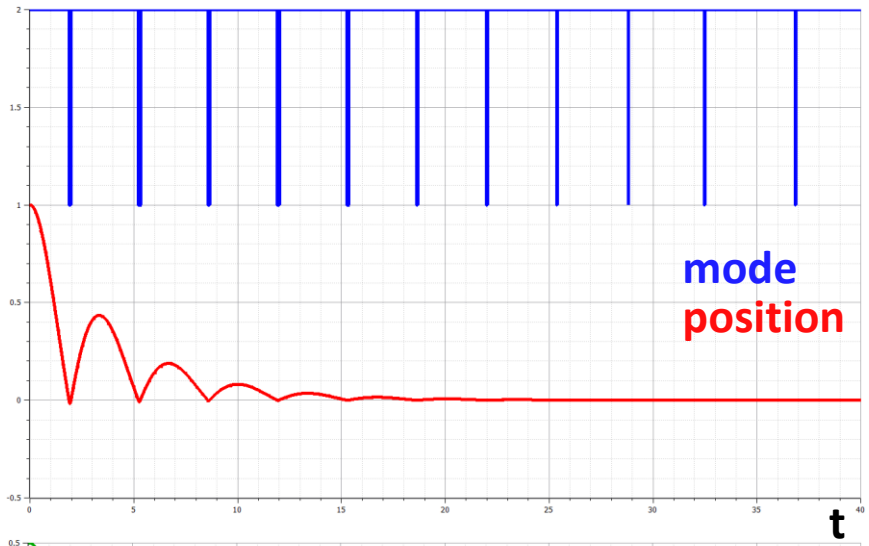
The Challenge: Energy Conservation

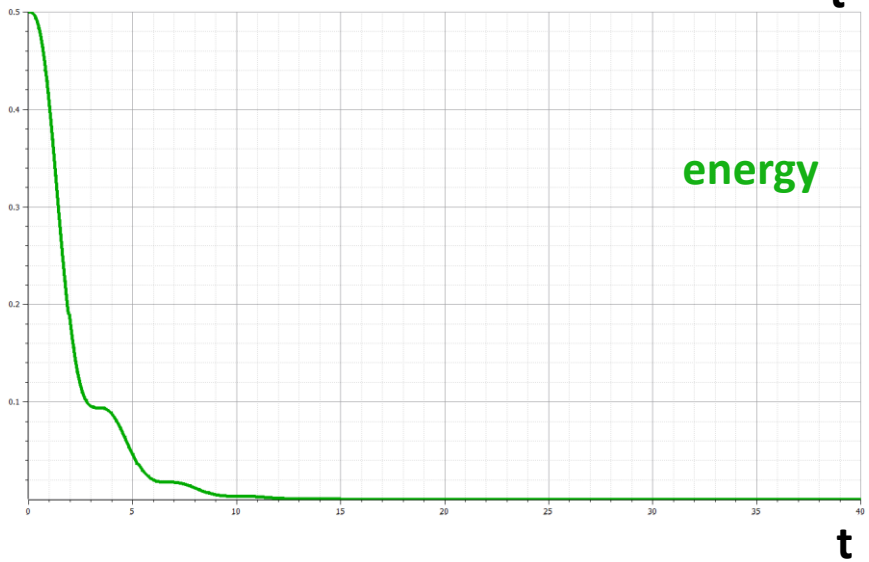




H=0

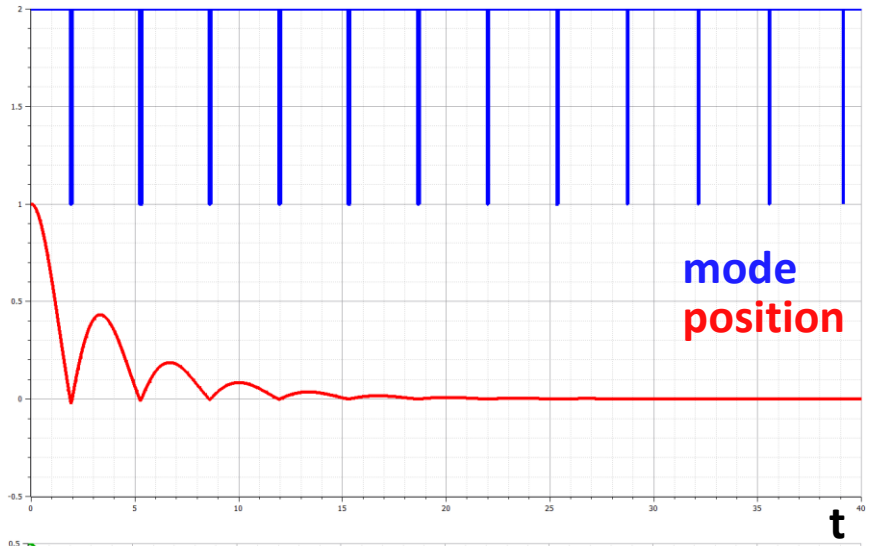
The Challenge: Energy Conservation

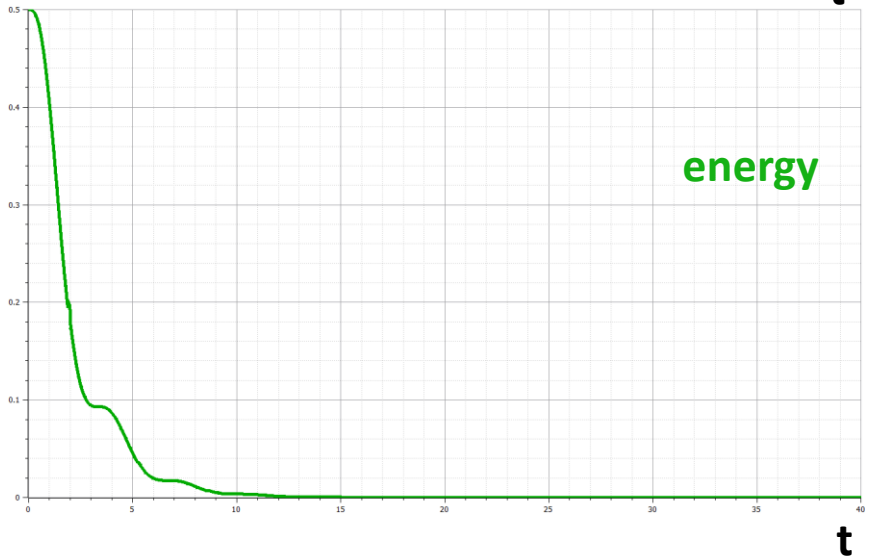




H=0.001

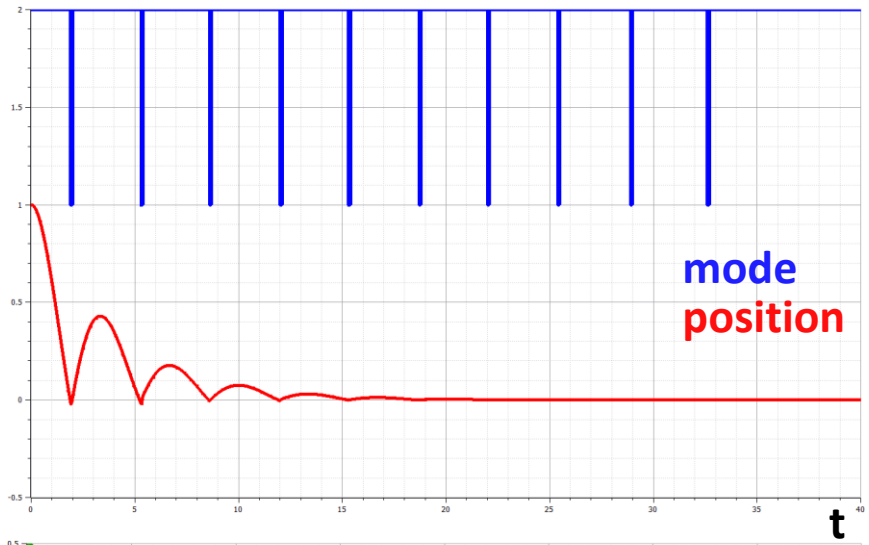
The Challenge: Energy Conservation

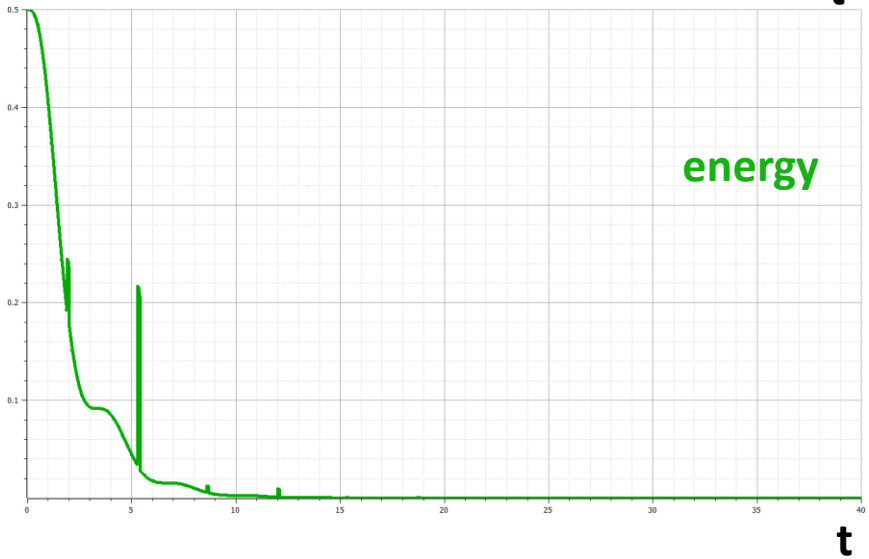




H=0.01

The Challenge: Energy Conservation

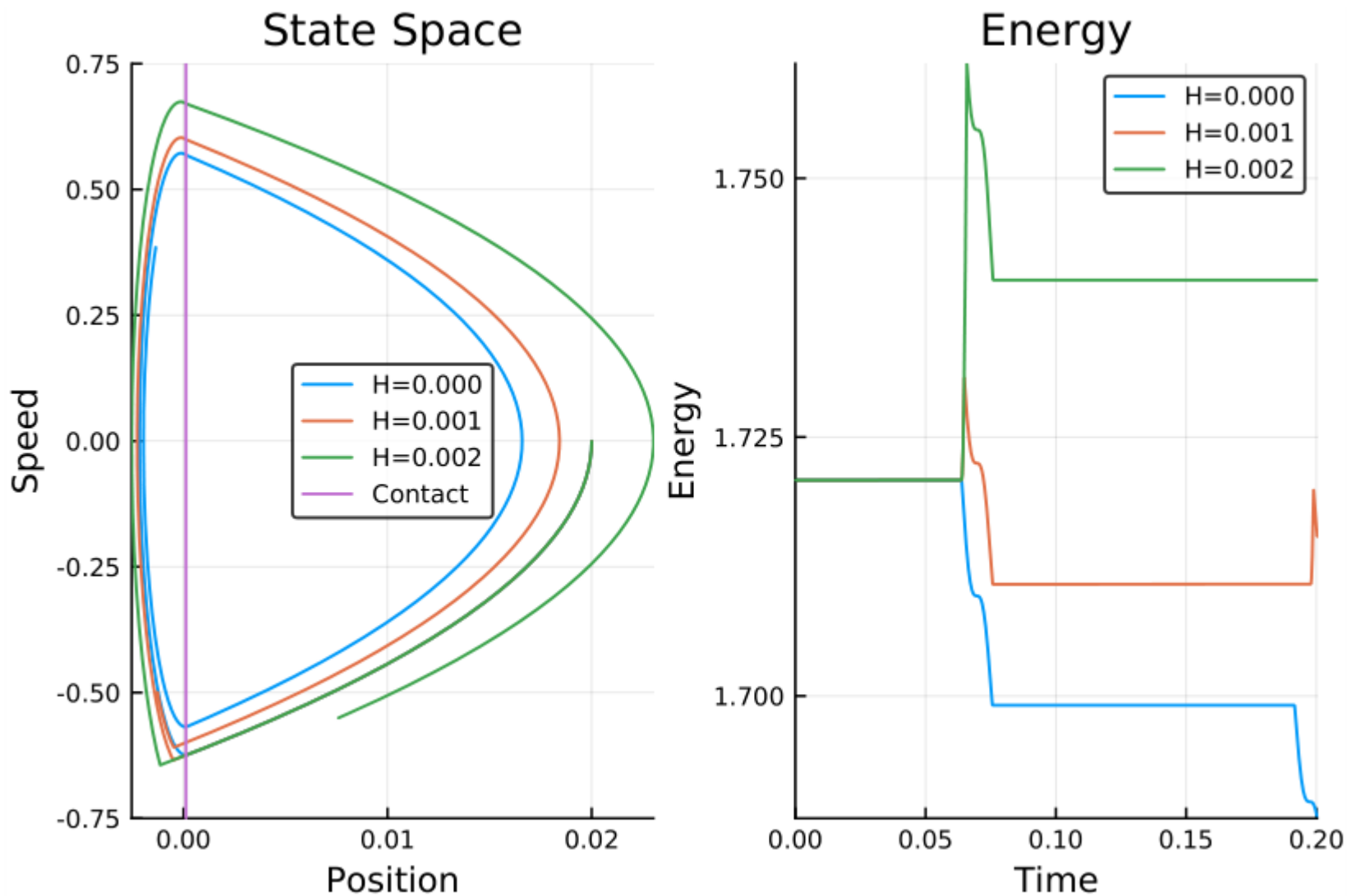




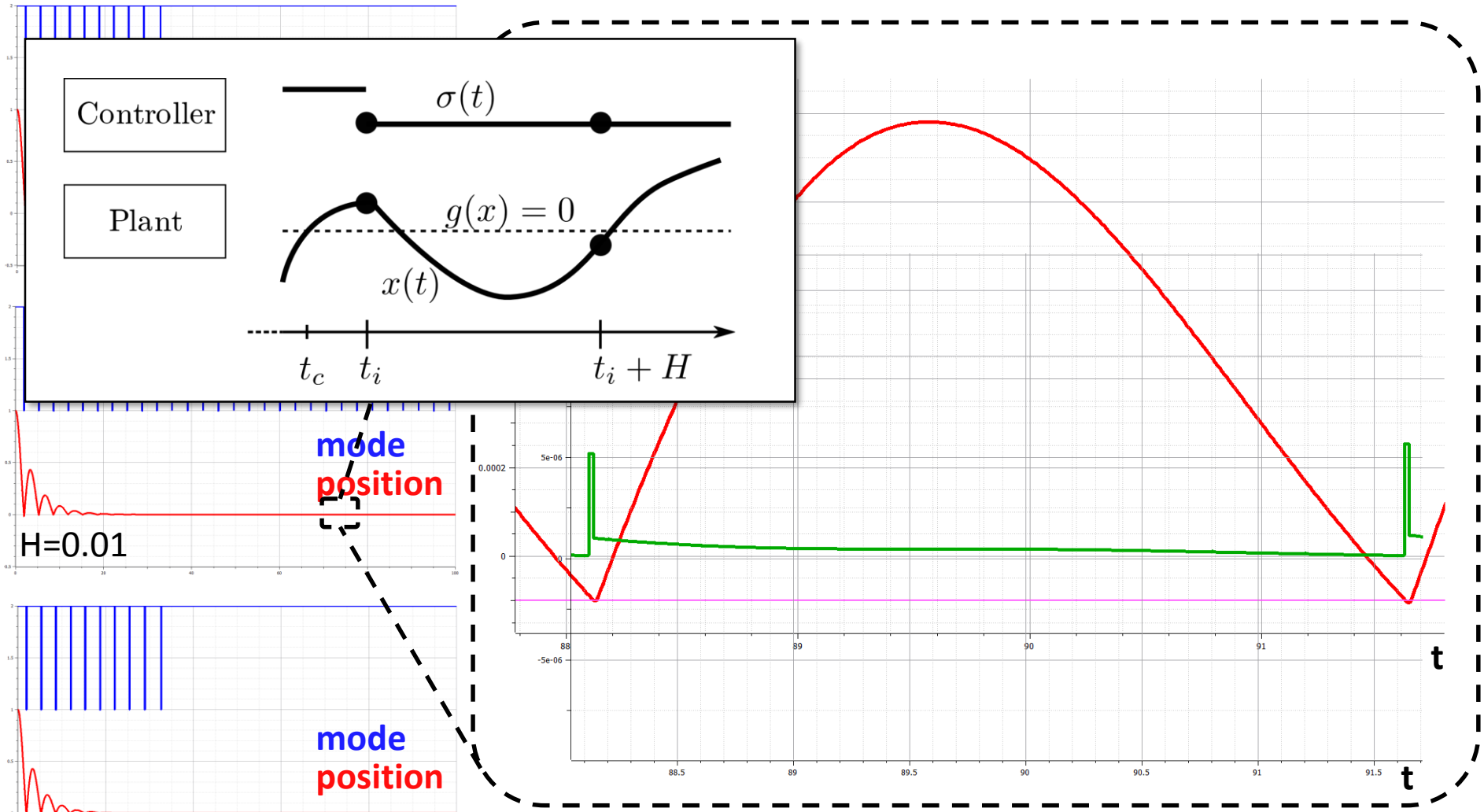
H=0.1

Benedikt, M., Watzenig, D., Zehetner, J., & Hofer, A. (2013). NEPCE-A Nearly Energy Preserving Coupling Element for Weak-coupled Problems and Co-simulation. In *IV International Conference on Computational Methods for Coupled Problems in Science and Engineering, Coupled Problems*.

The Challenge: Energy Conservation

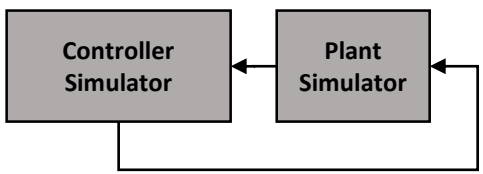


The Challenge: Energy Conservation

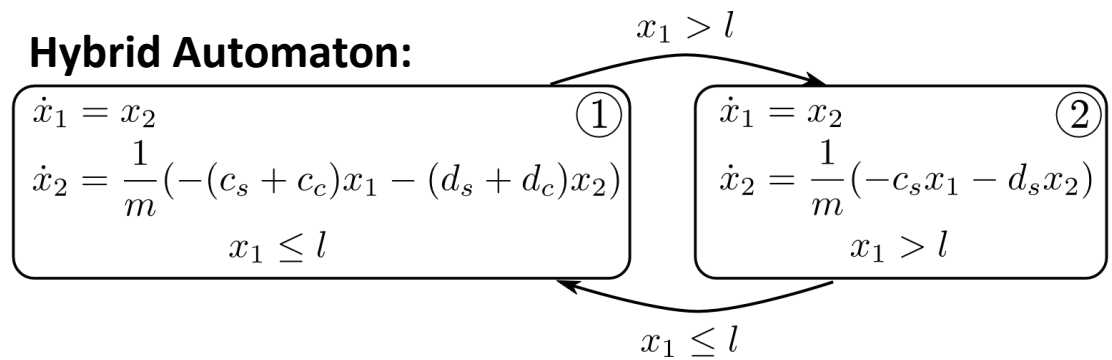


Outline of the Approach

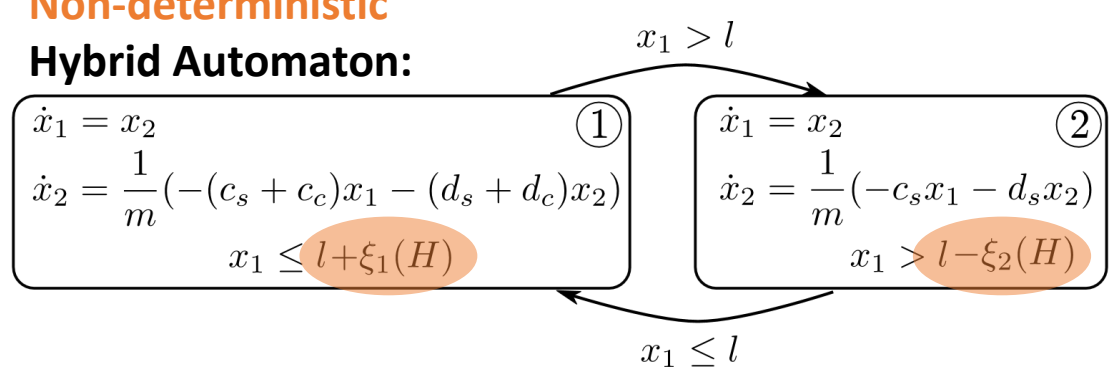
Original System:



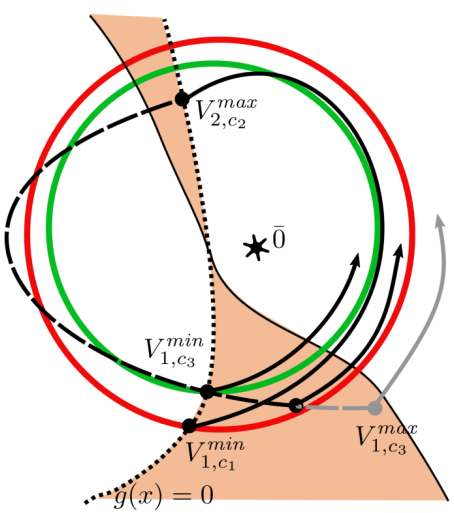
Hybrid Automaton:



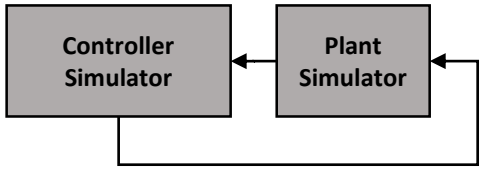
Non-deterministic Hybrid Automaton:

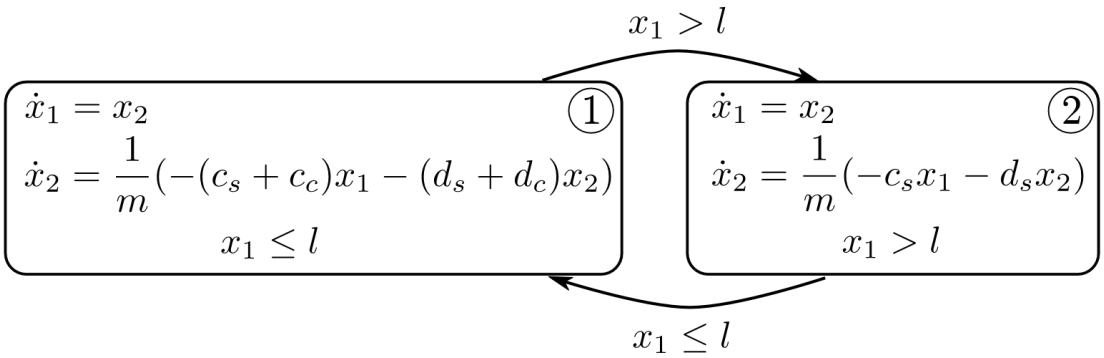


Stability range computation:



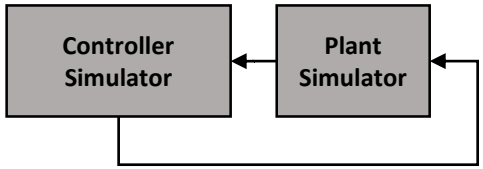
Hybrid Automata Creation

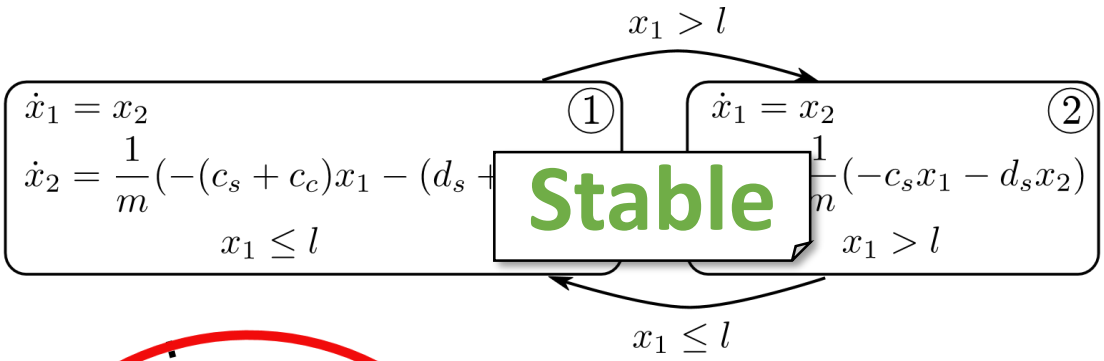


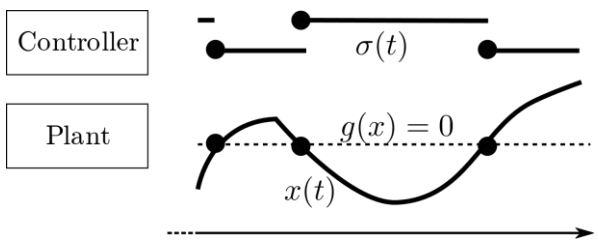


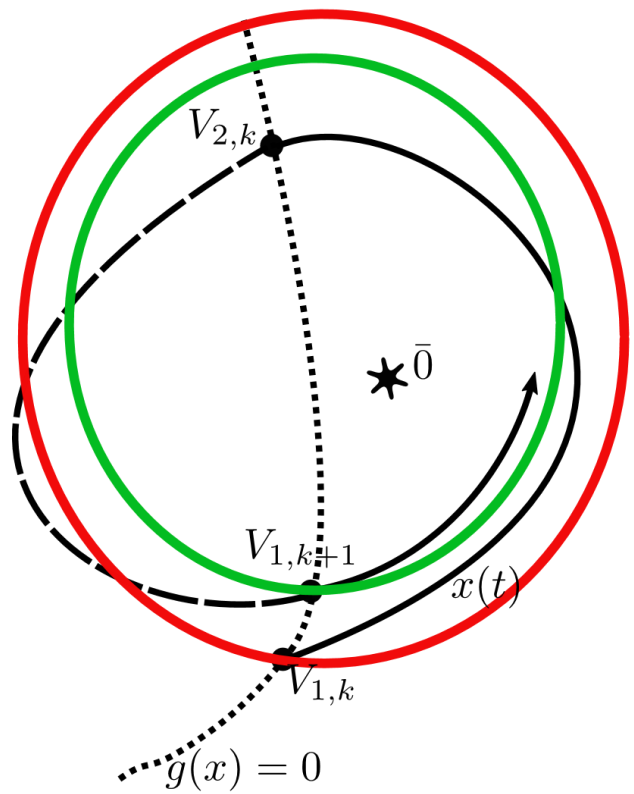
- Limited to bi-modal scenarios
- Requires access to each mode's dynamics

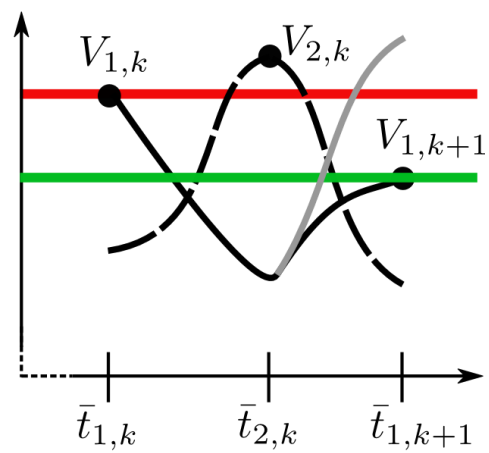
Hybrid Automata Creation





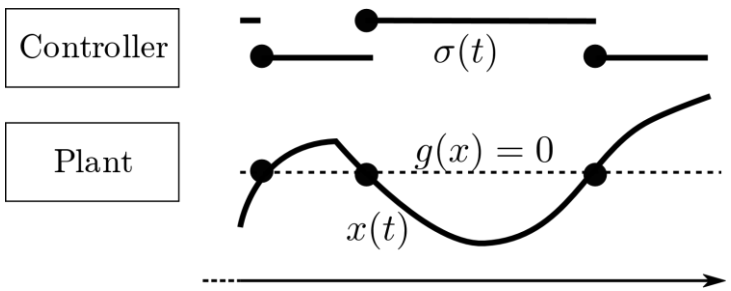




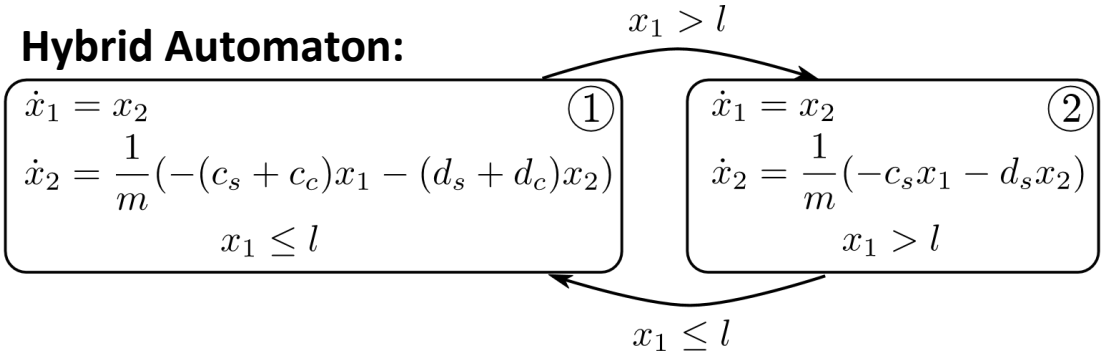


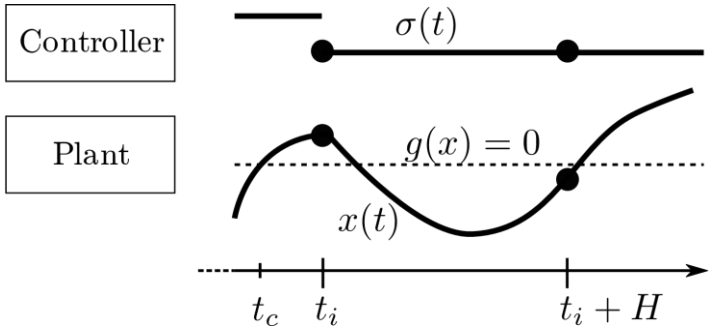
Branicky, M. S. (1998). Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, *43*(4), 475–482. http://doi.org/10.1109/9.664150

Invariant Relaxation

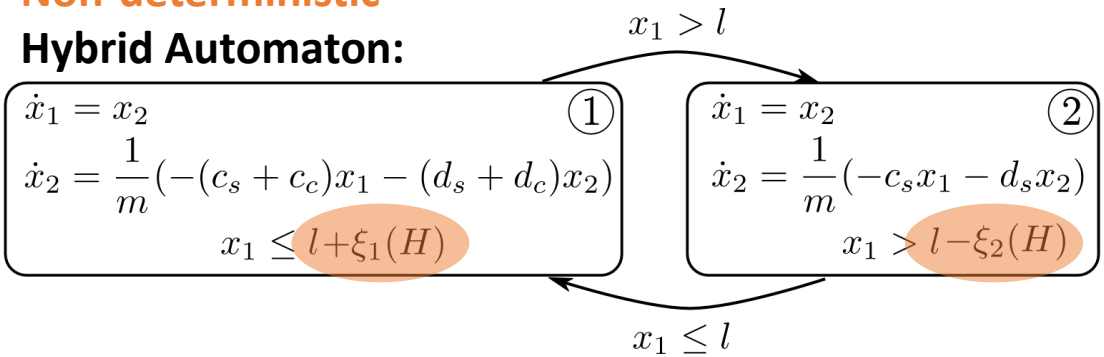


Hybrid Automaton:





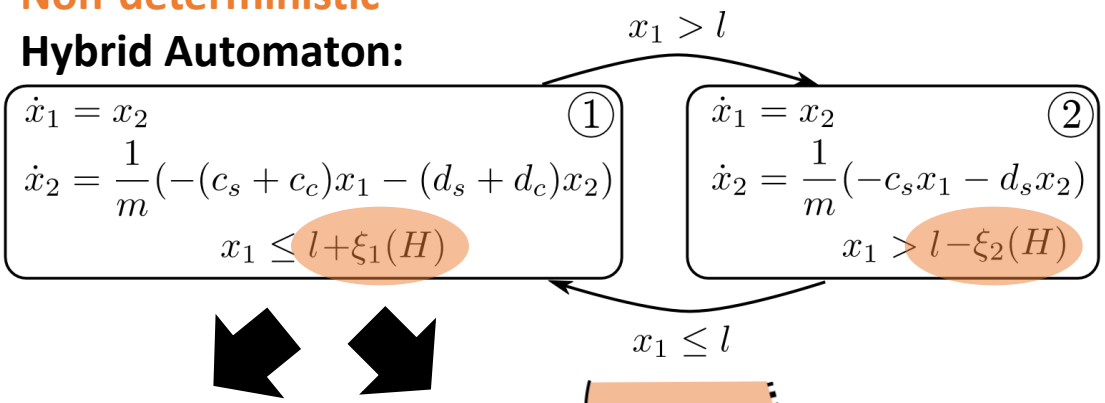
Non-deterministic
Hybrid Automaton:

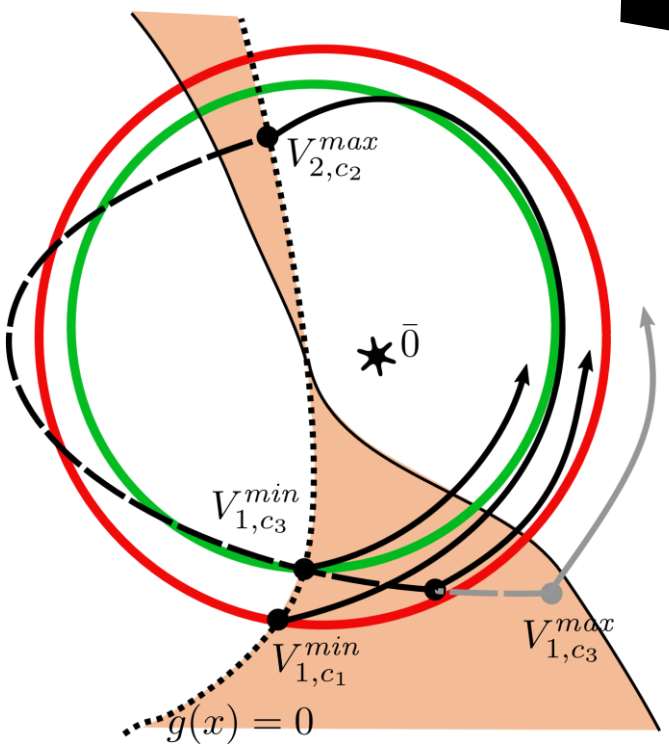


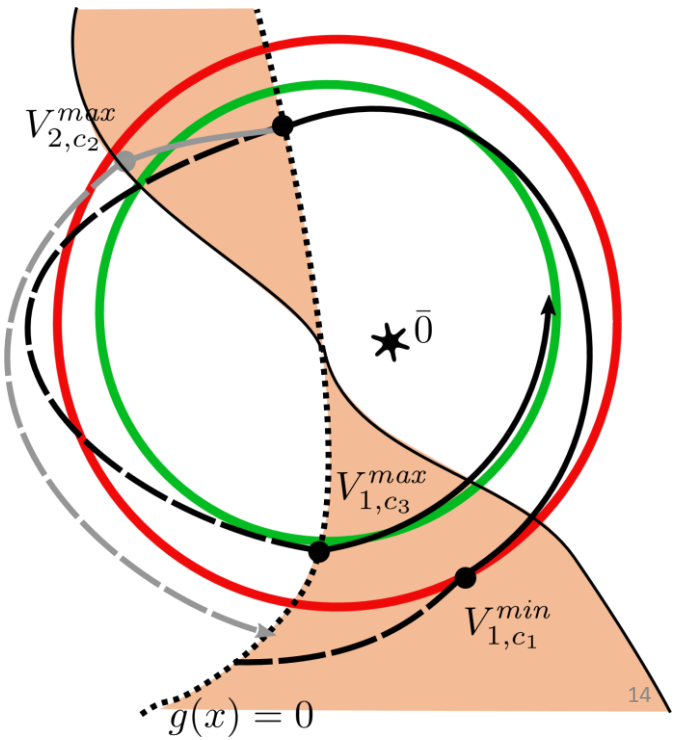
Approximate Stability Analysis

Non-deterministic

Hybrid Automaton:

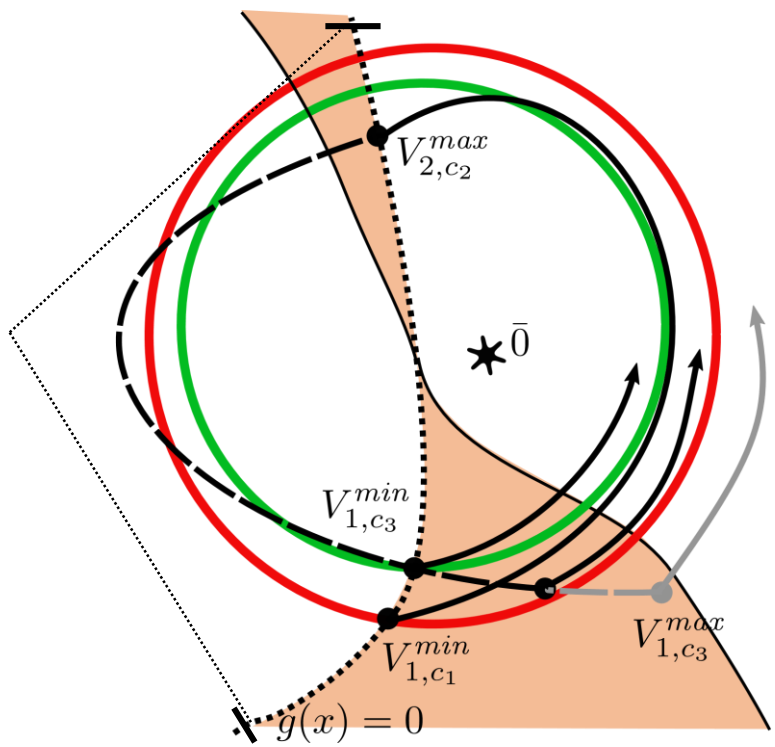






Algorithm

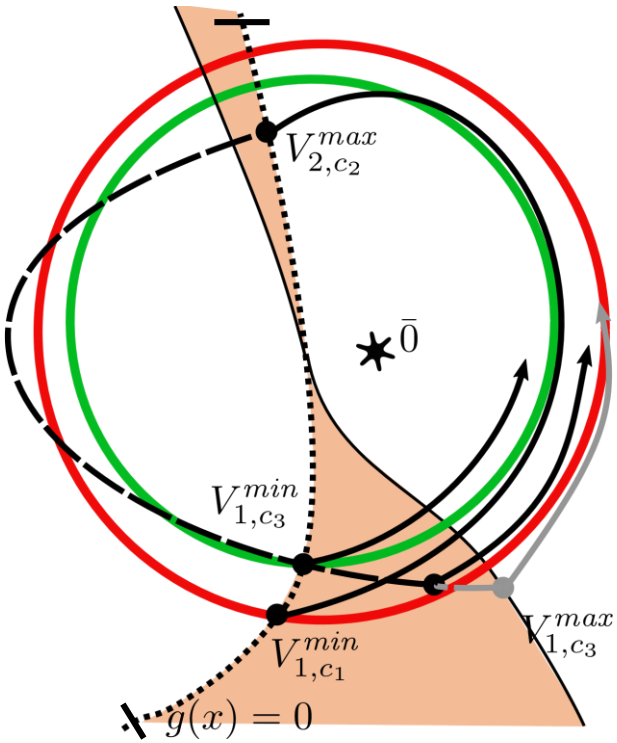
Range of initial conditions given



Initial maximum step size:

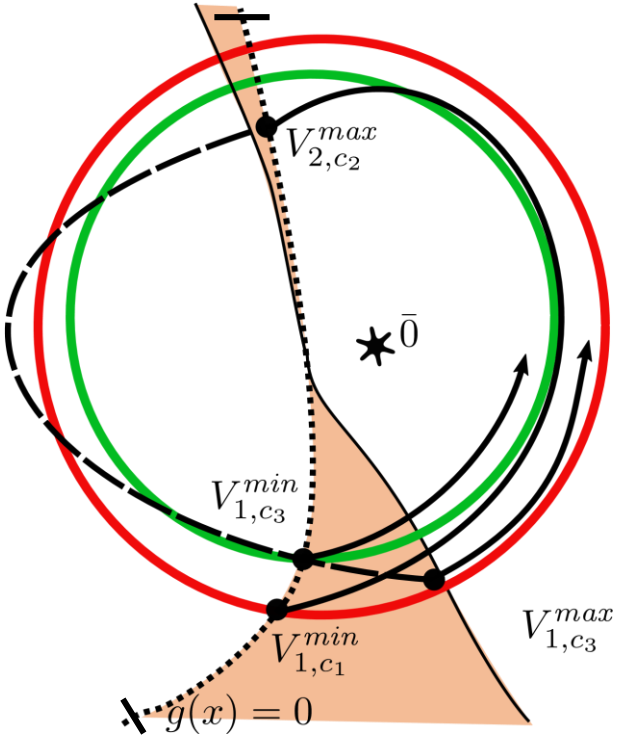
H=0.1

Algorithm



H=0.01

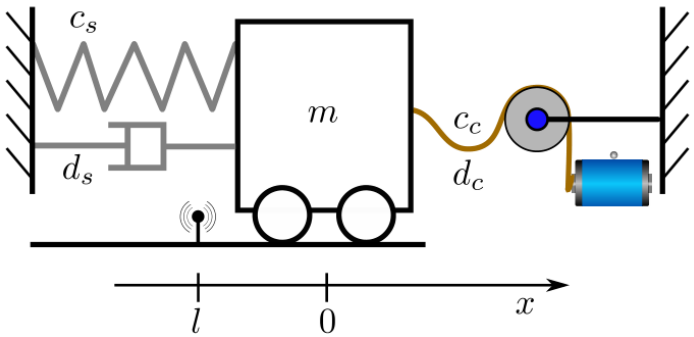
Algorithm

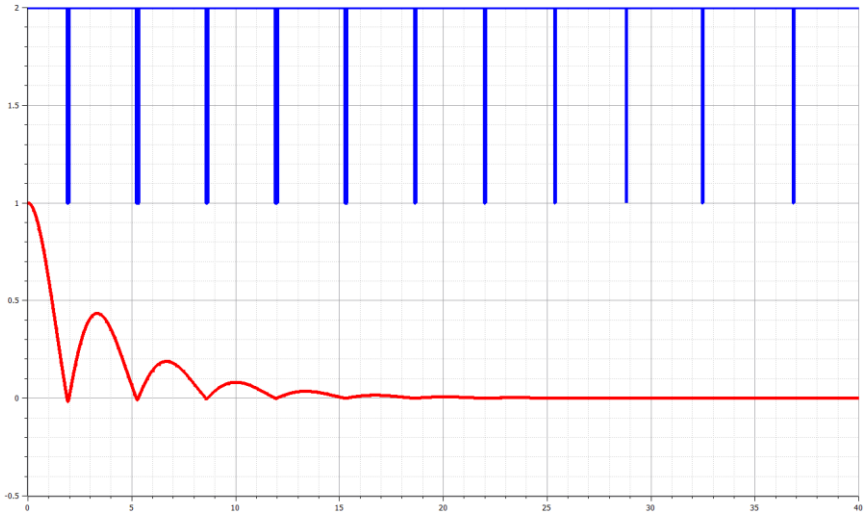


H=0.001

Results – Cart Example

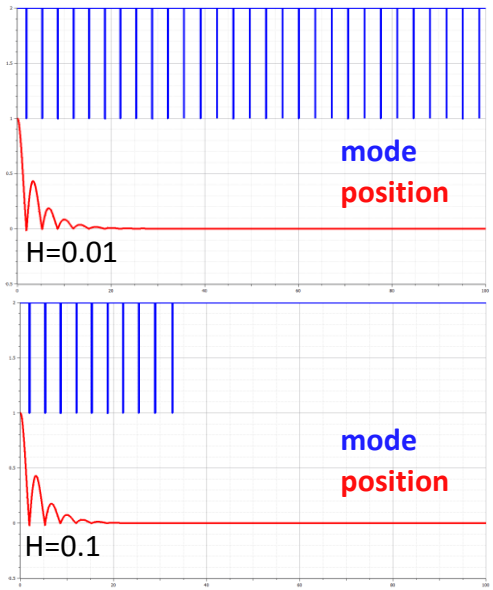
- Mode 1: H < 0.002232
- Mode 2: H < 0.002204
- Safest H = 0.002204

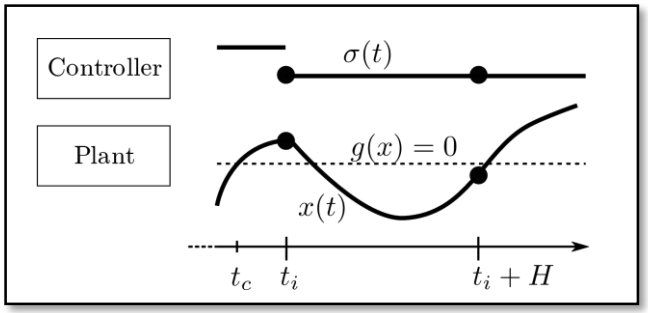




Conclusion & Future Work

- Conservative algorithm.
- Restricted to bi-modal planar hybrid systems.
- Restricted to bounded initial conditions.
- Requires access to state and derivatives.
- Jacobi orchestration only.
- Use stronger stability theorems





Navarro-López, E. M., & Laila, D. S. (2013). Group and Total Dissipativity and Stability of Multi-Equilibria Hybrid Automata. *IEEE Transactions on Automatic Control*, *58*(12), 3196–3202. http://doi.org/10.1109/TAC.2013.2261185