EQUATION-BASED MODEL REDUCTION IN OPENMODELICA COMPILER

OPENMODELICA WORKSHOP
05.02.2018, LINKÖPING, SWEDEN

OLIVER LENORD, NIKLAS WORSCHECH,
FATEMEH DAVOUDI, UMUT DURAK, SVEN HARTMANN

BOSCH

Equation-based Model Reduction in OpenModelica Compiler

Content

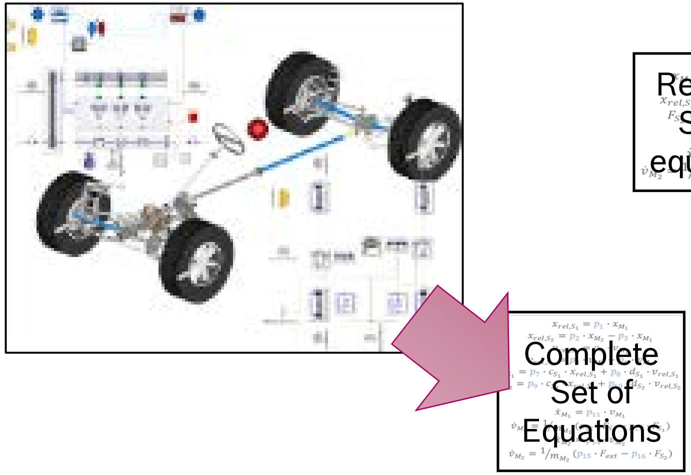
- Motivation
- Model Order Reduction
- Equation-based Model Reduction
- Results of Previous Works
- Implementation in OpenModelica Compiler
- Performance of Current Implementation
- Conclusions
- Outlook

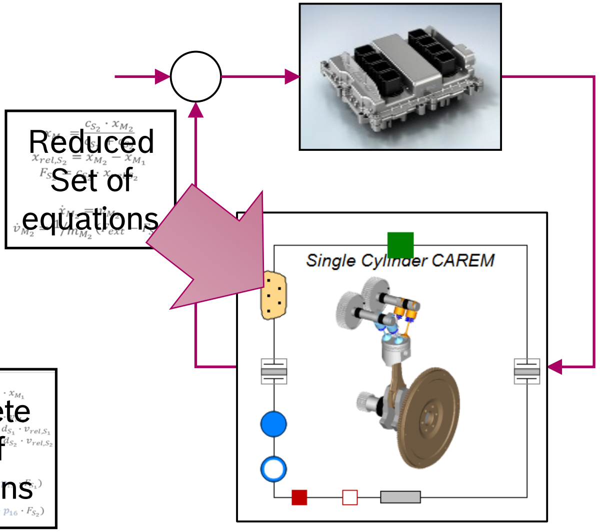
Motivation

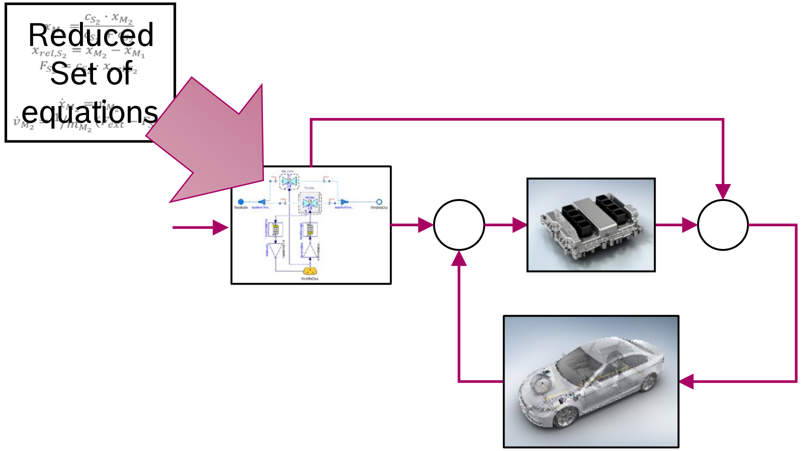
Equation-based Model Reduction in OpenModelica Compiler

Model Reduction Use Cases

- System Model
  - Predict the system behavior in the design space
- SiL/HiL Model
  - Reproduce the system behavior in the operation space
- Control Design Model
  - Reproduce the I/O behavior of the subsystem in the controller range







Lenord, Worschech, Davoudi, Durak, Hartmann, OpenModelica Workshop, Linköping, SE | 02/05/2018

Equation-based Model Reduction in OpenModelica Compiler
Proper Model Generation – Use Cases

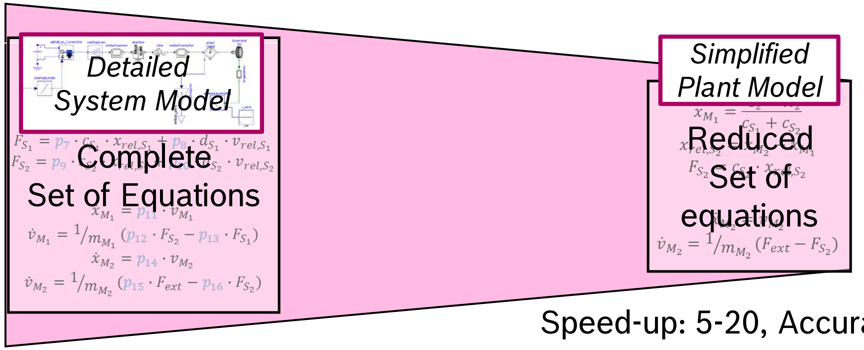
- System Model
  - Predict the system behavior in the design space
  - Detailed model considering all relevant physical effects
  - Parameters derived from design or considered as design variables
  - Validated component models
  - Explore the design space offline to find optimal design
- HiL Model
  - Reproduce the system behavior in the operation space
  - Simplified model considering most relevant phys. effects
  - Parameters derived from detailed model
  - Validated against detailed model
  - Realtime simulation to verify the stability of the controller
- Control Design Model
  - Reproduce the I/O behavior of the subsystem in the controller range
  - Simplified model considering only the required phys. effects
  - Parameter identified through measurements
  - Validate against measurements
  - ECU runtime capable model as part of the control algorithm

BOSCH

Model Order Reduction

Equation-based Model Reduction in OpenModelica Compiler
Equation-based Model Reduction vs. Model Order Reduction

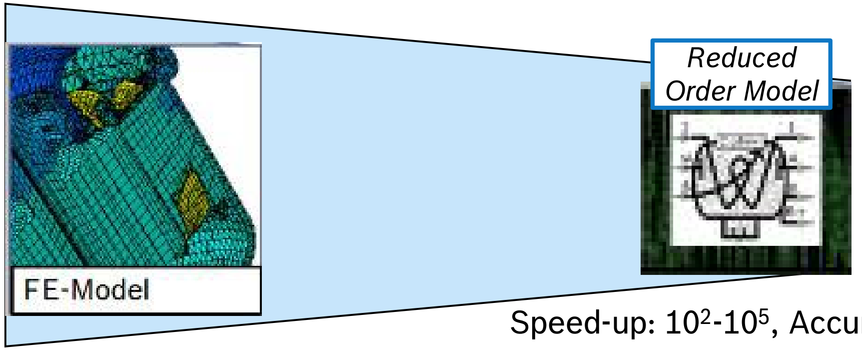
Equation-based Model Reduction



Speed-up: 5-20, Accuracy >90%

- Based on a set of symbolic equations and a reference solution.
- Derive a reduced set of equations in a defined accuracy of the required outputs.

Model Order Reduction (MOR)



Speed-up: 10^2-10^5, Accuracy >98%

- Based on finite element models.
- Derive very fast and highly accurate compact models[1)].

1) Antoulas, A. C. 2005. “An Overview of Approximation Methods for Large-Scale Dynamical Systems.” *Annual Reviews in Control* 29 (2): 181–90. doi:10.1016/j.arcontrol.2005.08.002.

Equation-based Model Reduction in OpenModelica Compiler
Equation-based Model Reduction vs. Model Order Reduction

Equation-based Model Reduction

- Prerequisite:
  - Validated equation based physical model
  - Reference Scenario (Test function + known reference solution)
  - Required accuracy of selected outputs OR required number of FLOPS on target HW
- Limitations:
  - Applicable only to comparatively small models.
  - Does not scale well.
  - Result depends heavily on the quality of the scenario.

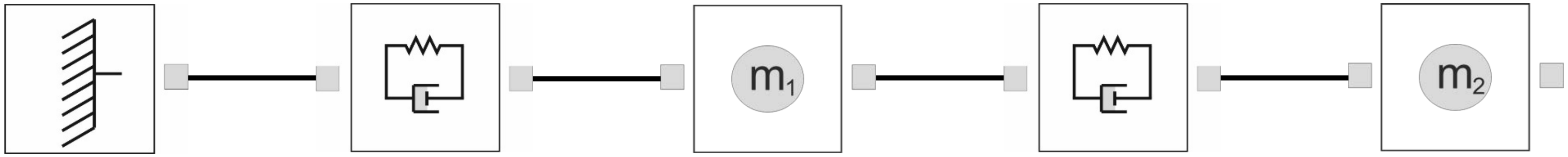
Model Order Reduction (MOR)

- Prerequisite:
  - Validated finite element model
  - Boundary conditions
  - Required accuracy of selected outputs in defined operation range
- Limitations/Challenges:
  - Discontinuities.
  - Parametric / Non-linear Reduced Order Model (ROM) requires special techniques[1)].
  - Dense models from system simulations that do not stem from an integral kernel.

1) Dorwarth, M., S. Kehrberg, R. Maul, R. Eid, F. Lang, B. Schmidt, and J. Mehner. 2014. "Nonlinear Model Order Reduction for High Q MEMS Gyroscopes." In *2014 IEEE 11th International Multi-Conference on Systems, Signals Devices (SSD14)*, 1–4. doi:10.1109/SSD.2014.6808820.

Equation-based Model Order Reduction

Equation-based Model Reduction in OpenModelica Compiler
Manual Procedure

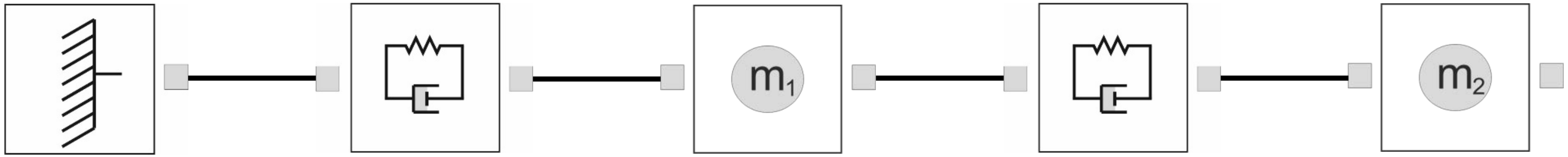


- Assumptions:
  - d, T small
  - $m_{M_1} \ll m_{M_2}$

$$x_{rel,S_1} = x_{M_1}$$
$$x_{rel,S_2} = x_{M_2} - x_{M_1}$$
$$v_{rel,S_1} = v_{M_1}$$
$$v_{rel,S_2} = v_{M_2} - v_{M_1}$$
$$F_{S_1} = c_{S_1} \cdot x_{rel,S_1} + d_{S_1} \cdot v_{rel,S_1}$$
$$F_{S_2} = c_{S_2} \cdot x_{rel,S_2} + d_{S_2} \cdot v_{rel,S_2}$$

$$\dot{x}_{M_1} = v_{M_1}$$
$$\dot{v}_{M_1} = {}^{1}/_{m_{M_1}} (F_{S_2} - F_{S_1})$$
$$\dot{x}_{M_2} = v_{M_2}$$
$$\dot{v}_{M_2} = {}^{1}/_{m_{M_2}} (F_{ext} - F_{S_2})$$

Equation-based Model Reduction in OpenModelica Compiler
Manual Procedure



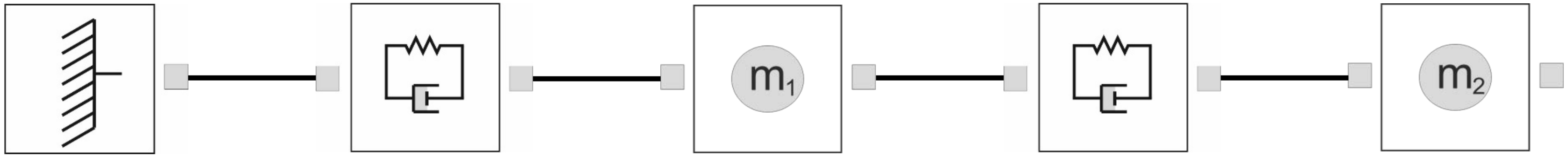
- Assumptions:
  - d, T small
  - $m_{M_1} \ll m_{M_2}$

$$x_{rel,S_1} = x_{M_1}$$
$$x_{rel,S_2} = x_{M_2} - x_{M_1}$$
$$v_{rel,S_1} = v_{M_1}$$
$$v_{rel,S_2} = v_{M_2} - v_{M_1}$$
$$F_{S_1} = c_{S_1} \cdot x_{rel,S_1} + d_{S_1} \cdot v_{rel,S_1}$$
$$F_{S_2} = c_{S_2} \cdot x_{rel,S_2} + d_{S_2} \cdot v_{rel,S_2}$$

$$\dot{x}_{M_1} = v_{M_1}$$
$$\dot{v}_{M_1} = {}^1/_{m_{M_1}} (F_{S_2} - F_{S_1})$$
$$\dot{x}_{M_2} = v_{M_2}$$
$$\dot{v}_{M_2} = {}^1/_{m_{M_2}} (F_{ext} - F_{S_2})$$

Equation-based Model Reduction in OpenModelica Compiler

Manual Procedure

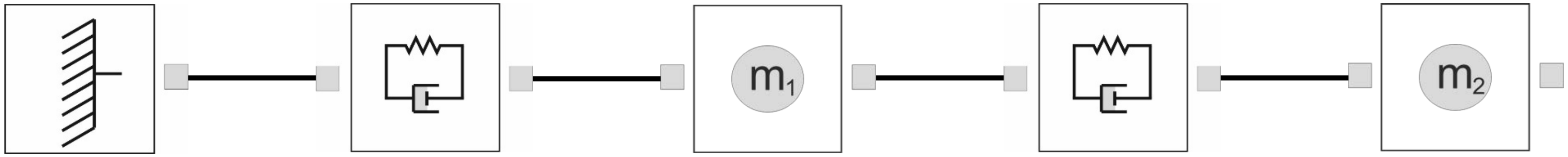


- Assumptions:
  - d, T small
  - $m_{M_1} \ll m_{M_2}$

$$x_{rel,S_1} = x_{M_1}$$
$$x_{rel,S_2} = x_{M_2} - x_{M_1}$$
$$v_{rel,S_1} = v_{M_1}$$
$$v_{rel,S_2} = v_{M_2} - v_{M_1}$$
$$F_{S_1} = c_{S_1} \cdot x_{rel,S_1}$$
$$F_{S_2} = c_{S_2} \cdot x_{rel,S_2}$$

$$\dot{x}_{M_1} = v_{M_1}$$
$$\dot{v}_{M_1} = {}^{1}/_{m_{M_1}} (F_{S_2} - F_{S_1})$$
$$\dot{x}_{M_2} = v_{M_2}$$
$$\dot{v}_{M_2} = {}^{1}/_{m_{M_2}} (F_{ext} - F_{S_2})$$

Equation-based Model Reduction in OpenModelica Compiler
Manual Procedure



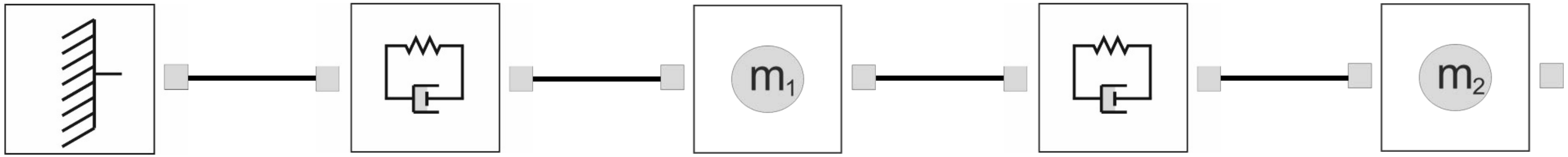
- Assumptions:
  - d, T small
  - $m_{M_1} \ll m_{M_2}$

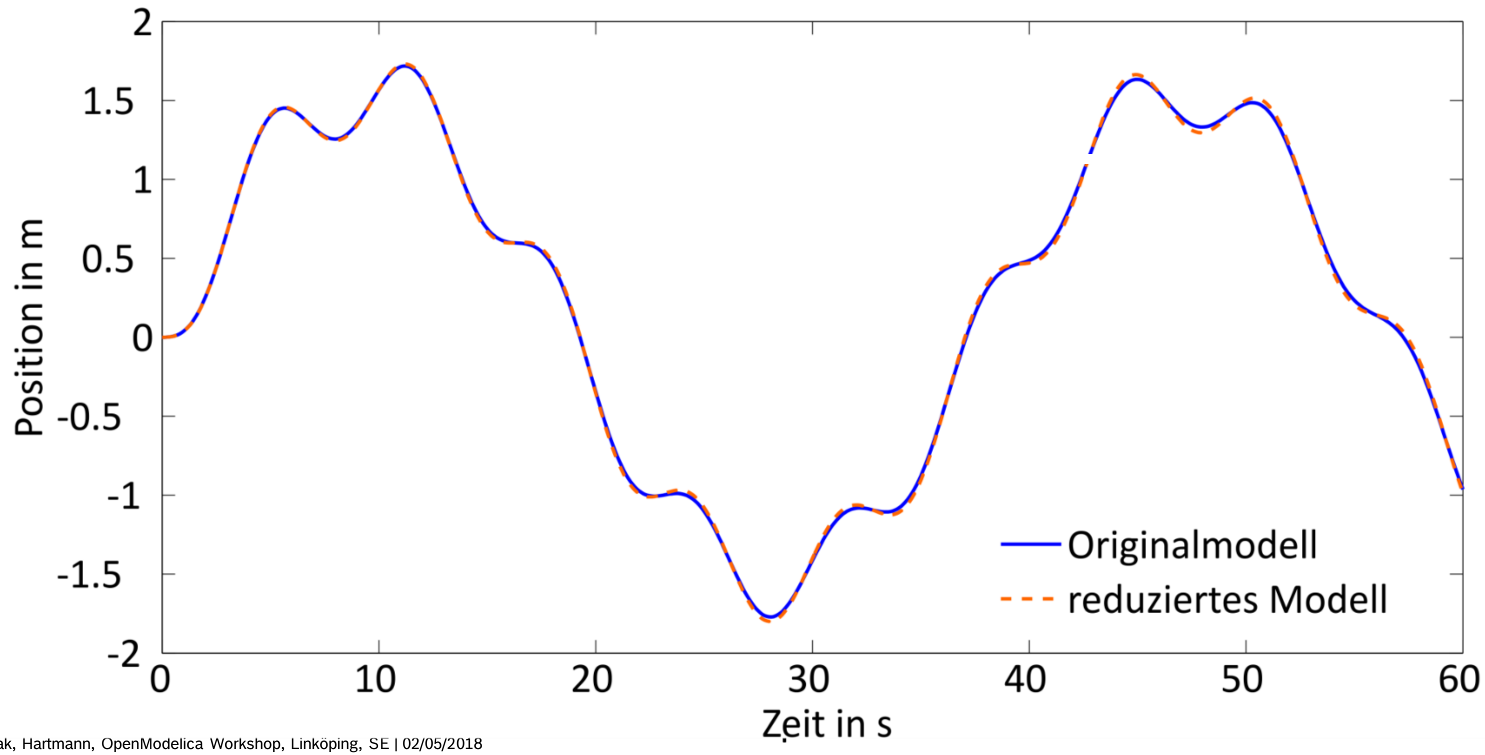
$$x_{M_1} = \frac{c_{S_2} \cdot x_{M_2}}{c_{S_1} + c_{S_2}}$$
$$x_{rel,S_2} = x_{M_2} - x_{M_1}$$
$$F_{S_2} = c_{S_2} \cdot x_{rel,S_2}$$

$$\dot{x}_{M_2} = v_{M_2}$$
$$\dot{v}_{M_2} = {}^{1}/_{m_{M_2}} \left(F_{ext} - F_{S_2}\right)$$

Equation-based Model Reduction in OpenModelica Compiler

Behavior of Manually Reduced Model

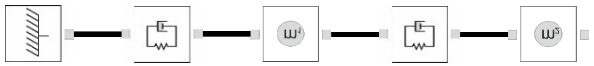




Equation-based Model Reduction in OpenModelica Compiler
Equation-based Model Reduction Algorithm

- User Input:
  - Physical Model (DAE)
  - Reference Scenario
  - Error Boundaries
- Algorithm Output:
  - Ranking List: Terms with low impact first
  - Reduction List: Reduced terms
  - Error of Reduced Model vs. Original Model

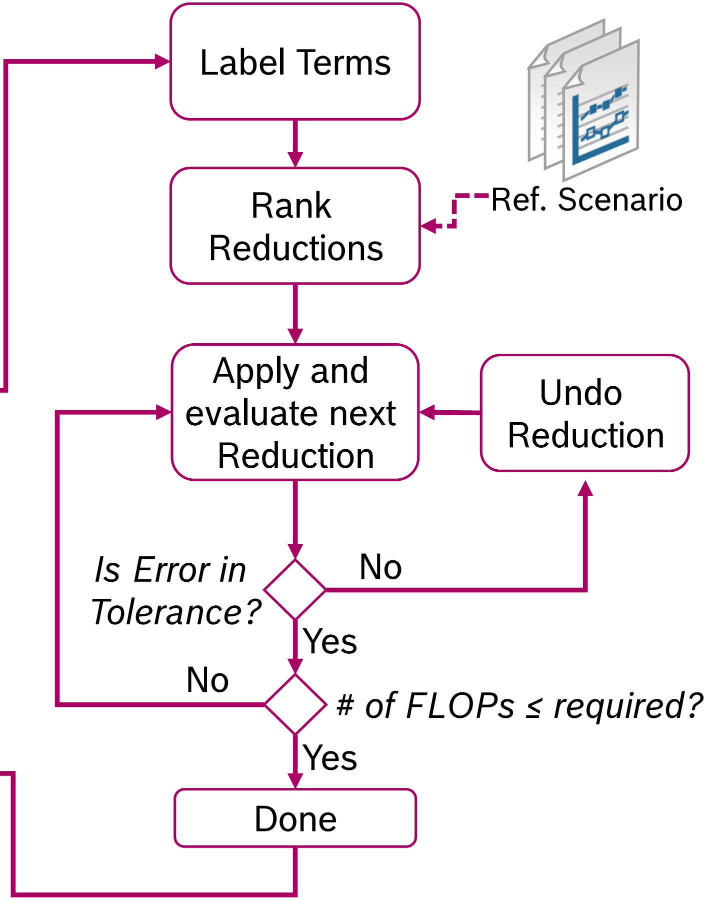
Physical Model



Original Set of Equations

$$
\begin{aligned}
x_{rel,S_1} &= p_1 \cdot x_{M_1} \\
x_{rel,S_2} &= p_2 \cdot x_{M_2} - p_3 \cdot x_{M_1} \\
v_{rel,S_1} &= p_4 \cdot v_{M_1} \\
v_{rel,S_2} &= p_5 \cdot v_{M_2} - p_6 \cdot v_{M_1} \\
F_{S_1} &= p_7 \cdot c_{S_1} \cdot x_{rel,S_1} + p_8 \cdot d_{S_1} \cdot v_{rel,S_1} \\
F_{S_2} &= p_9 \cdot c_{S_2} \cdot x_{rel,S_2} + p_{10} \cdot d_{S_2} \cdot v_{rel,S_2}
\end{aligned}
$$

$$
\begin{aligned}
\dot{x}_{M_1} &= p_{11} \cdot v_{M_1} \\
\dot{v}_{M_1} &= {}^1/_{m_{M_1}} \left(p_{12} \cdot F_{S_2} - p_{13} \cdot F_{S_1}\right) \\
\dot{x}_{M_2} &= p_{14} \cdot v_{M_2} \\
\dot{v}_{M_2} &= {}^1/_{m_{M_2}} \left(p_{15} \cdot F_{ext} - p_{16} \cdot F_{S_2}\right)
\end{aligned}
$$



Reduced Set of Equations

$$
\begin{aligned}
x_{M_1} &= \frac{c_{S_2} \cdot x_{M_2}}{c_{S_1} + c_{S_2}} \\
x_{rel,S_2} &= x_{M_2} - x_{M_1} \\
F_{S_2} &= c_{S_2} \cdot x_{rel,S_2}
\end{aligned}
$$

$$
\begin{aligned}
\dot{x}_{M_2} &= v_{M_2} \\
\dot{v}_{M_2} &= {}^1/_{m_{M_2}} \left(F_{ext} - F_{S_2}\right)
\end{aligned}
$$

Equation-based Model Reduction in OpenModelica Compiler
Reduction Methods

- Reduction Method
  - **Term reduction (elimination)**
  - Linearization
  - Replace with constant
  - Symmetry

- Automated Identification of applicable reductions
  - Estimation of the impact
  - Trade-off between effort and precision
  - Ranking Methods

Lenord, Worschech, Davoudi, Durak, Hartmann, OpenModelica Workshop, Linköping, SE | 02/05/2018

BOSCH

Equation-based Model Reduction in OpenModelica Compiler
Ranking Methods in Comparison

Residual Ranking

+ Computational Effort
+ Implementation
- Precision
- Relative Ranking

One-Step Ranking

- Computational Effort
o Implementation
+ Precision
+ Absolute Ranking

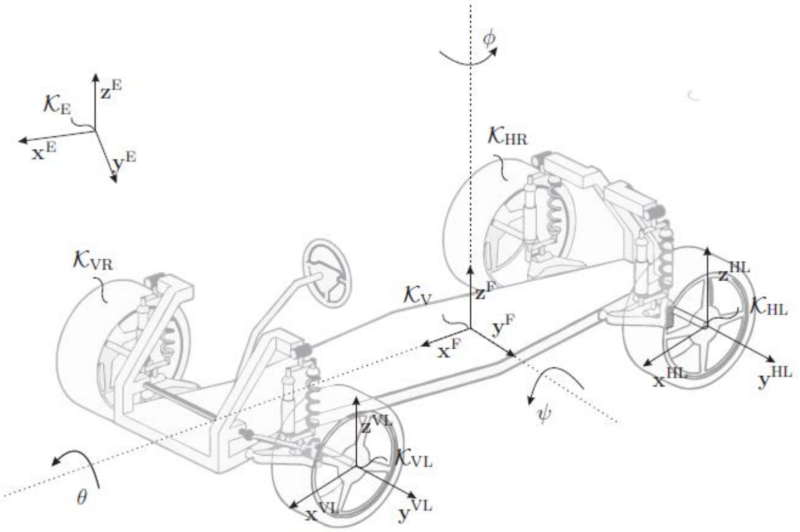
Sensitivity Ranking

+ Computational Effort
+ Implementation
+ Precision
- Relative Ranking
- Applicability

Results of Previous Works

Equation-based Model Reduction in OpenModelica Compiler
Spatial Two-lane Vehicle Dynamics Model

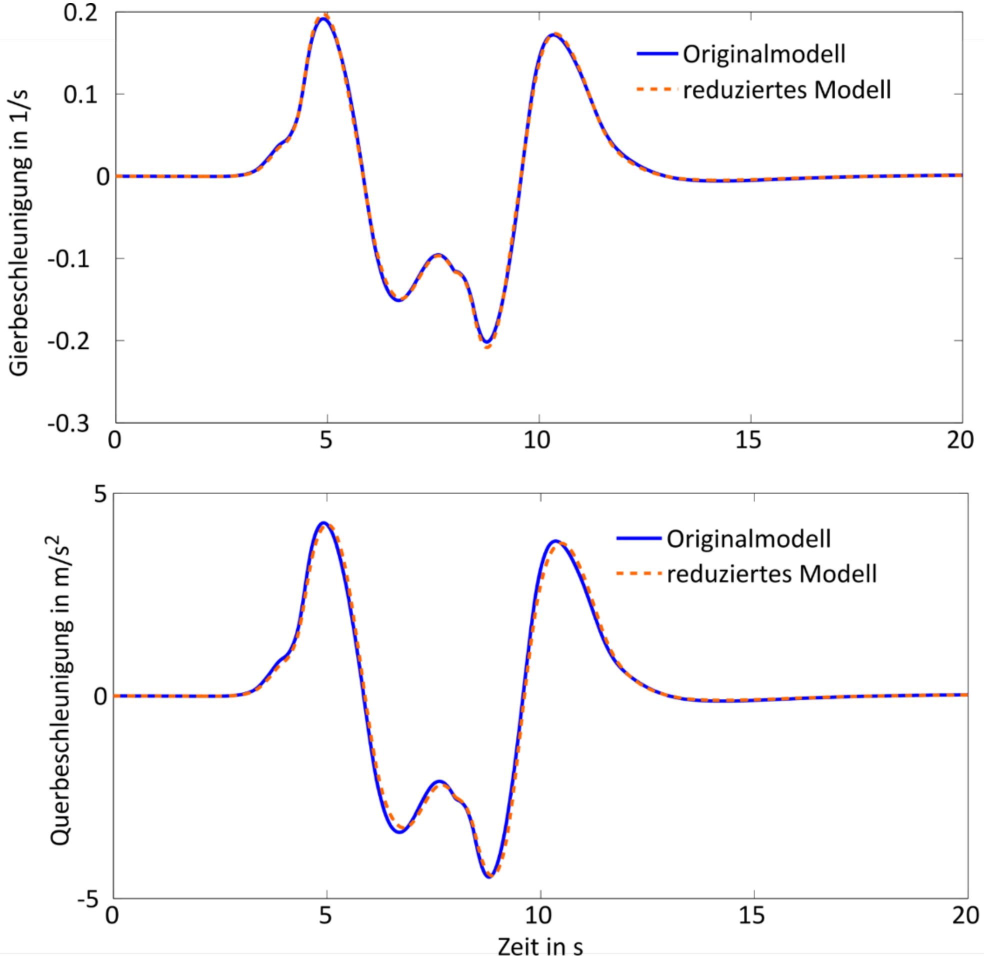
- Spatial Two-lane Vehicle Dynamics Model
  - 30 States
  - Pacejka tire model
- Scenario
  - Double lane change at 40 km/h
  - Outputs: $a_y, \ddot{\psi}$
- Real time target
  - Sony Playstation 2 (10^8 FLOPS)
  - 1ms cycle time → FLOPS per step < 10^5
- Applied methods
  - One-step ranking
  - Term reduction



Mikelsons, Lars, Dieter Schramm, and Hans-Bernd Knoop. 2012. *Generierung vereinfachter Modelle mechatronischer Systeme auf Basis symbolischer Gleichungen*. Duisburg; Essen: Universitätsbibliothek Duisburg-Essen.

Equation-based Model Reduction in OpenModelica Compiler
Spatial Two-lane Vehicle Dynamics Model – Results

- Validation of reduced model
  - Double lane change at 80 km/h.
  - Error slightly larger than 10%.
  - Further reductions lead to known simplified vehicle dynamics models.
  - Reduced model requires 92215 FLOPs per integration step.
  - Speed-up by a factor of 18 compared to the original model.



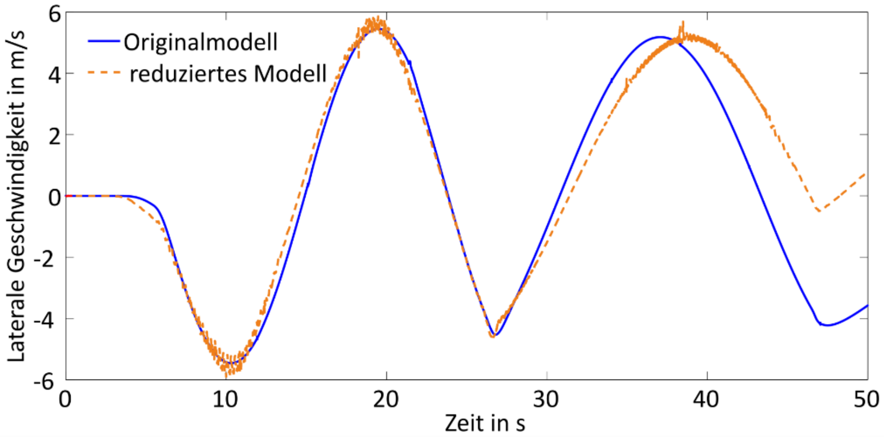
Equation-based Model Reduction in OpenModelica Compiler
Hydro-mechanical Vehicle Model

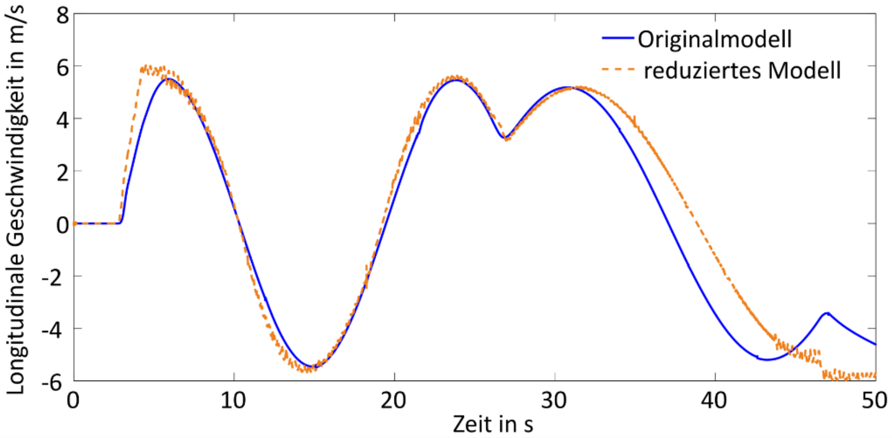
- Spatial Skid-Steer Loader model with hydraulic drive train
  - ~2000 equations in Modelica and
  - 52 states
- Scenario
  - Double loop maneuver
  - Outputs: v_x, v_y
  - Tolerance of the mean square error: $0.005\ ^{m^2}/_{s^2}$
- Applied methods
  - Residual ranking
  - Term reduction

Lars Mikelsons, Hongchao Ji, Thorsten Brandt, and Oliver Lenord. 2009. "Symbolic Model Reduction Applied to Realtime Simulation of a Construction Machine." In , 765–74. doi:10.3384/ecp09430136.

Equation-based Model Reduction in OpenModelica Compiler
Reduction Results

- Undesired oscillations in the velocities.
- Deviations in the vertical dynamics.
- Solution:
  - Apply DFT (Discrete Fourier Transformation) as additional acceptance criteria for the reductions.



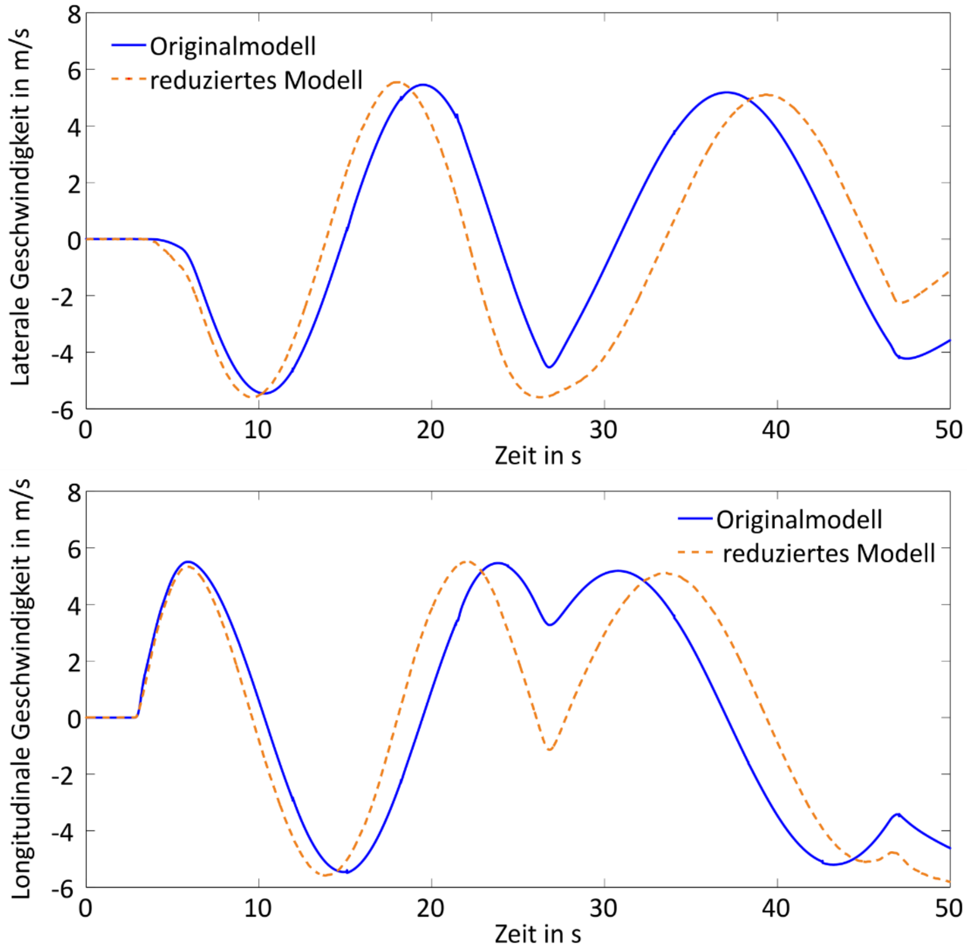


Lenord, Worschech, Davoudi, Durak, Hartmann, OpenModelica Workshop, Linköping, SE | 02/05/2018

Equation-based Model Reduction in OpenModelica Compiler

Results with DFT

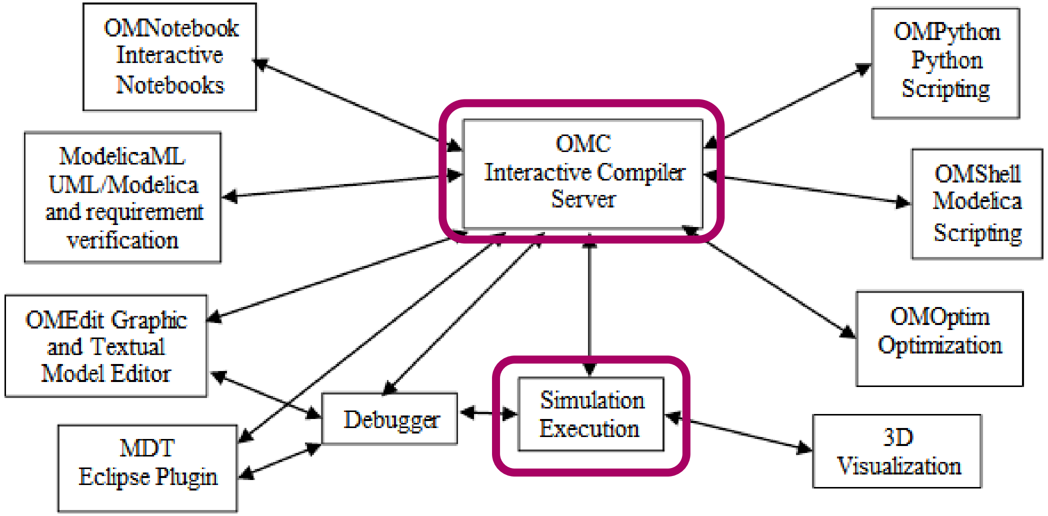
- Oscillations have vanished.
- Physically meaningful reductions are applied by neglecting:
  - Vertical dynamics
  - Pitch and roll dynamics
  - Dynamics of pressure relief valves
- Speed-up by a factor of 6 compared to the original model



Implementation in OpenModelica Compiler

Equation-based Model Reduction in OpenModelica Compiler
Implementation in OpenModelica Compiler

- Available in OpenModelica 1.13.0
  - C++ Runtime with Buffer
  - Compiler Flag: `--labeledReduction`



- mos Script

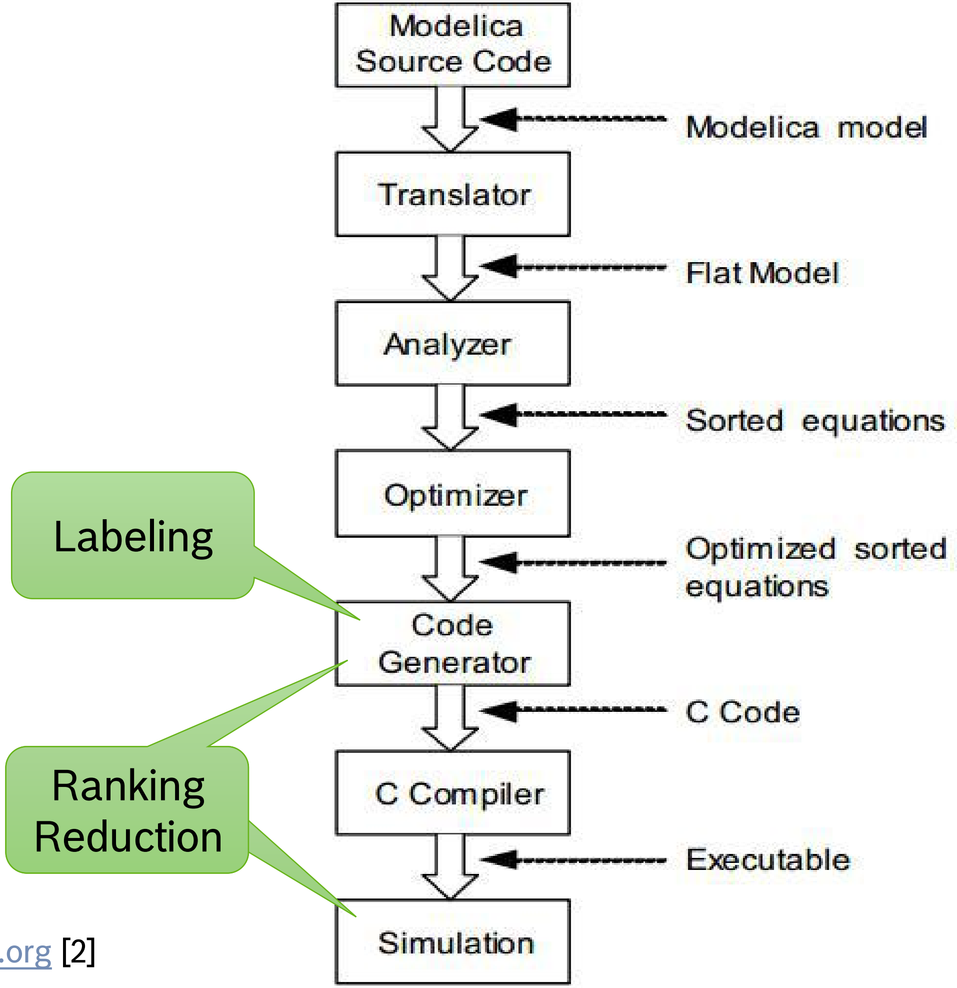
```
setCommandLineOptions("+simCodeTarget=Cpp -d=writeToBuffer --labeledReduction");
getErrorString();
loadModel(Modelica); getErrorString();
loadFile("Test.mo"); getErrorString();

simulate(Test,fileNamePrefix="Test", stopTime = 20.0, outputFormat="buffer"); getErrorString();
```

Source: http://www.openmodelica.org [2]

Equation-based Model Reduction in OpenModelica Compiler

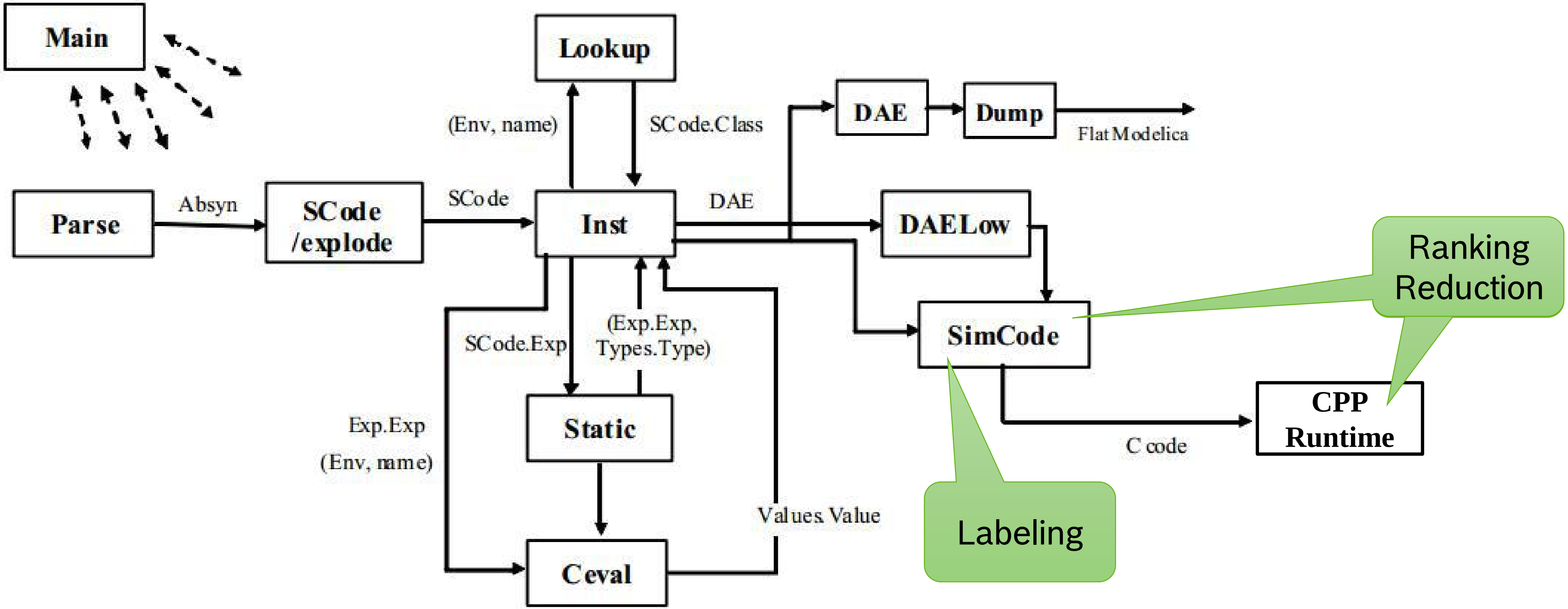
Implementation in OpenModelica

- Labels are added in the Code Generation phase.
- Evaluation and setting of the labels for Ranking and Reduction is managed during the simulation phase.



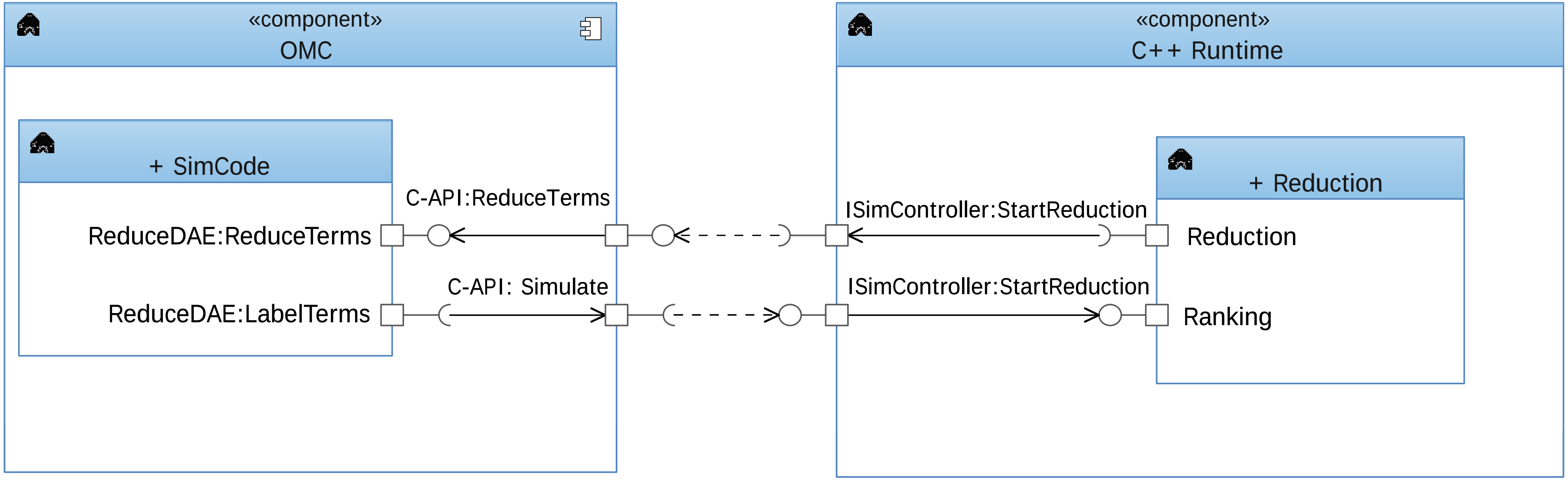
Source: http://www.openmodelica.org [2]

Equation-based Model Reduction in OpenModelica Compiler
Implementation in OpenModelica



Source: http://www.openmodelica.org [2]

Equation-based Model Reduction in OpenModelica Compiler
Implementation in OpenModelica



- OMC includes:
  - Labeling of all terms of an equation.
  - Reduction algorithm to remove selected terms from the equations.
- C++ Runtime includes:
  - Ranking algorithm of all labeled terms
  - Reduction algorithm that checks which terms can be removed from the equations

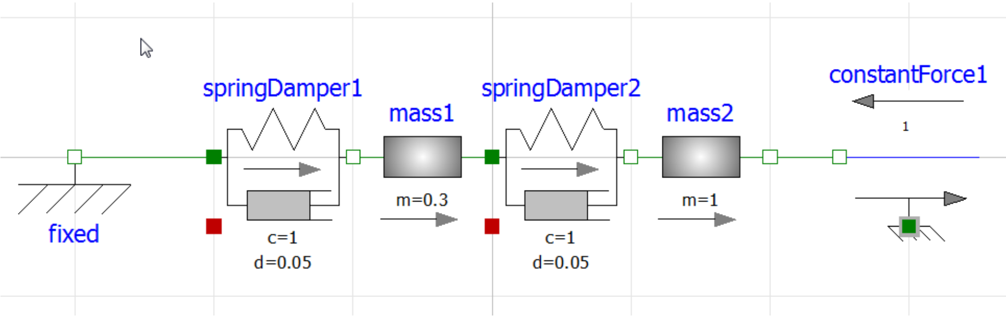
Performance of Current Implementation

Equation-based Model Reduction in OpenModelica Compiler

Performance Test of Current Implementation

- Evaluation using two mass model

```
model Test
 parameter Real m1 = 0.3,m2=1;
 parameter Modelica.SIunits.TranslationalSpringConstant c1 = 1,c2=1;
 parameter Modelica.SIunits.TranslationalDampingConstant d1 = 0.05,d2=0.05;
  Modelica.SIunits.Position x_s1 (start = 0),x_s2 (start = 0);
  Modelica.SIunits.Position x_m1 (start = 0),x_m2 (start = 0);
  Modelica.SIunits.Velocity v_s1 (start = 0),v_s2 (start = 0);
  Modelica.SIunits.Velocity v_m1 (start = 0),v_m2 (start = 0);
  input Modelica.SIunits.Force f(start=1);
  output Modelica.SIunits.Force f1,f2;

equation
  x_s1=x_m1;
  x_s2=x_m2;
  v_s1=v_m1;
  v_s2=v_m2;
  f1=c1*x_s1+d1*v_s1;
  f2=c2*x_s2+d2*v_s2;

  der(x_m1)=v_m1;
  der(v_m1)=(f2-f1)/m1;
  der(x_m2)=v_m2;
  der(v_m2)=(f1-f2)/m2;
end Test;
```



- Assumptions:
  - d, T small
  - $\mathrm{m}_{\mathrm{M}_1} \ll \mathrm{m}_{\mathrm{M}_2}$
- Manually reduced set of equations:

$$x_{M_1} = \frac{c_{S_2} \cdot x_{M_2}}{c_{S_1} + c_{S_2}}$$

$$x_{rel,S_2} = x_{M_2} - x_{M_1}$$

$$F_{S_2} = c_{S_2} \cdot x_{rel,S_2}$$

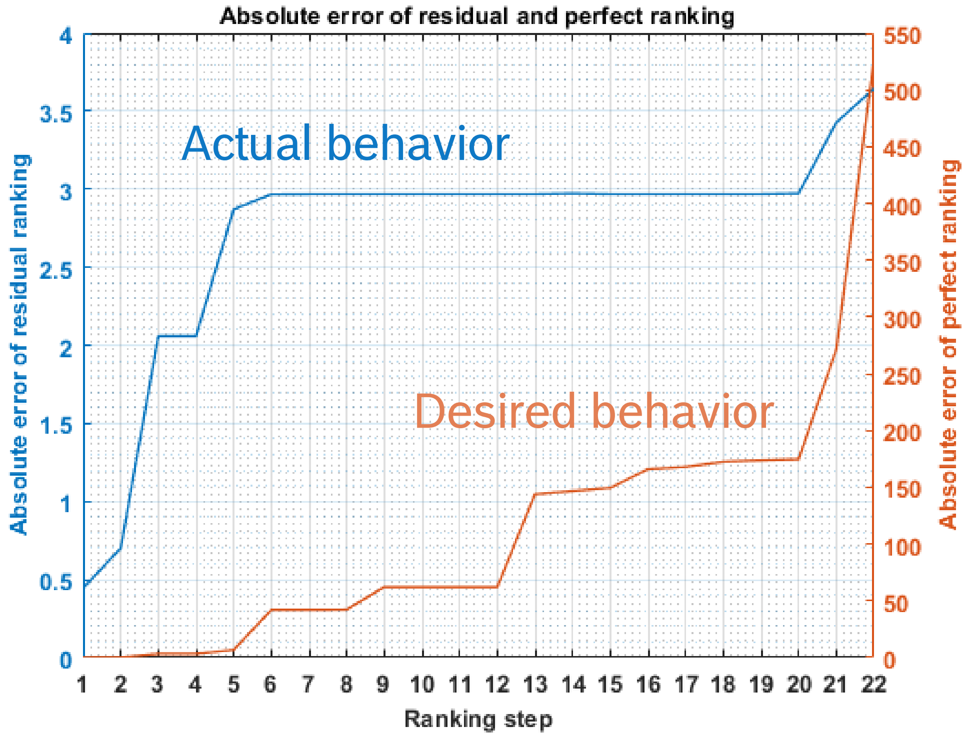
$$\dot{x}_{M_2} = v_{M_2}$$

$$\dot{v}_{M_2} = {}^{1}/_{m_{M_2}} \left(F_{ext} - F_{S_2}\right)$$

Equation-based Model Reduction in OpenModelica Compiler
Ranking Results

Reduction	Labels to zero	v_m2 error
Step 1	8	10,7966
Step 2	11	3,42727
Step 3	12	3,42727
Step 4	14	3,57779
Step 5	15	10,7966
Step 6	0	0
Step 7	0,2	0,494477
Step 8	0,5	0
Step 9	0,5,6	2,80696
Step10	0,5,10	10,7966
Step11	0,5,13	0,106507
Step12	0,5,3	2,8069
Step13	0,5,4	2,8069
Step14	0,5,7	2,15583
Step15	0,5,6	2,8069
Step16	0,5,1	10,7966

Accepted Reductions



- Only reductions with small impact are accepted.
- Deviation of v_{M_2} remains in error bound.

✓ Term reduction works.

✗ Residual ranking is poor.

Equation-based Model Reduction in OpenModelica Compiler
Impact of Ranking on Overall Performance

- ✓ Ranking reduces the reduction effort by more than a factor of three.
- ✓ *Residual Ranking* is twice as fast as the *Perfect Ranking*.
- ✓ *Residual Ranking* does lead to an overall time saving of ~20%.
- ✗ *Residual Ranking* does have a negative impact on achieved speed-up. Not all possible reductions are applied.
- ✗ Overall effort of *Perfect Ranking* is bigger than using *No Ranking*.
- ✗ Based on current implementation it is recommended to use *No Ranking*.

	Perfect Ranking	Residual Ranking	No Ranking
Ranking Effort	460 ms	230 ms	-
Reduction Effort	140 ms	130 ms	450 ms
Overall Effort	600 ms	360 ms	450 ms
Error	0.528	0.103	0.528
Speed-up	1.13	1.04	1.13

BOSCH

Equation-based Model Reduction in OpenModelica Compiler
Reduction Results

- Implemented reduction:

$$
\begin{aligned}
x_{rel,S_1} &= x_{M_1} \\
x_{rel,S_2} &= x_{M_2} - x_{M_1} \\
v_{rel,S_1} &= v_{M_1} \\
v_{rel,S_2} &= \cancel{v_{M_2} - v_{M_1}} \\
F_{S_1} &= c_{S_1} \cdot x_{rel,S_1} + \cancel{d_{S_1} \cdot v_{rel,S_1}} \\
F_{S_2} &= c_{S_2} \cdot x_{rel,S_2} + \cancel{d_{S_2} \cdot v_{rel,S_2}} \\
\dot{x}_{M_1} &= v_{M_1} \\
\dot{v}_{M_1} &= {}^{1}/_{m_{M_1}} \left(F_{S_2} - F_{S_1}\right) \\
\dot{x}_{M_2} &= v_{M_2} \\
\dot{v}_{M_2} &= {}^{1}/_{m_{M_2}} \left(F_{ext} - F_{S_2}\right)
\end{aligned}
$$

- Theoretical reduction:

$$
\begin{aligned}
x_{M_1} &= \frac{c_{S_2} \cdot x_{M_2}}{c_{S_1} + c_{S_2}} \\
x_{rel,S_2} &= x_{M_2} - x_{M_1} \\
F_{S_2} &= c_{S_2} \cdot x_{rel,S_2}
\end{aligned}
$$

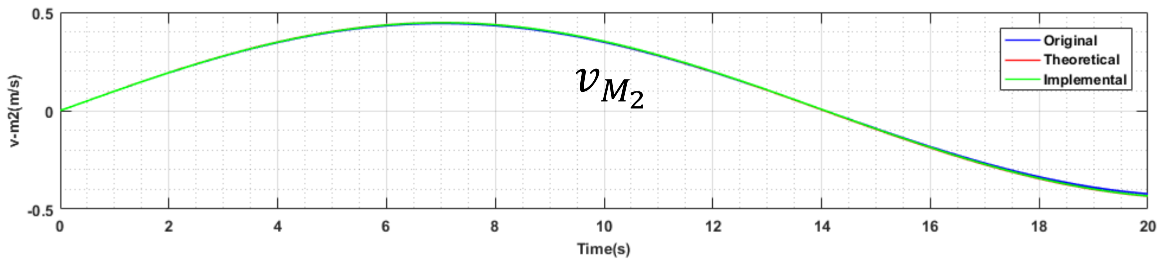
$$
\begin{aligned}
\dot{x}_{M_2} &= v_{M_2} \\
\dot{v}_{M_2} &= {}^{1}/_{m_{M_2}} \left(F_{ext} - F_{S_2}\right)
\end{aligned}
$$

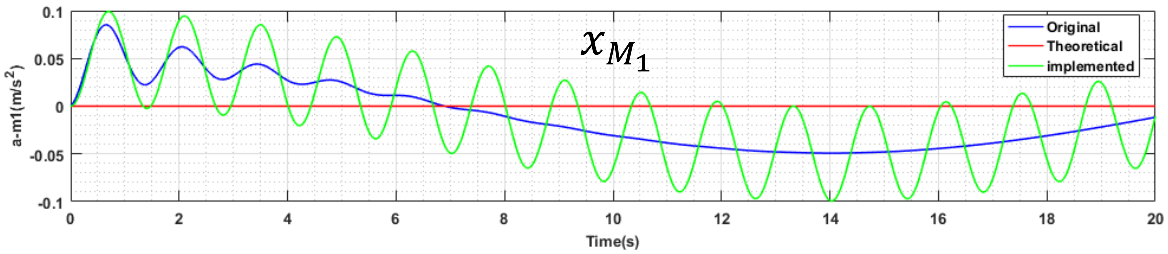
✓ Algebraic terms are removed.

✗ Differential terms have not been removed.

Equation-based Model Reduction in OpenModelica Compiler
Simulation Results

- ✓ Abs. error of output is in bound.
- ✓ 10% speed-up of RHS evaluation by reduction of algebraic terms.
- ✗ Decrease in simulation performance using variable step size solver due to undesired oscillations.
- ✗ Theoretical speed-up by a factor of three is not achieved.





	Original	Implemented Reduction	Theoretical Reduction
Abs. Error v_{M_2}	0	0.103	0.528
Speed-up (evaluation of RHS)	1	1.1	1.1
Speed-up (simulation time dassl)	1	0.6	2.9

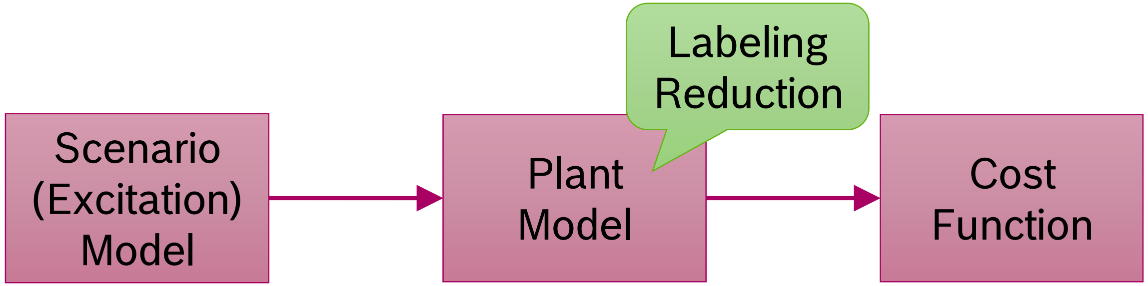
Conclusions & Outlook

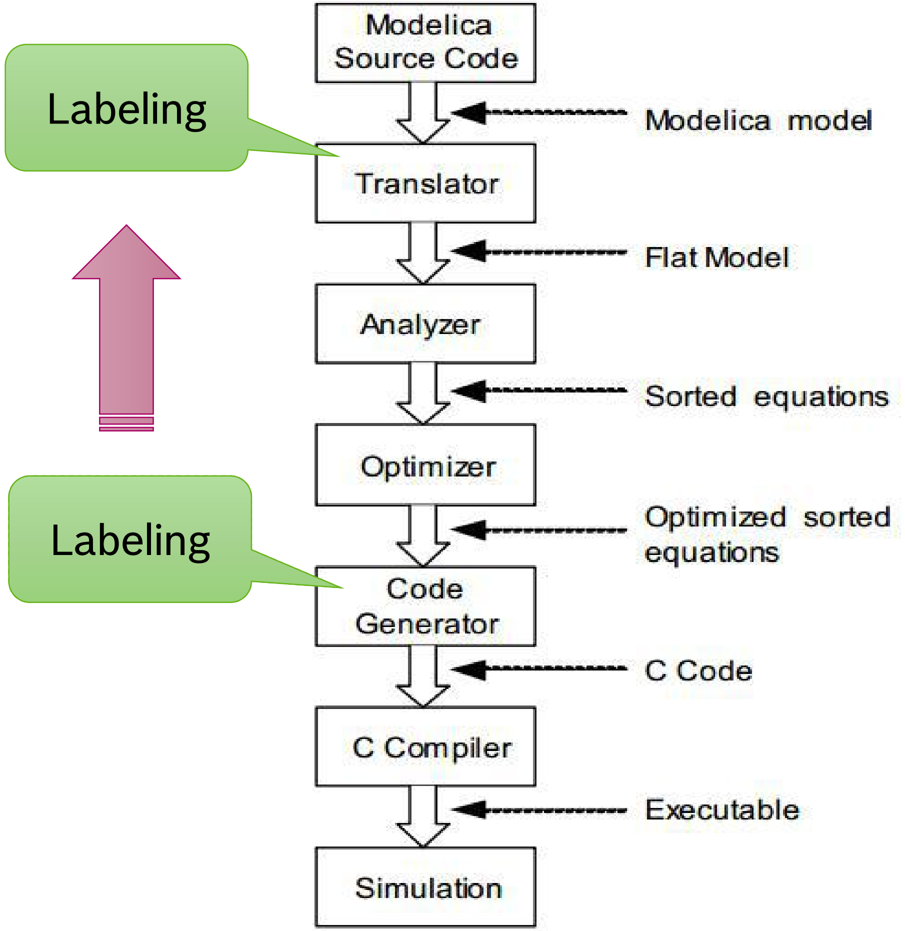
Equation-based Model Reduction in OpenModelica Compiler
Conclusions

- Labeling
  - Labeling of the C Code does not allow symbolic optimization of the reduced equation system
    → No order reduction, no optimized algebraic loops.
  - Proposal: Labeling should be applied to Modelica code or SCode.
- Ranking
  - Sensitivity Analysis or One Step Ranking is expected to give better results than current Residual Ranking.
  - Overall time saving of ranking compared to No Ranking is moderate and therefore not as relevant.
  - Order reduction ("fast states") has biggest impact on speed-up and should be handled at highest priority.
- Reduction
  - Better performance criteria required (e.g. DFT) to avoid undesired oscillations as side effects of term reduction.
  - "Dead code" (evaluation of not output relevant code) should be excluded up-front.

Equation-based Model Reduction in OpenModelica Compiler
Outlook

- Debugging in Visual Studio
  - Fix current crash after merge into master.
- Re-design
  - Definition of scenario (excitation) and error criteria need to be independent of the model to be reduced.
  - Applied reductions must be back traceable and the reduced model should be reusable as Modelica model.
  - Shifting of current implementation to SCode or to an external module (e.g. by using an extended Python API)?





Lenord, Worschech, Davoudi, Durak, Hartmann, OpenModelica Workshop, Linköping, SE | 02/05/2018

Equation-based Model Reduction in OpenModelica Compiler
Outlook

- Additional cost measures need to be defined:
  - Additional error norms considering stiffness, Eigen frequency (DFT)
  - Preserving certain system properties (i.e. stability, passivity, linearity) [1].
  - Computational effort (FLOPs, CPU time) [2].
- Methods of Uncertainty Quantification may be applicable.
- Combination with other reduction approaches to be investigated [3,4].

1) Schilders, Wil. 2008. “Introduction to Model Order Reduction.” *Model Order Reduction: Theory, Research Aspects and Applications*, 3–32.
2) Mikelsons, Lars, Dieter Schramm, and Hans-Bernd Knoop. 2012. *Generierung vereinfachter Modelle mechatronischer Systeme auf Basis symbolischer Gleichungen*. Duisburg; Essen: Universitätsbibliothek Duisburg-Essen.
3) Gu, Chenjie. 2011. “Model Order Reduction of Nonlinear Dynamical Systems.” University of California, Berkeley.
4) Panzer, Heiko, Jan Mohring, Rudy Eid, and Boris Lohmann. 2010. “Parametric Model Order Reduction by Matrix Interpolation.” *At-Automatisierungstechnik Methoden Und Anwendungen Der Steuerungs-, Regelungs-Und Informationstechnik* 58 (8): 475–484.

Equation-based Model Reduction in OpenModelica Compiler

Outlook – Future Use Case: Proper Models for Control Design

- Control Design Models in Physics based Adaptive Controllers*
  - Control Design Model
  - Feed Forward
  - Controller
  - Adaptive Controllers
    - Online Parameter Identification
  - Fault-tolerant Adaptive Controller
    - Observer
    - Diagnostics

➔ Require physical models in a particular equation structure

- Hypothesis:
  - Apply the Model Equation Reduction approach to generate proper models for control design from more detailed system models.

Linear (LIN)

$$\sum_{k=0}^{n} a_k \cdot \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{m} b_k \cdot \frac{d^k u(t)}{dt^k} \qquad n \ge m$$

Linear dynamics and static nonlinearity (STANLIN)

$$\sum_{k=1}^{n} a_k \cdot \frac{d^k y(t)}{dt^k} = \sum_{k=1}^{m} b_k \cdot \frac{d^k u(t)}{dt^k} + f(y(t), u(t)) \qquad n \ge m$$

Static nonlinearity and frictions, linear rest-dynamics (FRICSTANLIN)

$$\sum_{k=1}^{n} a_k \cdot \frac{d^k y(t)}{dt^k} = \sum_{k=1}^{m} b_k \cdot \frac{d^k u(t)}{dt^k} + f(y(t), u(t)) + g\left(\frac{dy(t)}{dt}\right) \qquad n \ge m$$

Fully nonlinear dynamics (NLIN)

$$f\left(y(t), \ldots, \frac{d^n y(t)}{dt^n}, u(t), \ldots, \frac{d^m u(t)}{dt^m}\right) = 0 \qquad n \ge m$$

Thank you for your attention.

Oliver Lenord[1], Niklas Worschech[2],
Fatemeh Davoudi[3], Umut Durak [3], Sven Hartmann [3]

[1] Robert Bosch GmbH, Corporate Sector Research and Advance Engineering,
71272 Renningen, Germany
[2] Bosch Rexroth AG, Engineering Systems and Solutions,
Rexrothstraße 3, 97816 Lohr a.M., Germany
[3] Technical University Clausthal, Department of Computer Science,
Albrecht-von-Groddeck-Straße 7, 38678 Clausthal-Zellerfeld, Germany