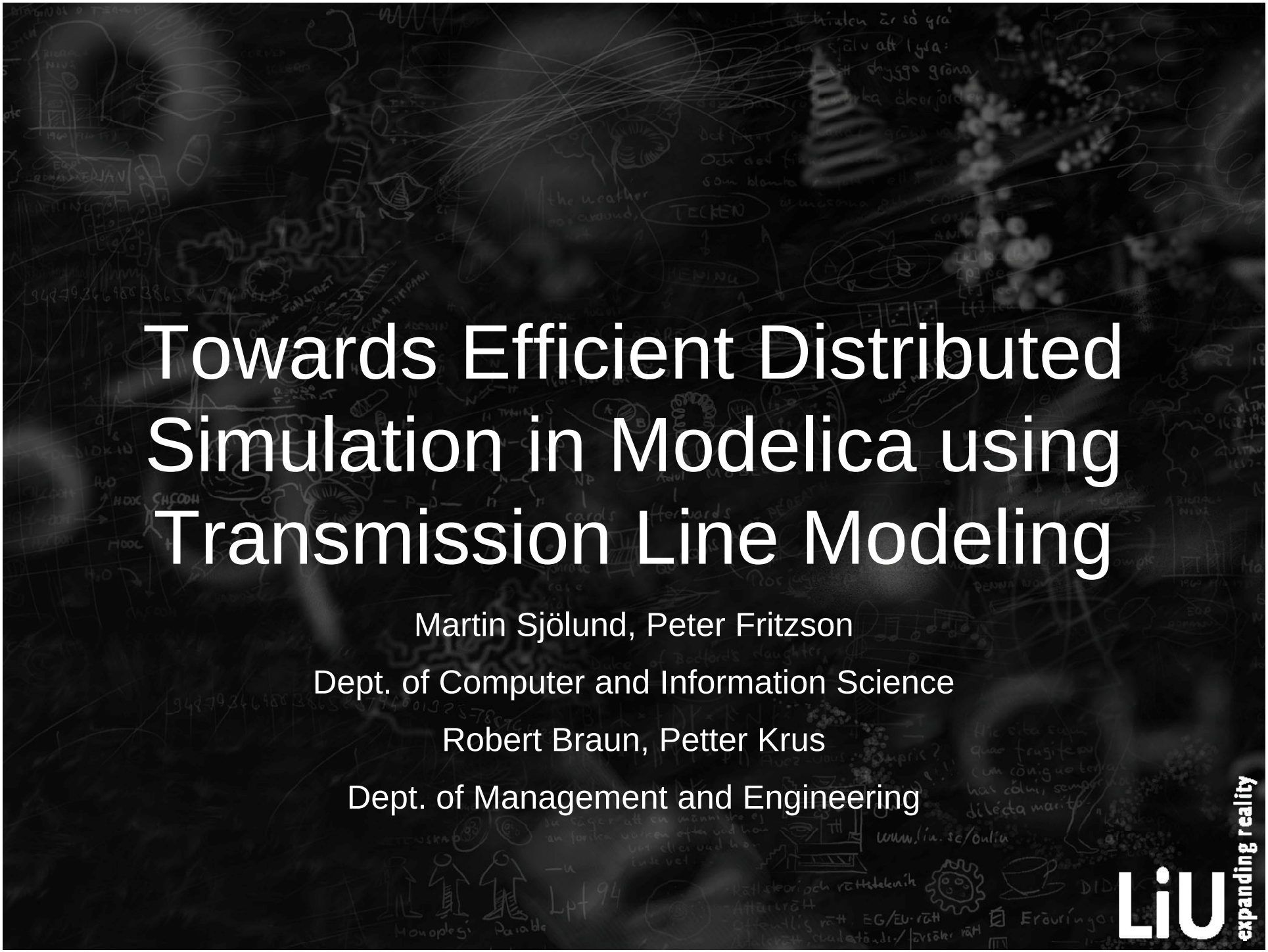


Towards Efficient Distributed Simulation in Modelica using Transmission Line Modeling

Martin Sjölund, Peter Fritzson

Dept. of Computer and Information Science

Robert Braun, Petter Krus

Dept. of Management and Engineering

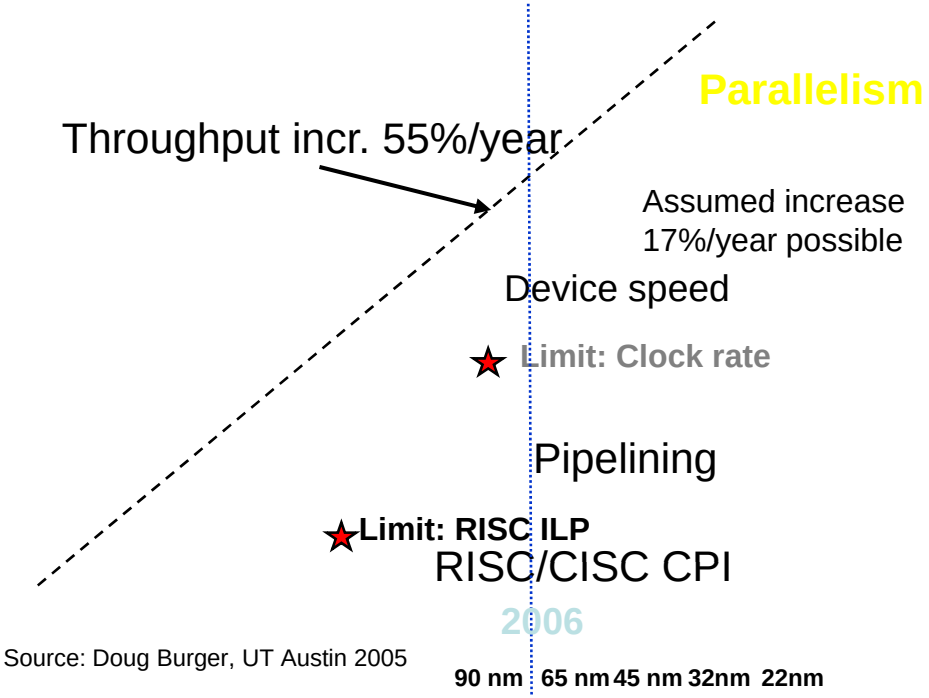
LiU expanding reality

Simulations are slow

- Centralized solvers are hard to parallelize
- Large systems do not scale well
- Stability depends on a global step size

Modern computation units

- Multi-Core is the standard even for home users
- Penalizes single-threaded applications
- Stuck with CPU performance from 2004

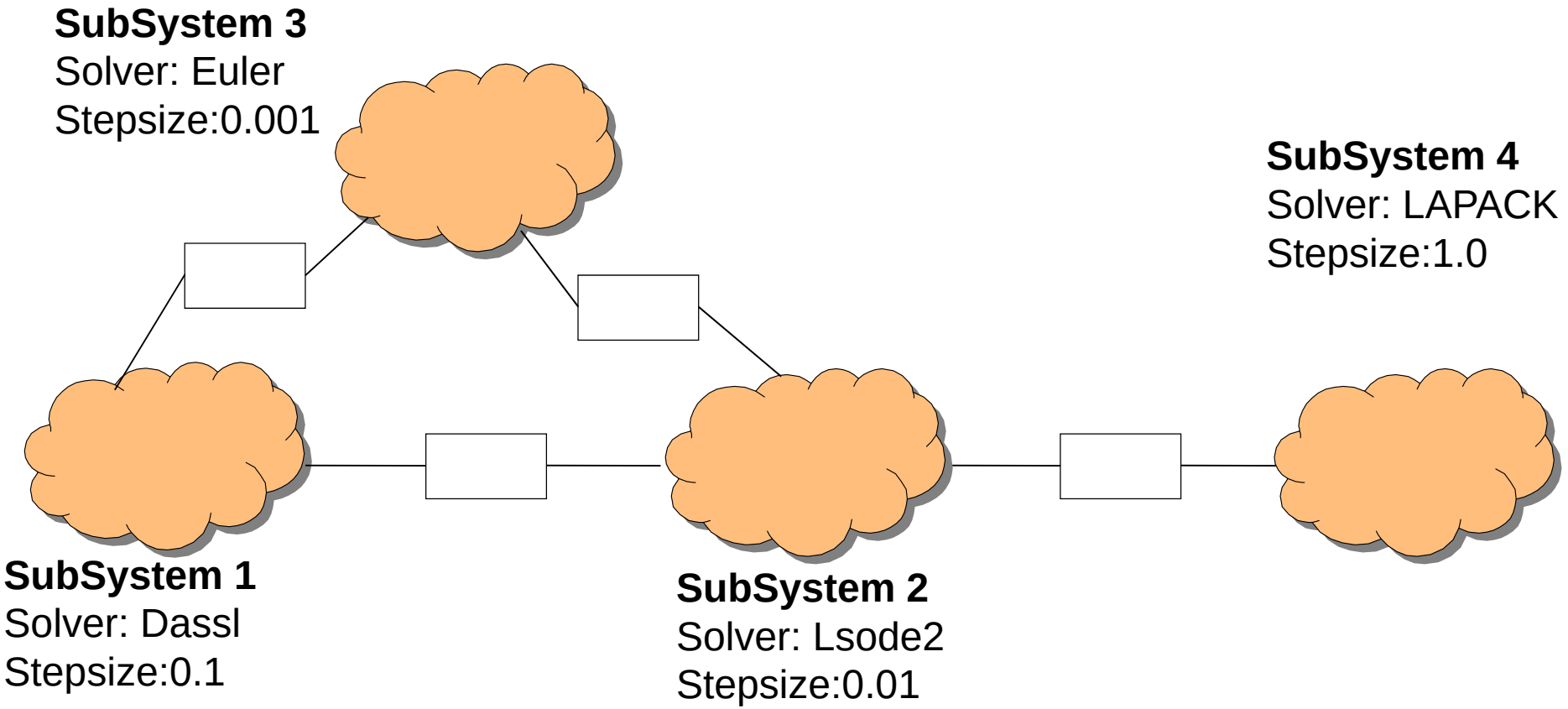


Source: Doug Burger, UT Austin 2005

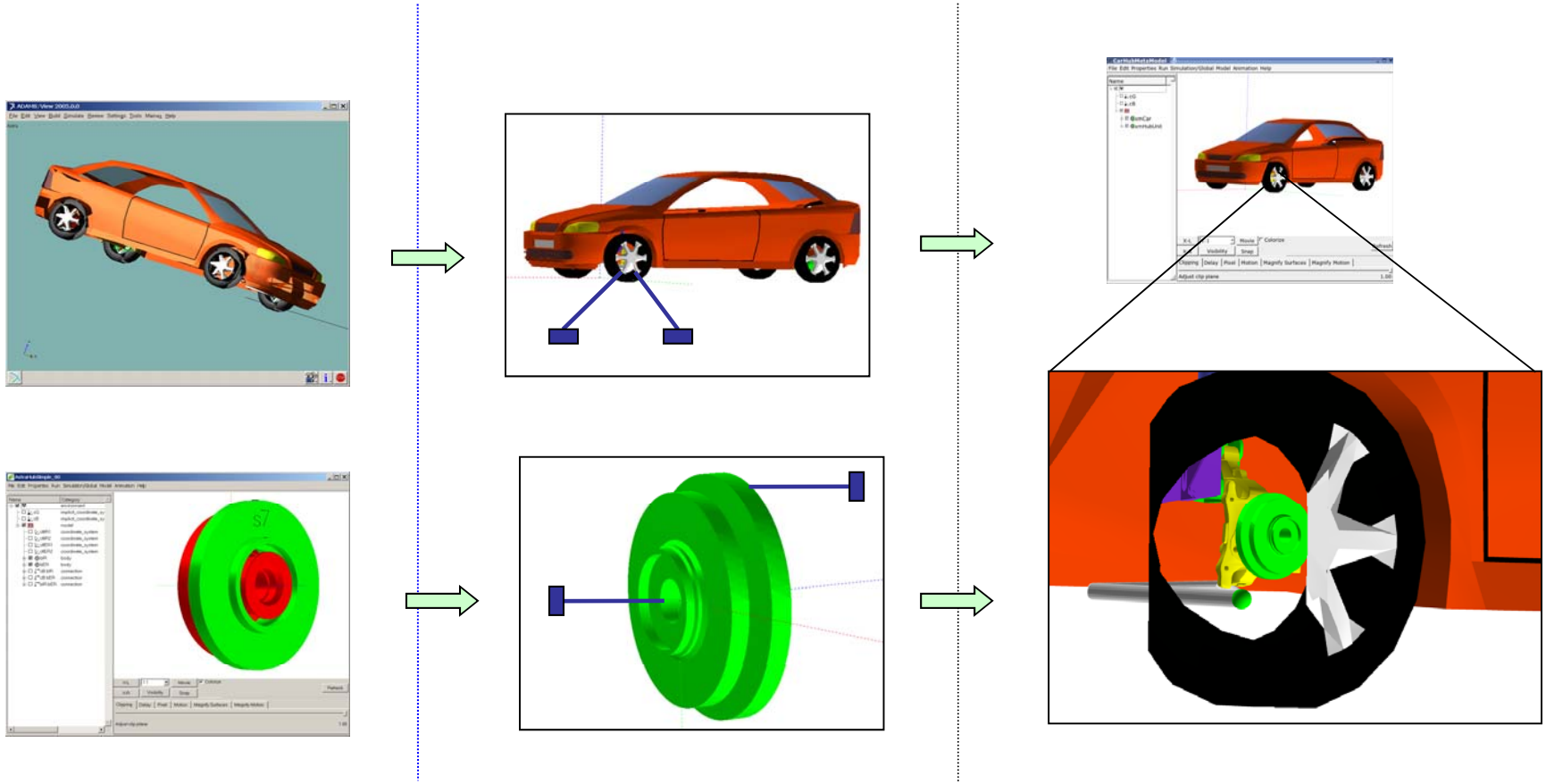
Problems and Solutions

- Problem 1: Speeding up group of simulations
  - Parallel parameter sweep with one simulation per core
- Problem 2: Speeding up single simulation for short real-time deadlines
  - Simplifying the model
  - Parallelizing single simulation

Distributed model



Partitioning the model



Changing the model

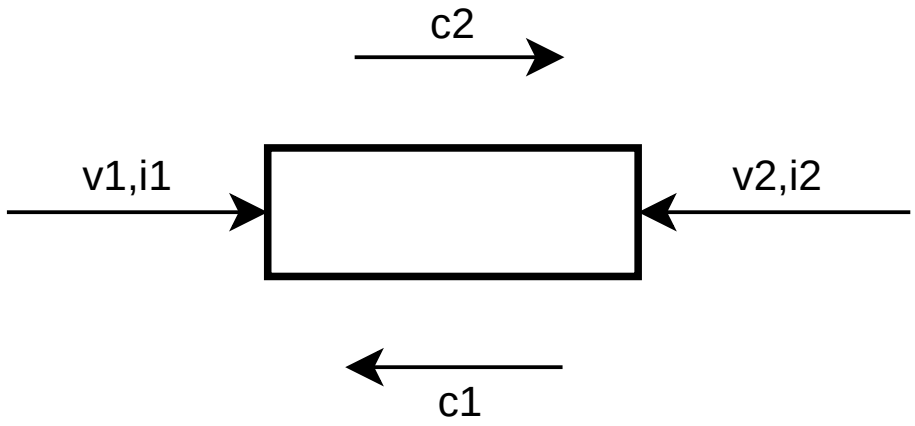
- Changing the connections?
- Find a better model
- Retain physically accuracy

Transmission Line Modeling

- TLM – Transmission Line Modeling – numerically stable co-simulation
- Physically motivated delays are inserted into TLM element models
- Originally used in hydraulics with propagation delays along pipes
- Generalized to other engineering domains

$$c_1(t) = V_2(t - T_{tlm}) + Z_F I_2(t - T_{tlm})$$
$$c_2(t) = V_1(t - T_{tlm}) + Z_F I_1(t - T_{tlm})$$
$$P_1(t) = Z_F I_1(t) + c_1(t)$$
$$P_2(t) = Z_F I_2(t) + c_2(t)$$

c1, c2 are the TLM-parameters
Ttlm is the information propagation time
Zf is the implicit impedance



HOPSAN

- TLM-specific simulation software
- Efficient simulations
- Easy to connect components
- Hard to create new components
  - Written in C++ or Fortran
- Hard to read old code
  - Explicit discretization

Does TLM work in Modelica?

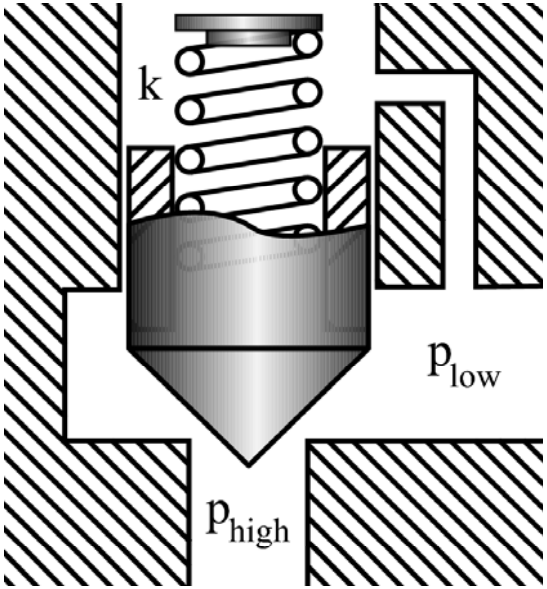
- The discretization is done by the compiler and/or solver
- There are multiple solvers to choose from
- Equation-based models are easy to read and maintain

Decoupling subsystems using delay()?

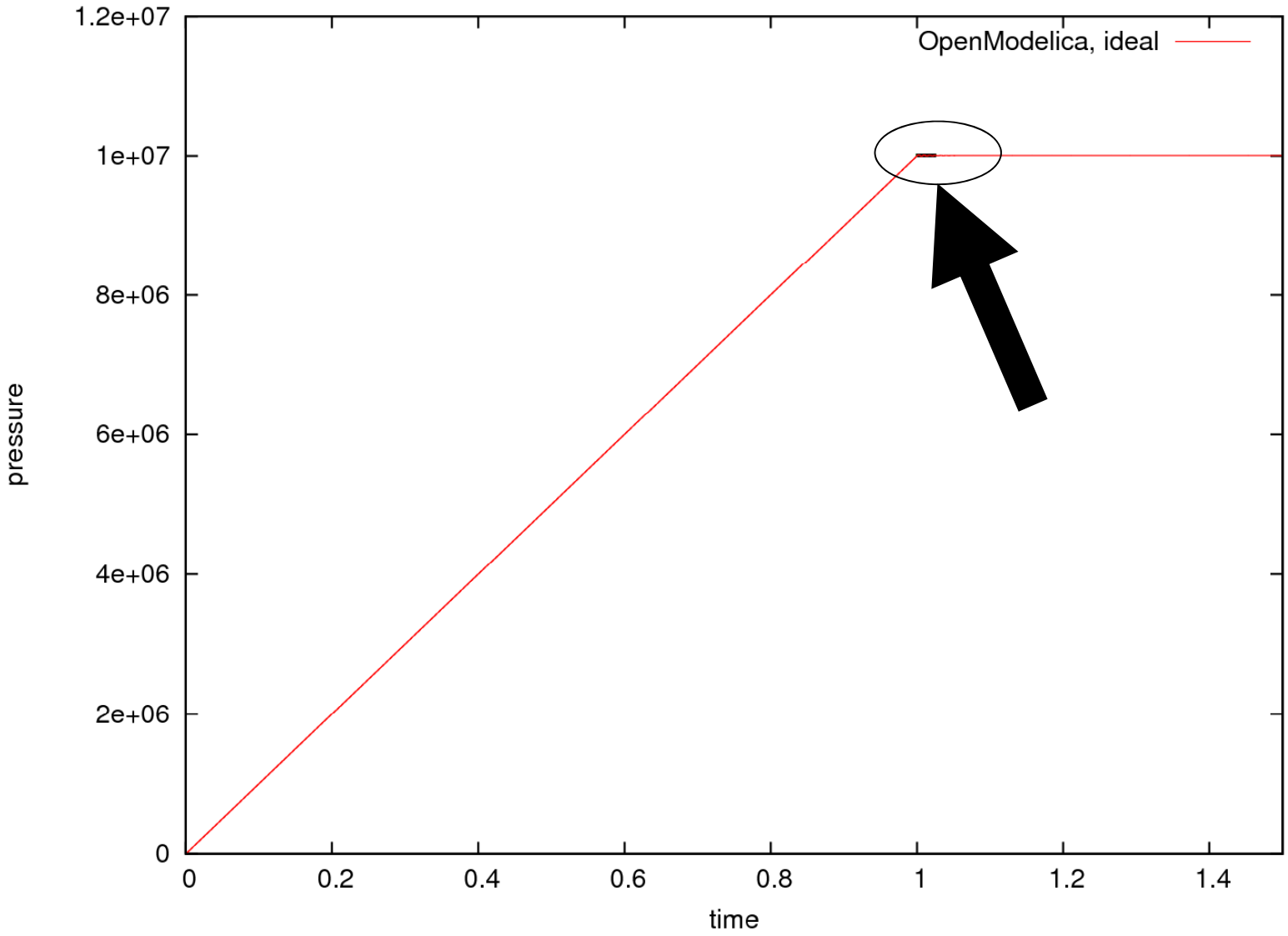
- Breaks dependencies between components
- Interpolation and performance problems
- Initialization cannot be decoupled
  - During initialization, the delay is the actual expression without delay in Modelica

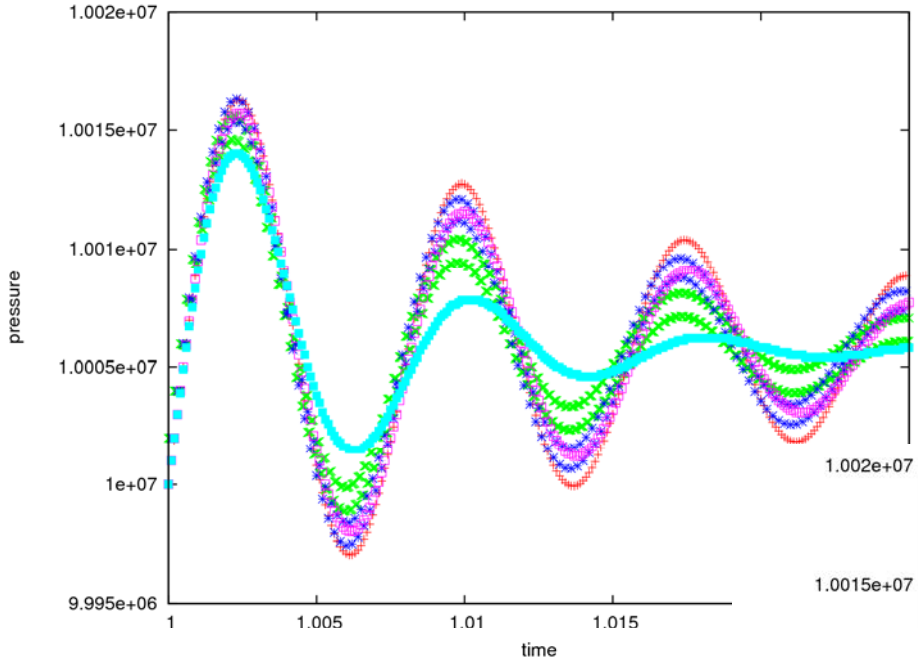
Prototype model

- Pressure relief valve
- 3x2 variations of the model
  - TLM delay-lines between subsystems using delay(), sample(), or der()
  - Spring modeled using der() or explicit euler using the delay() operator



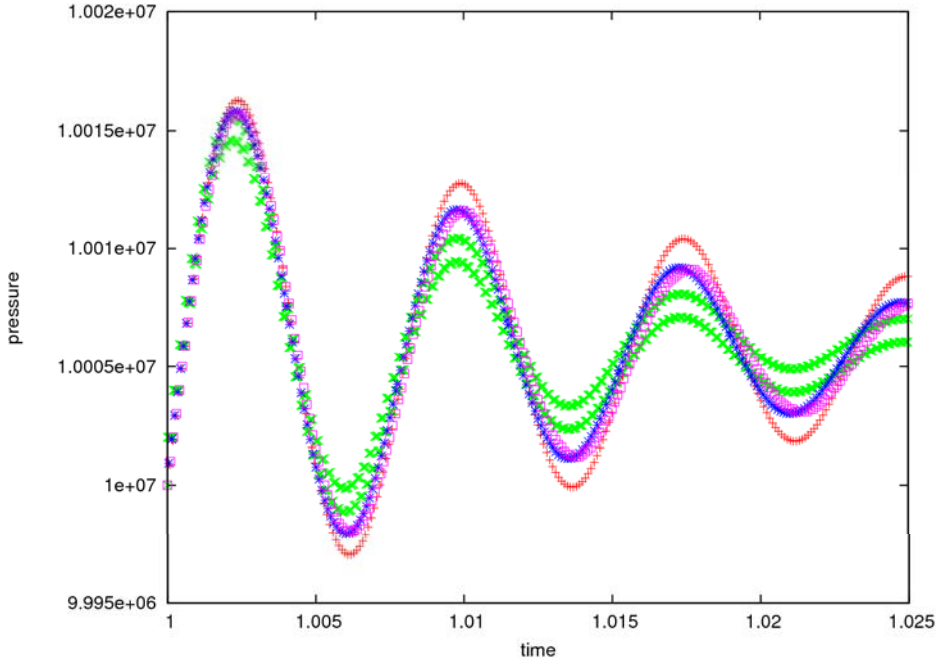

```
Real v = (x-delay(x,T))/T;
// v = der(x)

Real a = (v-delay(v,T))/T;
// a = der(v)
```


1.2e+07
1e+07
8e+06
6e+06
4e+06
2e+06
0
pressure
0
0.2
0.4
0.6
0.8
1
1.2
1.4
time
OpenModelica, ideal



Spring using delay()

der()
sample()
delay()
HOPSAN
HOPSAN, a=0



Spring using der()

Future

- Support manual partitioning of models in OpenModelica
- Integrate these methods in the new OpenModelica parallel backend
- Integrate HOPSAN and OpenModelica models


LINKÖPINGS UNIVERSITET
LINKÖPINGS UNIVERSITET
Linköping University
expanding reality
www.liu.se