Towards the Verification of Hybrid Co-simulation Algorithms

Casper Thule, **Cláudio Gomes**, Julien Deantoni,

Peter G. Larsen, Jörg Brauer, and Hans Vangheluwe

MPM4CPS

fwo

Ansymo
Antwerp Systems & Software Modelling
University of Antwerp

Verified

FLANDERS MAKE
MANUFACTURING INNOVATION NETWORK

Nexor
Cyber-Physical Systems
University of Antwerp

Universiteit Antwerpen

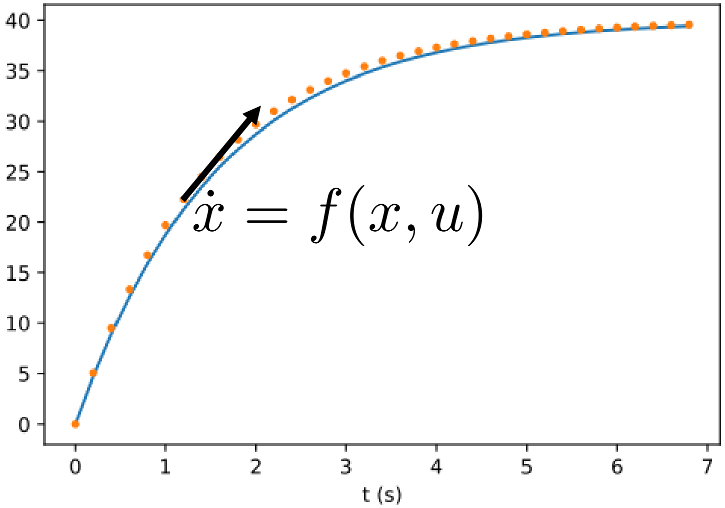
AARHUS UNIVERSITY

INTO-CPS

POLYTECH
NICE-SOPHIA

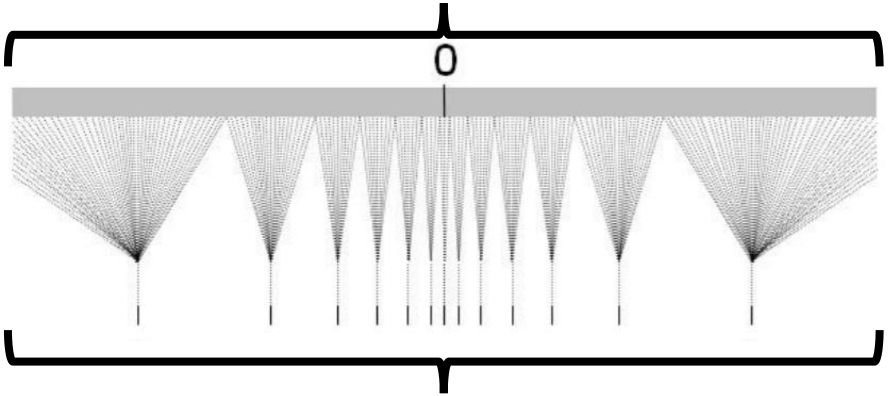
Sources of Errors in Co-simulation

Solver Approximation



$$\dot{x} = f(x, u)$$

Real Numbers



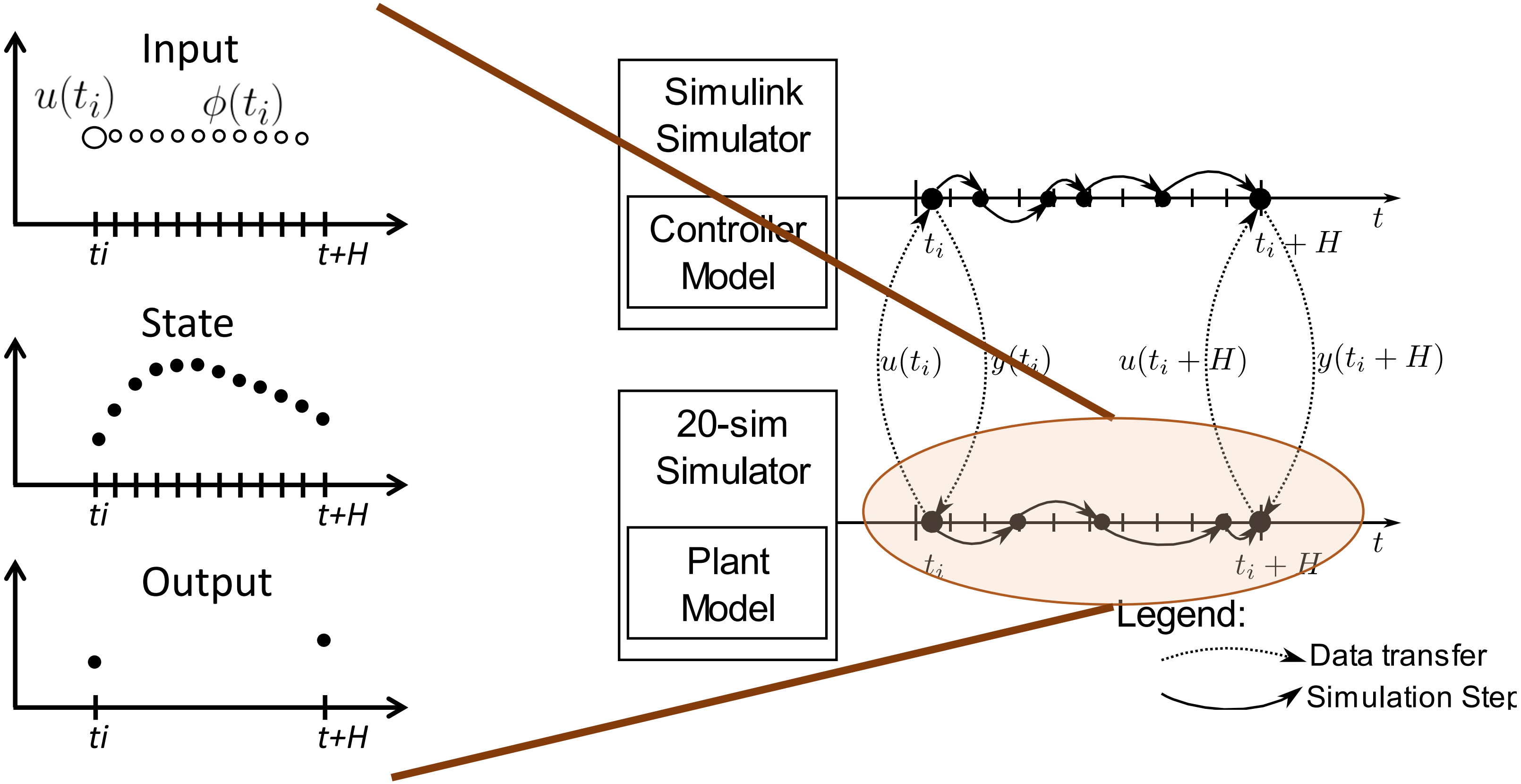
www.cs.cmu.edu

Floating-Point Numbers

Finite Representation

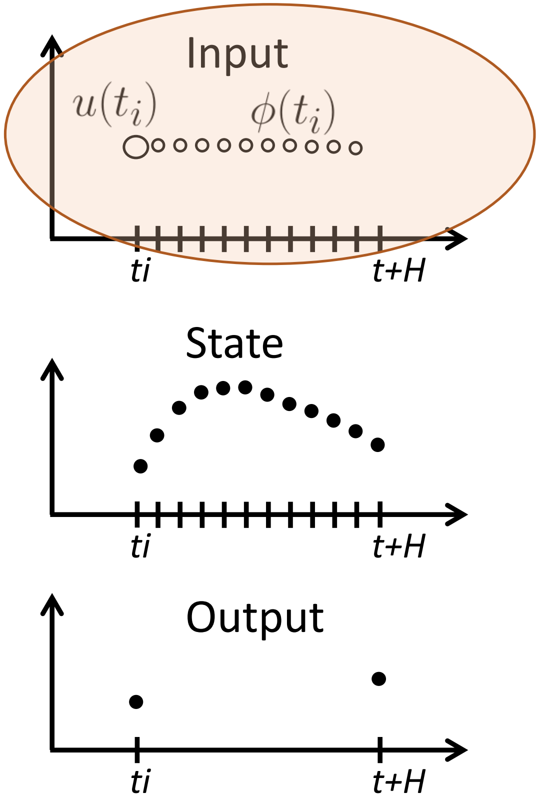
Sources of Errors in Co-simulation

Input Approximation



Sources of Errors in Co-simulation

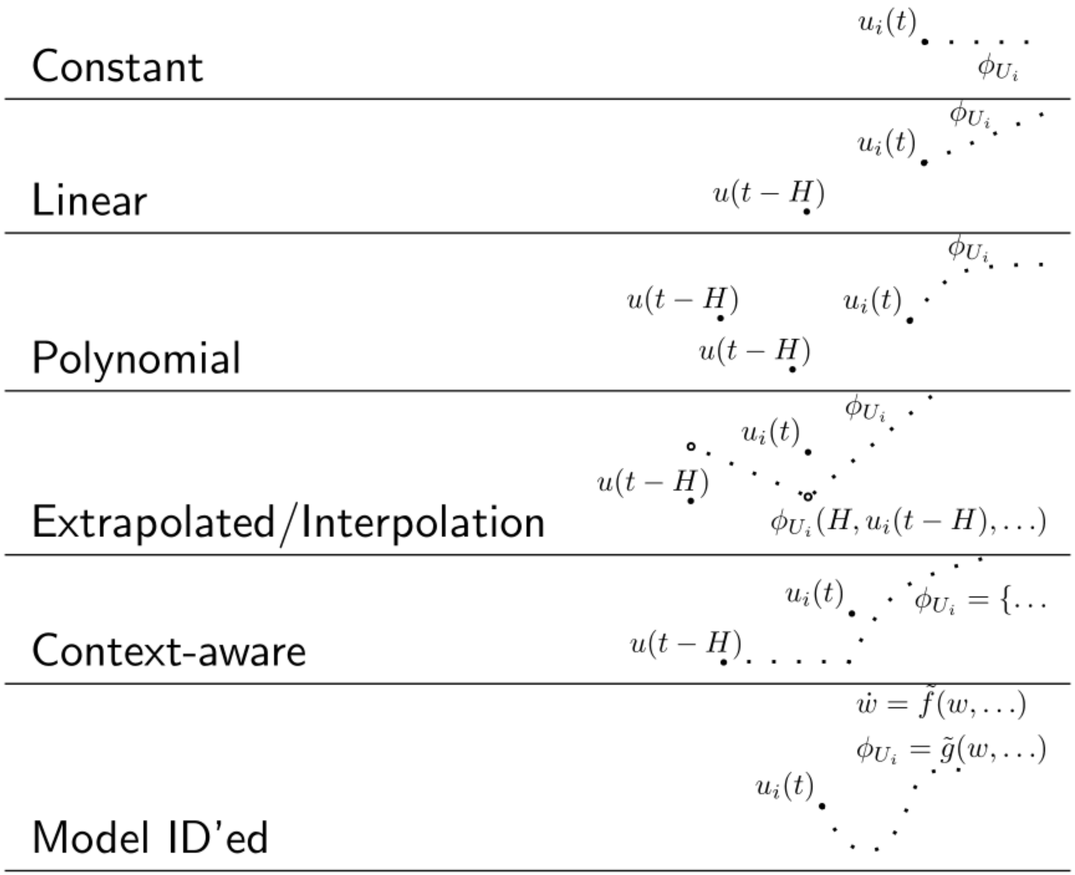
Input Approximation



Input

State

Output



Constant

Linear

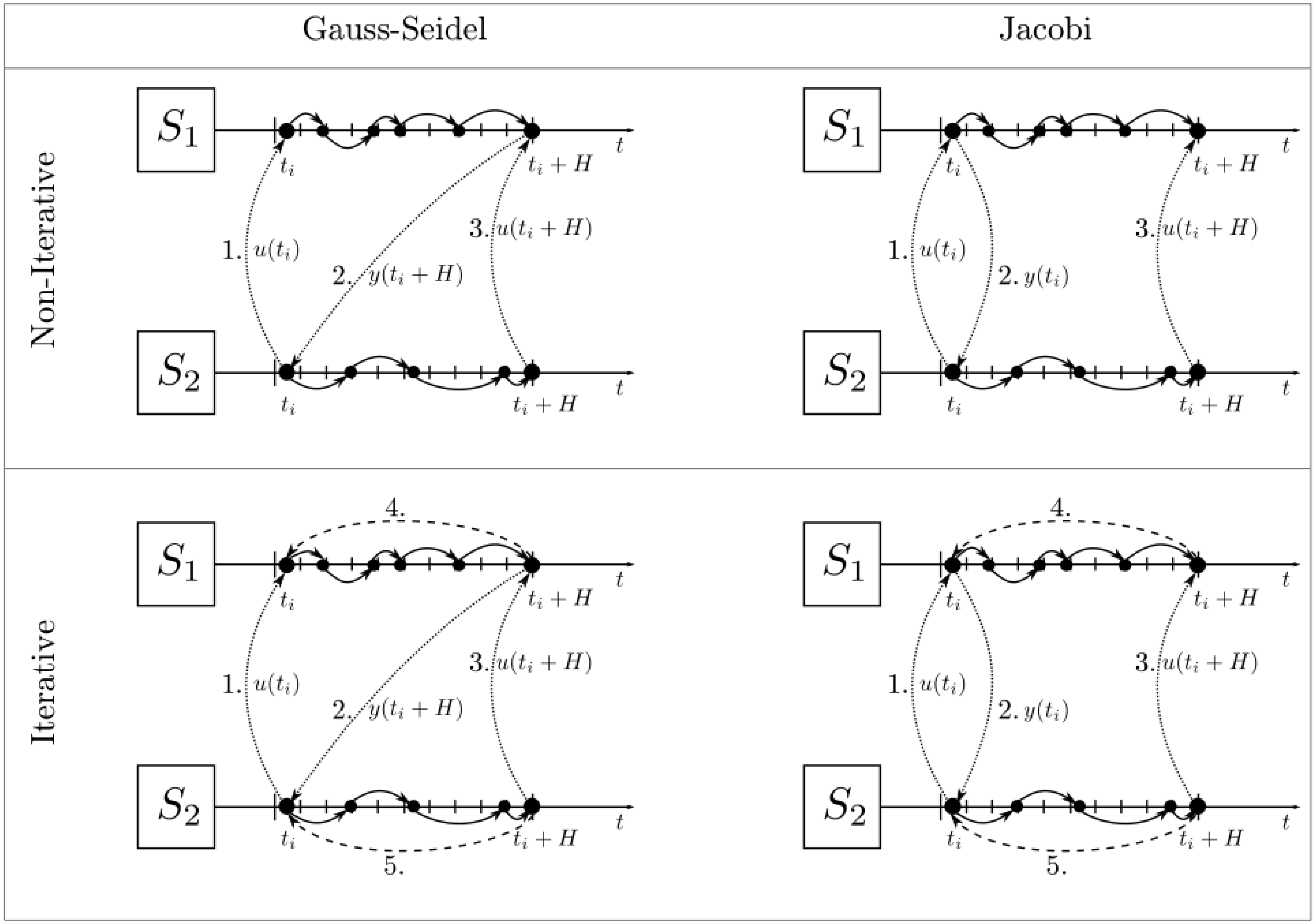
Polynomial

Extrapolated/Interpolation

Context-aware

Model ID'ed

Sources of Errors in Co-simulation



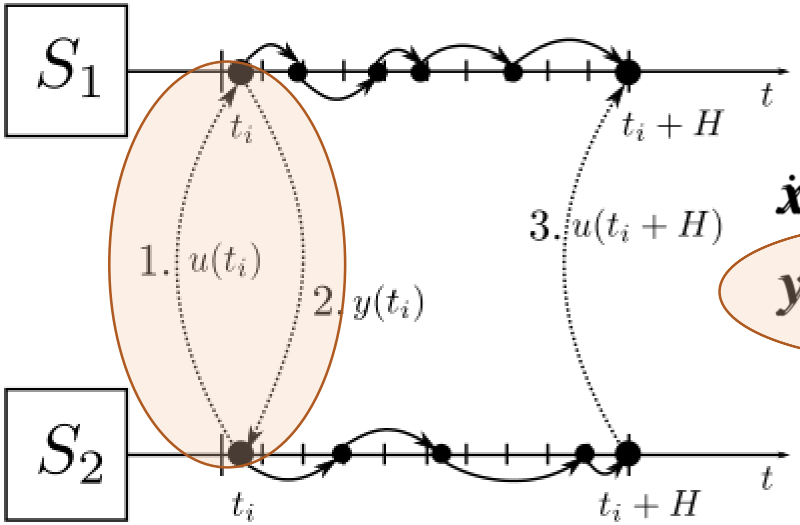
Master Algorithm

Sources of Errors in Co-simulation

Simulation Unit Restrictions

“There is the additional restriction in “slaveInitialized” state that it is not allowed to call fmi2GetXXX functions after fmi2SetXXX functions without an fmi2DoStep call in between.”

Page 104, “Functional Mock-up Interface for Model Exchange and Co-Simulation,” 2014.

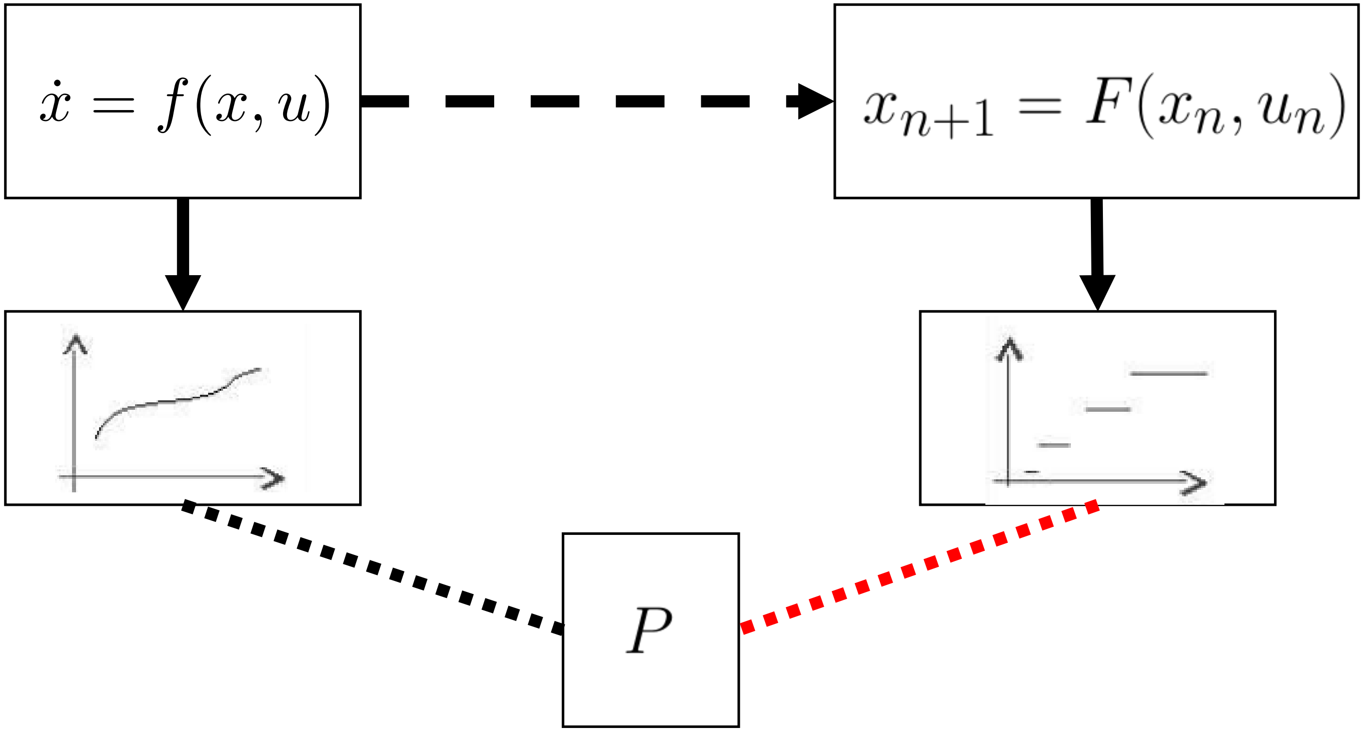


$$\dot{\boldsymbol{x}} = F(\boldsymbol{x}, \boldsymbol{u}),$$
$$\boldsymbol{y} = G(\boldsymbol{x}, \boldsymbol{u}),$$

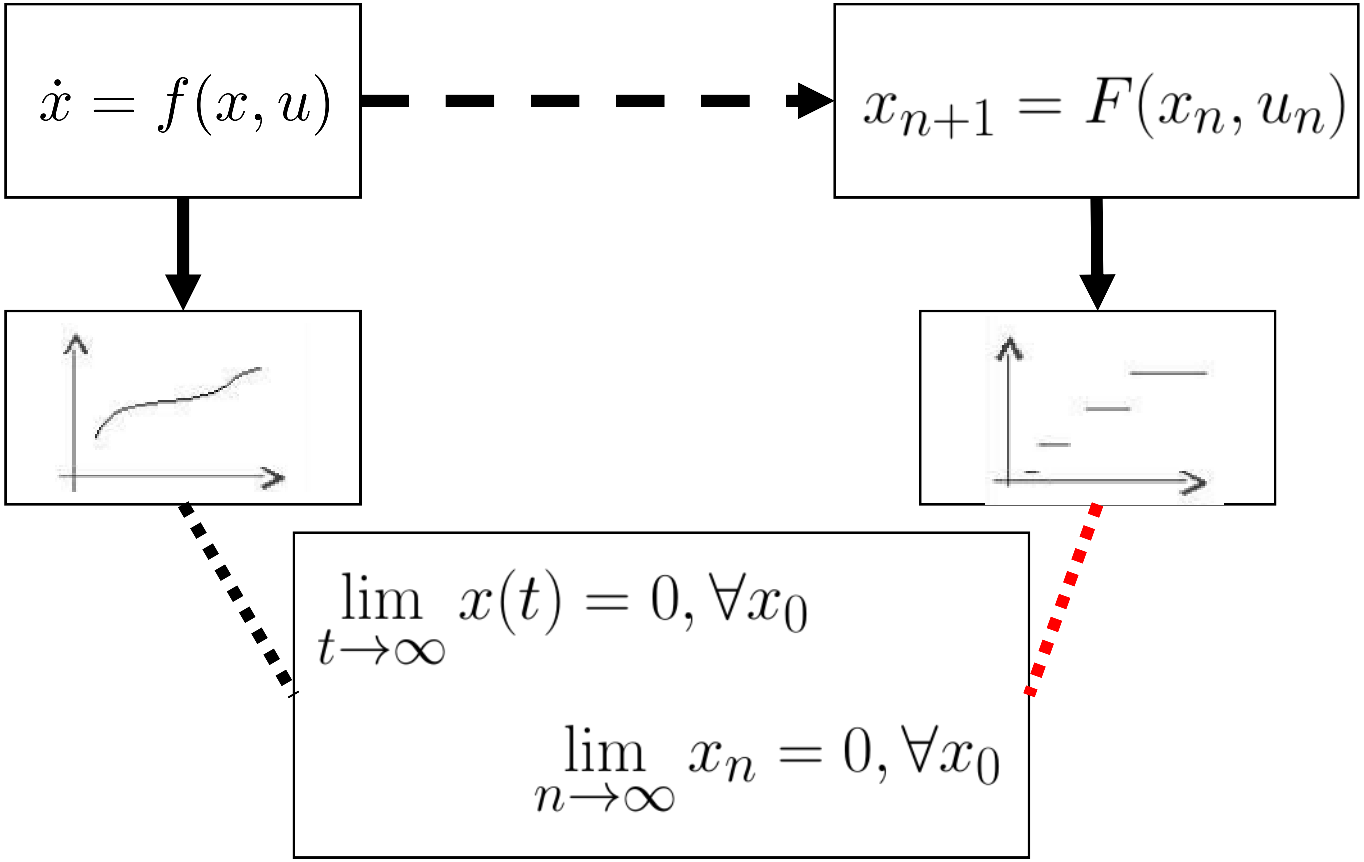
Properties - Definition

Given P satisfied by S,

co-simulation preserves P if cosim(S) satisfies P



Properties - Stability



$$\dot{x} = f(x, u)$$

$$x_{n+1} = F(x_n, u_n)$$

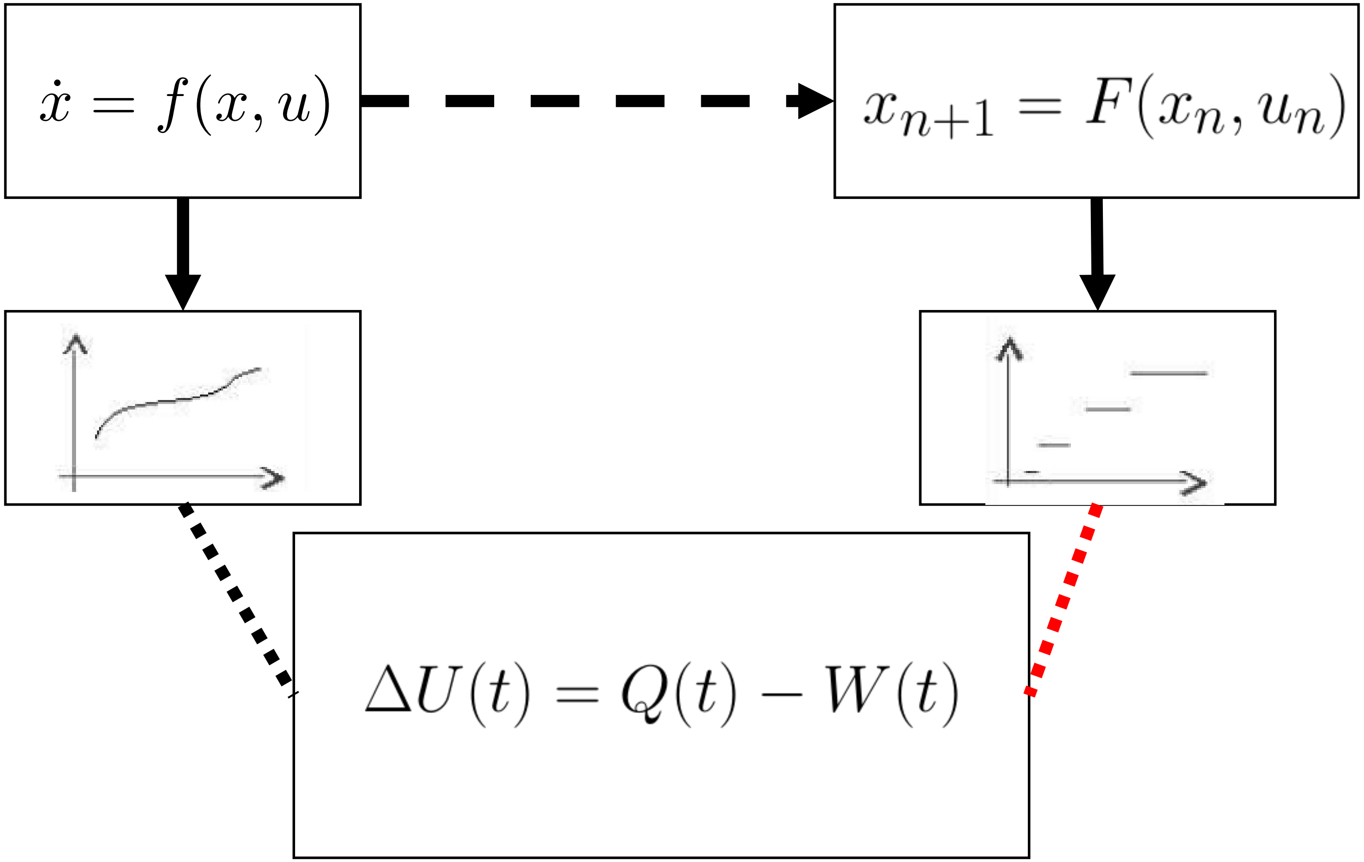
$$\lim_{t \to \infty} x(t) = 0, \forall x_0$$

$$\lim_{n \to \infty} x_n = 0, \forall x_0$$

Gomes, Cláudio, Benoît Legat, Raphaël M. Jungers, and Hans Vangheluwe. “Stable Adaptive Co-Simulation : A Switched Systems Approach.” In *IUTAM Symposium on Co-Simulation and Solver Coupling*, to appear. Darmstadt, Germany, 2017.

Busch, Martin. “Continuous Approximation Techniques for Co-Simulation Methods: Analysis of Numerical Stability and Local Error.” *ZAMM - Journal of Applied Mathematics and Mechanics* 96, no. 9 (September 1, 2016): 1061–81. https://doi.org/10.1002/zamm.201500196.

Properties – Energy Conservation



$$\dot{x} = f(x, u)$$

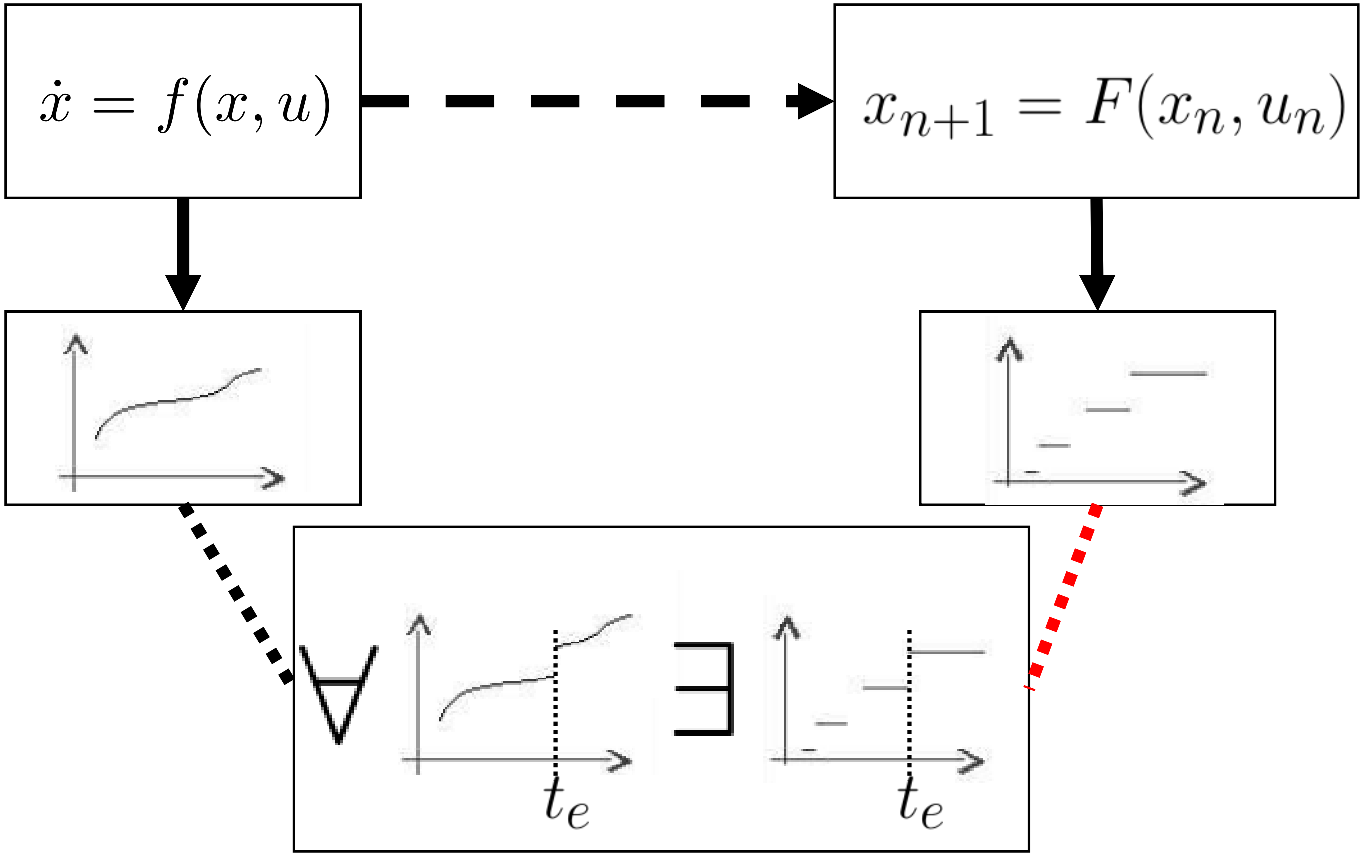
$$x_{n+1} = F(x_n, u_n)$$

$$\Delta U(t) = Q(t) - W(t)$$

Sadjina, Severin, and Eilif Pedersen. “Energy Conservation and Coupling Error Reduction in Non-Iterative Co-Simulations,” June 16, 2016. http://arxiv.org/abs/1606.05168.

Benedikt, M, D Watzenig, J Zehetner, and A Hofer. “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, 1–12. Ibiza, Spain, 2013.

Properties – Event Synchrony



$\dot{x} = f(x, u)$

$x_{n+1} = F(x_n, u_n)$

$\forall$ t_e $\exists$ t_e

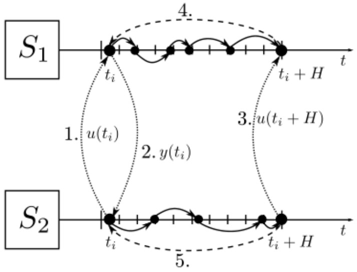
Gheorghe, L., F. Bouchhima, G. Nicolescu, and H. Boucheneb. “A Formalization of Global Simulation Models for Continuous/Discrete Systems.” In Summer Computer Simulation Conference, 559–66. San Diego, CA, USA: Society for Computer Simulation International San Diego, CA, USA, 2007.

Gomes, Cláudio, Paschalis Karalis, Eva M. Navarro-López, and Hans Vangheluwe. “Approximated Stability Analysis of Bi-Modal Hybrid Co-Simulation Scenarios.” In 1st Workshop on Formal Co-Simulation of Cyber-Physical Systems, 345–60. Trento, Italy: Springer, Cham, 2017. https://doi.org/10.1007/978-3-319-74781-1_24.

Verification of Master Algorithms

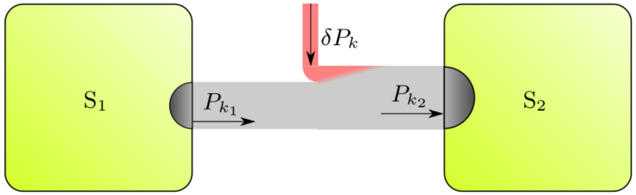
Long term goal: under which conditions the co-simulations preserve given properties…

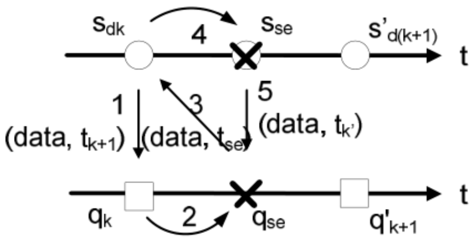
…and what can be done when these are not preserved?



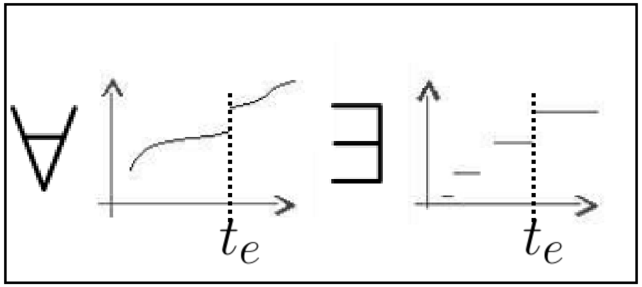
Examples:

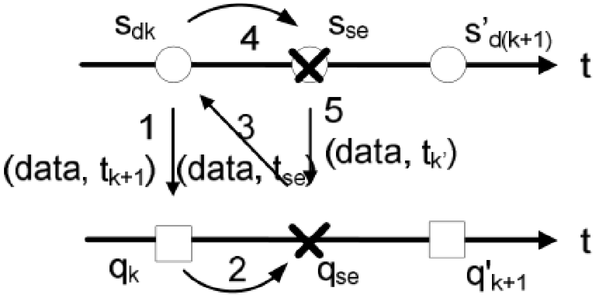
- Stability? Apply strong coupling.
- Energy conservation? Use power bonds and correct forces.
- Event Synchrony? Use event detection/location techniques.

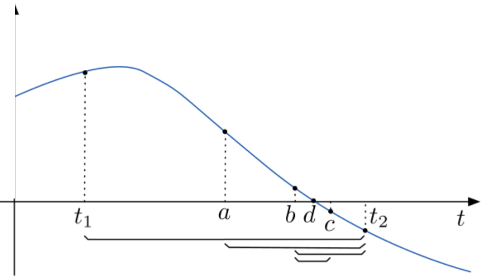


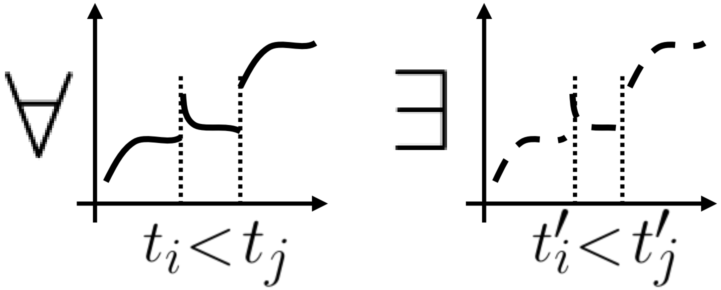


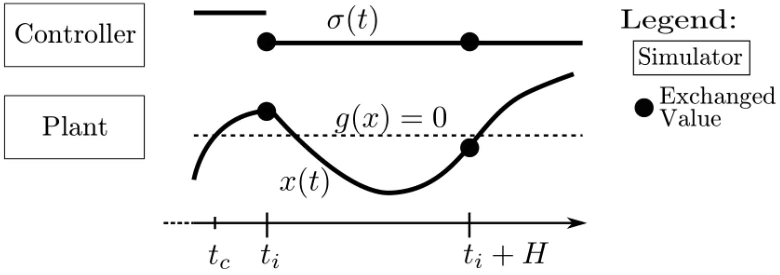
Event Ordering Property





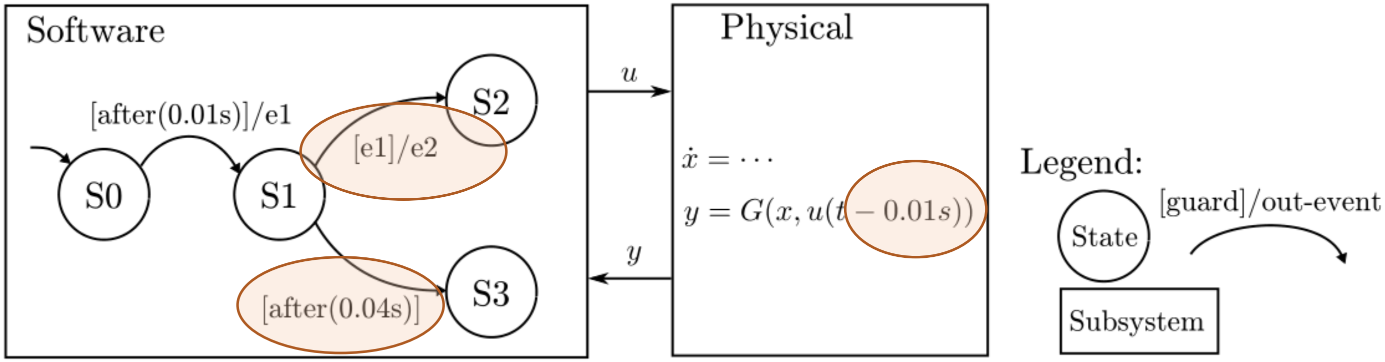


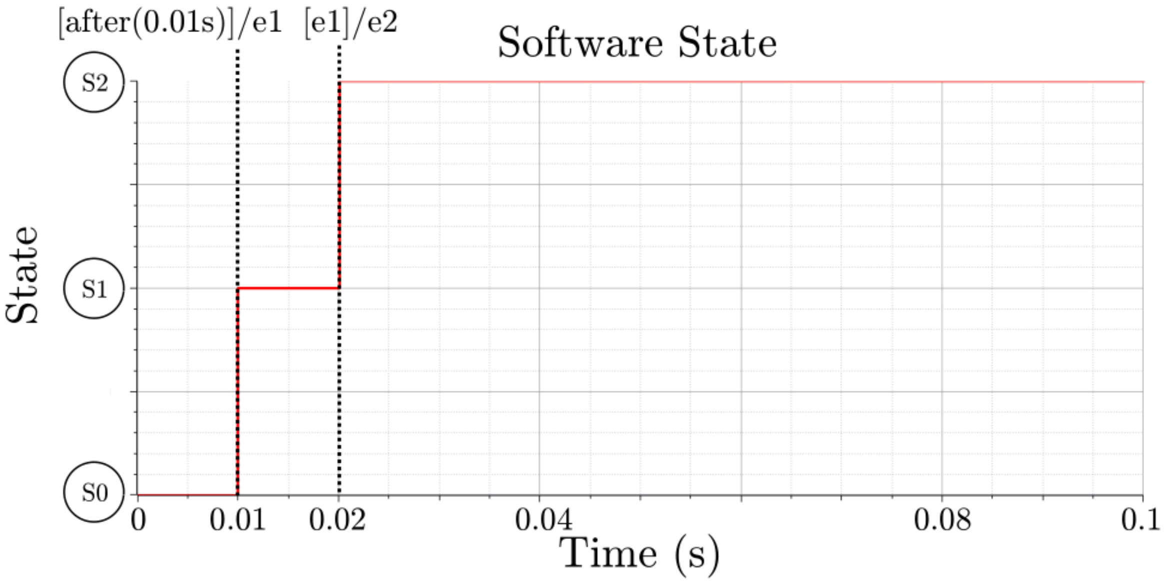




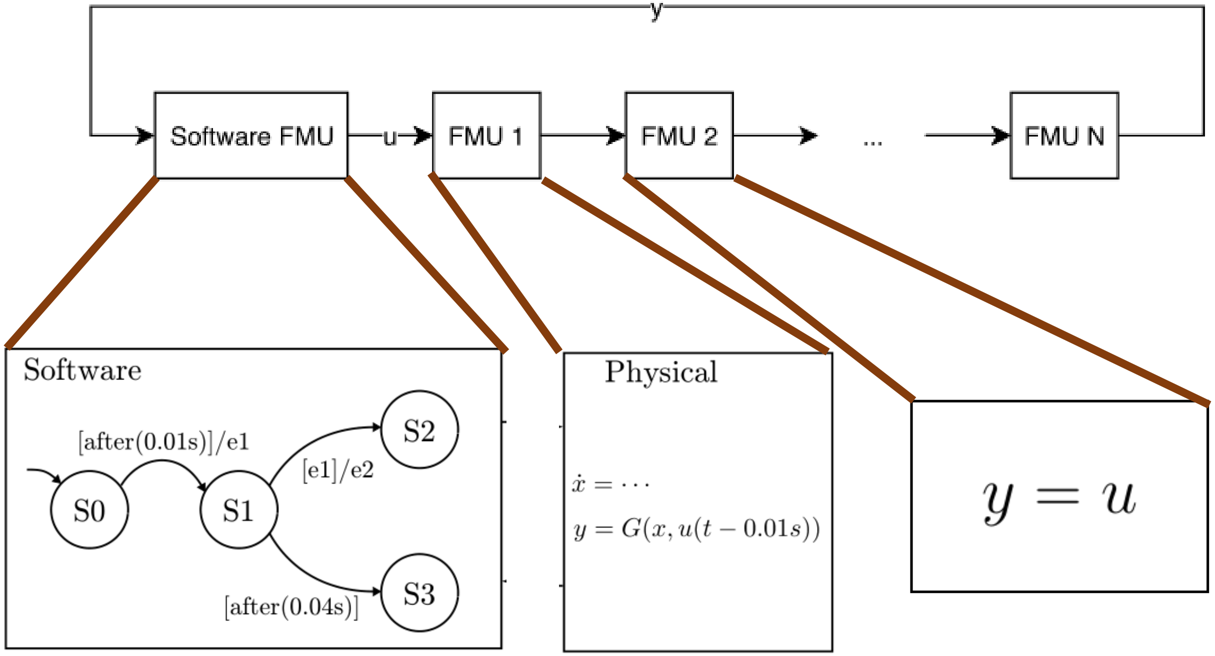
Gheorghe, Luiza. “Continuous/Discrete Co-Simulation Interfaces from Formalization to Implementation,” 2009. http://publications.polymtl.ca/137/.

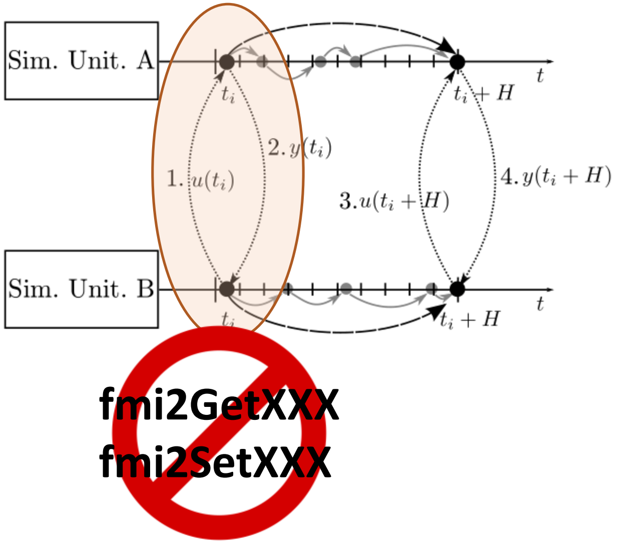
Systems under Study





Co-simulation Impact

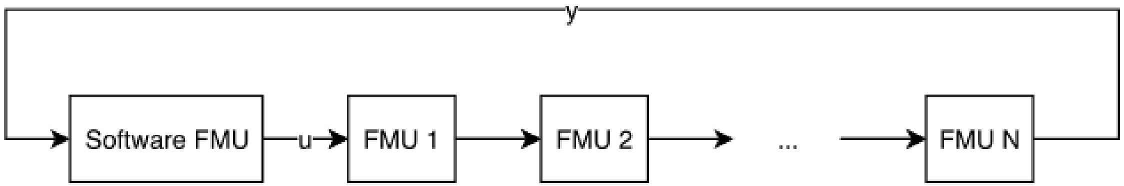


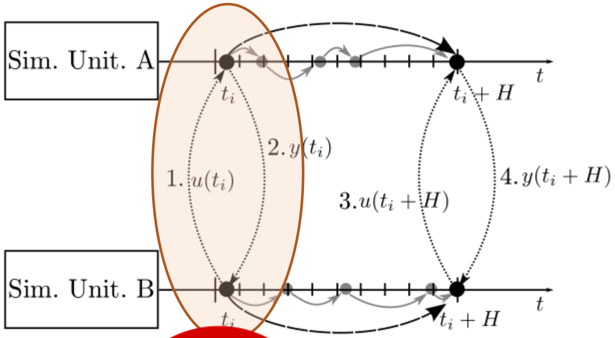


fmi2GetXXX
fmi2SetXXX

No event is detected when new input is set.

Co-simulation Impact

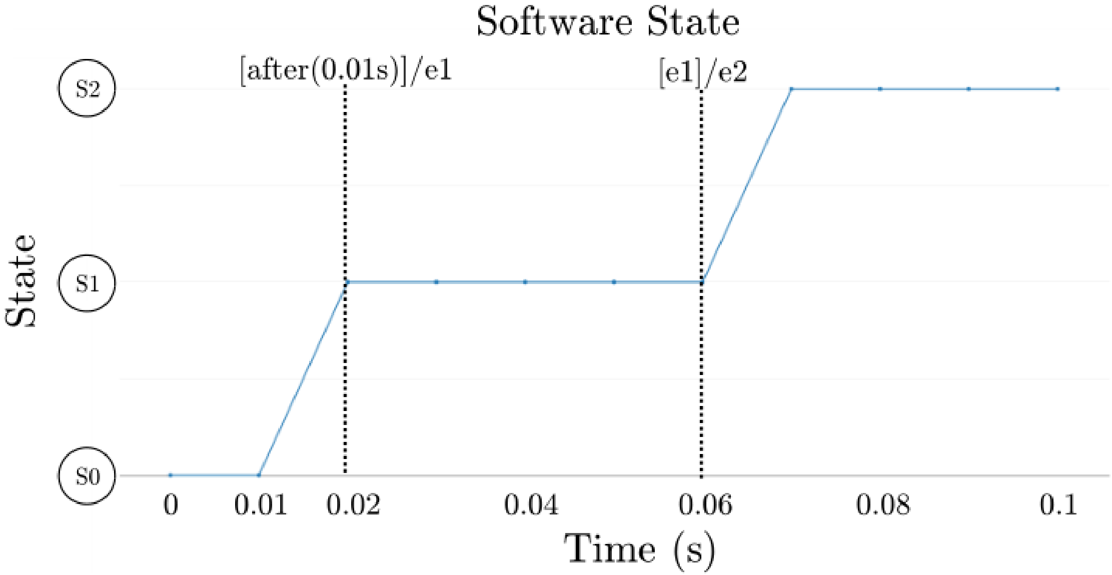


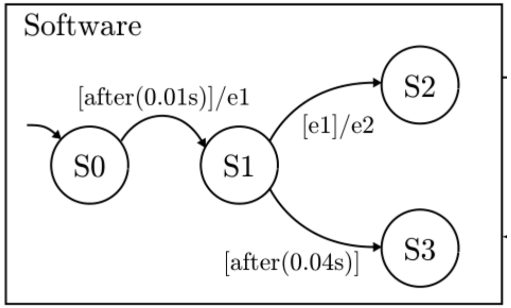


fmi2GetXXX

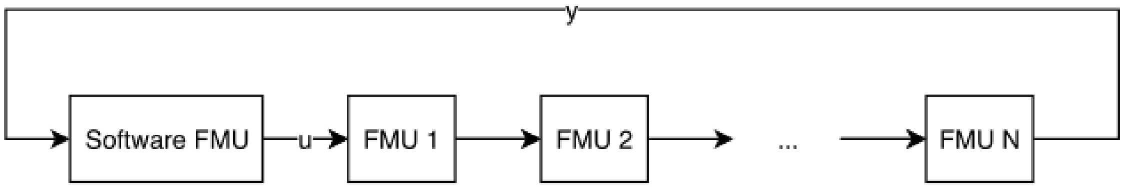
fmi2SetXXX

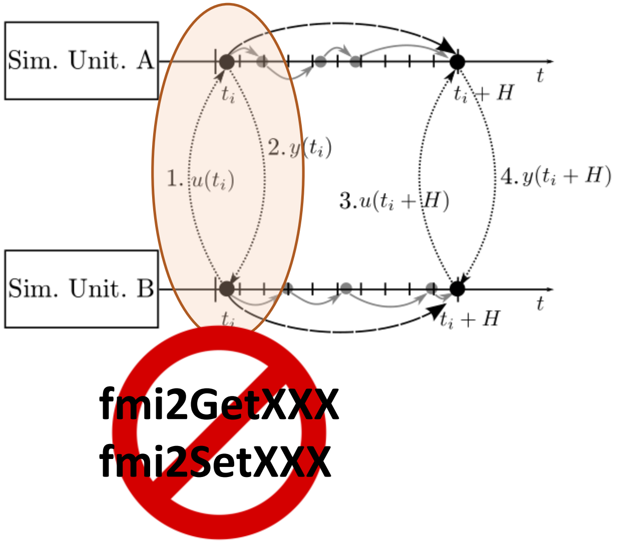
$$H = 0.01 \quad N = 3$$





Co-simulation Impact

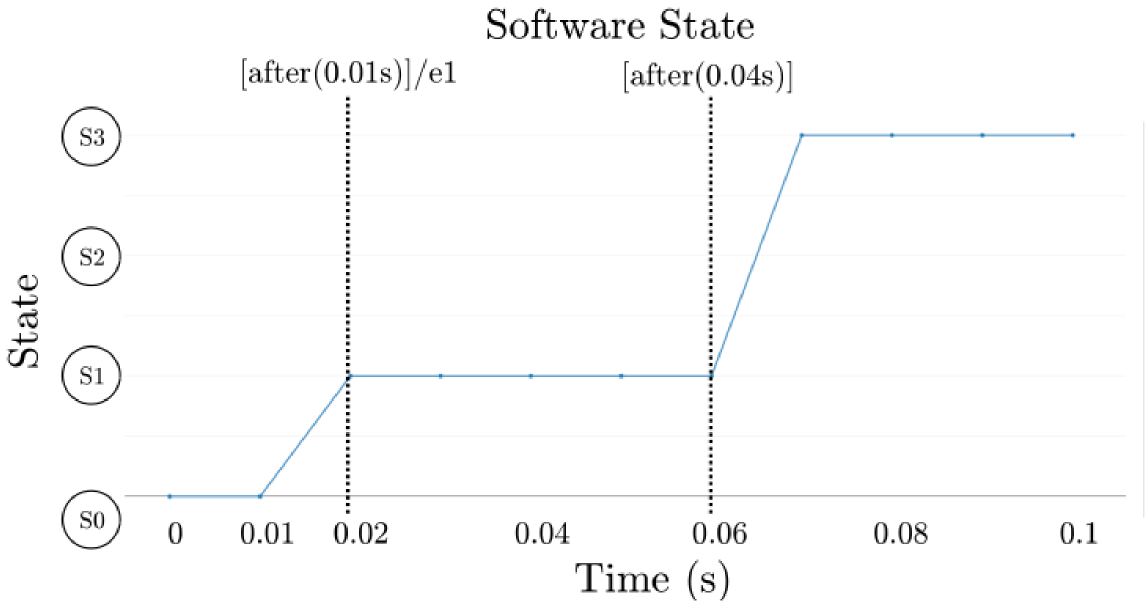


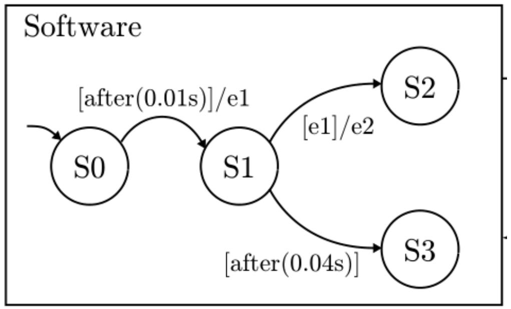


fmi2GetXXX

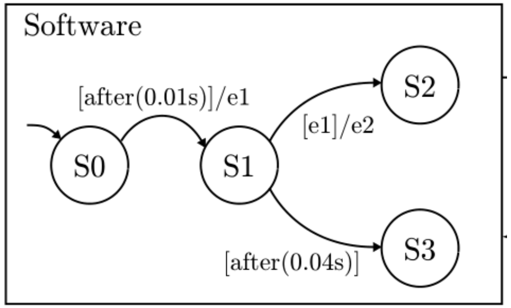
fmi2SetXXX

$$H = 0.01 \quad N = 6$$





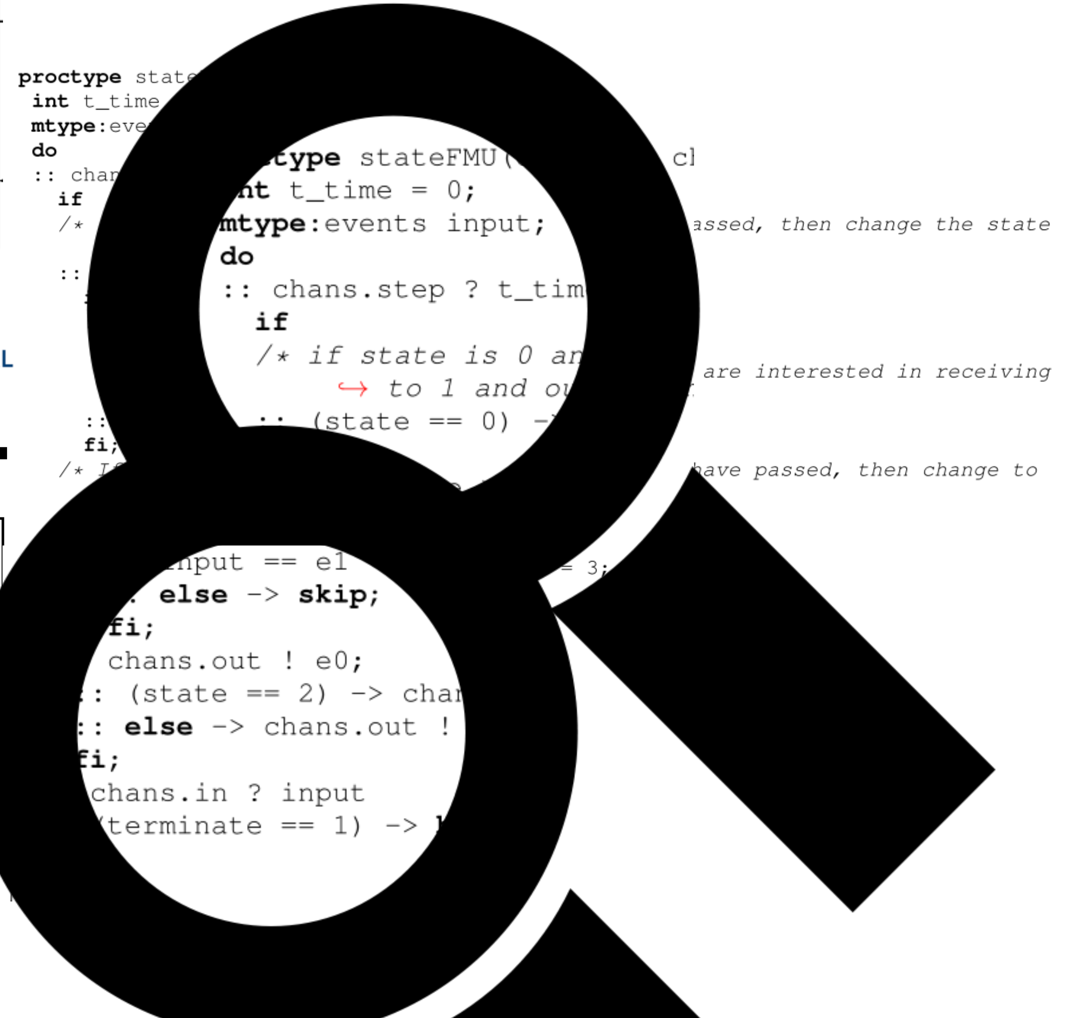
Model Checking – Software FMU



+

~~fmi2GetXXX~~
~~fmi2SetXXX~~

stateFMU.pml

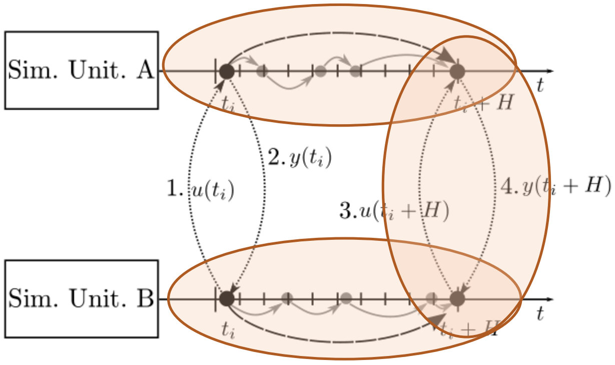


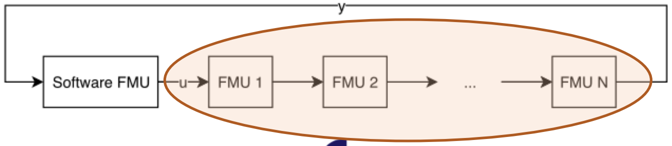
```
proctype state
 int t_time
 mtype:eve
 do
 :: char
   if
   /*
                                          assed, then change the state
   ::
                                          are interested in receiving
     ::
     fi;
   /* I                                   have passed, then change to
                                          = 3;
```

```
    type stateFMU(              cl
    nt t_time = 0;
    mtype:events input;
    do
    :: chans.step ? t_tim
      if
      /* if state is 0 an
         ↪ to 1 and ou
      :: (state == 0) ->
```

```
      input == e1
    : else -> skip;
    fi;
    chans.out ! e0;
  :: (state == 2) -> char
  :: else -> chans.out !
  fi;
  chans.in ? input
  (terminate == 1) ->
```

Model Checking – Jacobi



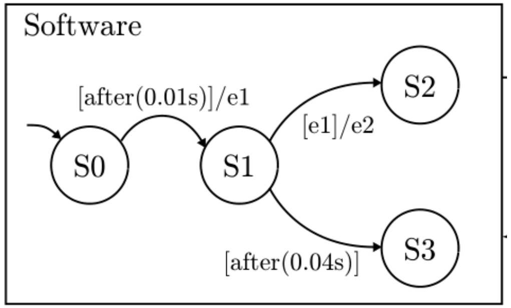


+

jacobi.pml

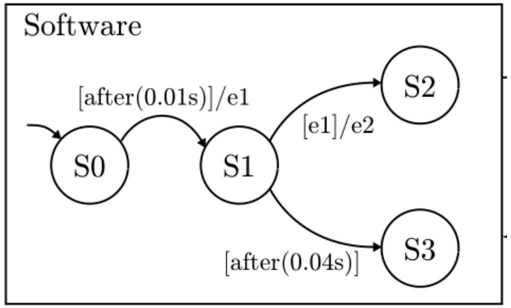
```
proctype MAJacobi(){
   int propagateCount;
   select ( propagateCount : 1 .. (maxN-1) );
   int FMUCount = propagateCount + 1;

   ...

/* Step the FMUs */
for(i : 0 .. FMUCount-1){
  fmuChannels[i].step ! time+1;
}

/* Retrieve the outputs */
for(i : 0 .. FMUCount-1){
  fmuChannels[i].out ? inputs[(i + 1) % (FMUCount)];
}

/* Set inputs */
for(i : 0 .. FMUCount-1){
  fmuChannels[i].in ! inputs[i]
}

time++;
```

Property



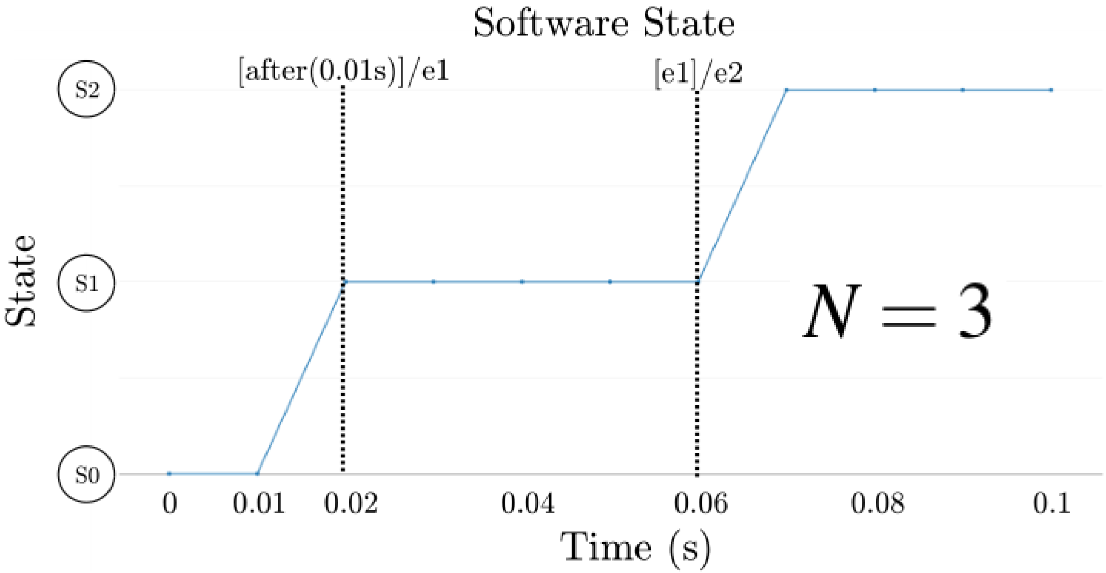
```
<> (state == 2)
```

Results – Counter Example



```
<> (state == 2)
```

$$N = 4$$



$$N = 3$$

Conclusions

- Gauss-Seidel algorithm better than Jacobi
  - Delay is scenario independent
  - Both fail to preserve property for arbitrary H
- Limitations
  - Restricted class of hybrid systems
  - Informal abstraction
- Future work
  - Generalize to other hybrid systems
  - Minimum information to enable proof on black box FMUs?
  - Produce benchmark scenarios to test master algorithms

Thank you!

Casper Thule	casper.thule@eng.au.dk
Cláudio Gomes	**claudio.gomes@uantwerp.be**
Julien Deantoni	julien.deantoni@polytech.unice.fr
Peter G. Larsen	pgl@eng.au.dk
Jörg Brauer	brauer@verified.de
Hans Vangheluwe	hans.vangheluwe@uantwerp.be