EQUATION-BASED MODEL REDUCTION IN OPENMODELICA COMPILER

OPENMODELICA WORKSHOP
05.02.2018, LINKÖPING, SWEDEN

OLIVER LENORD, NIKLAS WORSCHECH, FATEMEH DAVOUDI, UMUT DURAK, SVEN HARTMANN
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
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
Model Order Reduction
Equation-based Model Reduction vs. Model Order Reduction

Equation-based Model Reduction

- 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)

- Based on finite element models.
- Derive very fast and highly accurate compact models\(^1\).

---

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.

---

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

Manual Procedure

Assumptions:
- d, T small
- \( m_{M1} \ll m_{M2} \)

\[
\begin{align*}
x_{rel,S1} &= x_{M1} \\
x_{rel,S2} &= x_{M2} - x_{M1} \\
v_{rel,S1} &= v_{M1} \\
v_{rel,S2} &= v_{M2} - v_{M1}
\end{align*}
\]

\[
\begin{align*}
F_{S1} &= c_{S1} \cdot x_{rel,S1} + d_{S1} \cdot v_{rel,S1} \\
F_{S2} &= c_{S2} \cdot x_{rel,S2} + d_{S2} \cdot v_{rel,S2}
\end{align*}
\]

\[
\begin{align*}
\dot{x}_{M1} &= v_{M1} \\
\dot{v}_{M1} &= \frac{1}{m_{M1}} (F_{S2} - F_{S1}) \\
\dot{x}_{M2} &= v_{M2} \\
\dot{v}_{M2} &= \frac{1}{m_{M2}} (F_{ext} - F_{S2})
\end{align*}
\]
Equation-based Model Reduction in OpenModelica Compiler

Manual Procedure

Assumptions:
- d, T small
- \( m_{M1} \ll m_{M2} \)

\[ x_{rel,S1} = x_{M1} \]
\[ x_{rel,S2} = x_{M2} - x_{M1} \]
\[ v_{rel,S1} = v_{M1} \]
\[ v_{rel,S2} = v_{M2} - v_{M1} \]

\[ F_{S1} = c_{S1} \cdot x_{rel,S1} + d_{S1} \cdot v_{rel,S1} \]
\[ F_{S2} = c_{S2} \cdot x_{rel,S2} + d_{S2} \cdot v_{rel,S2} \]

\[ \dot{x}_{M1} = v_{M1} \]
\[ \dot{v}_{M1} = \frac{1}{m_{M1}} (F_{S2} - F_{S1}) \]
\[ \dot{x}_{M2} = v_{M2} \]
\[ \dot{v}_{M2} = \frac{1}{m_{M2}} (F_{ext} - F_{S2}) \]
Equation-based Model Reduction in OpenModelica Compiler

Manual Procedure

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

$x_{rel,S_1} = x_{M_1}$

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

$v_{rel,S_1} = v_{M_1}$

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

$F_{S_1} = c_{S_1} \cdot x_{rel,S_1}$

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

$\dot{x}_{M_1} = v_{M_1}$

$\dot{v}_{M_1} = \frac{1}{m_{M_1}} (F_{S_2} - F_{S_1})$

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

$\dot{v}_{M_2} = \frac{1}{m_{M_2}} (F_{ext} - F_{S_2})$
Equation-based Model Reduction in OpenModelica Compiler

Manual Procedure

Assumptions:
- $d, T$ small
- $m_{M1} \ll m_{M2}$

\[
x_{M1} = \frac{c_{S2} \cdot x_{M2}}{c_{S1} + c_{S2}}
\]
\[
x_{\text{rel},S2} = x_{M2} - x_{M1}
\]
\[
F_{S2} = c_{S2} \cdot x_{\text{rel},S2}
\]
\[
\dot{x}_{M2} = v_{M2}
\]
\[
\dot{v}_{M2} = \frac{1}{m_{M2}} (F_{\text{ext}} - F_{S2})
\]
Equation-based Model Reduction in OpenModelica Compiler

Behavior of Manually Reduced Model
**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**

\[
\begin{align*}
\dot{x}_{rel,1} &= p_1 \cdot x_{rel,1} \\
\dot{x}_{rel,2} &= p_2 \cdot x_{rel,2} \\
\dot{v}_{rel,1} &= p_3 \cdot v_{rel,1} - p_4 \cdot v_{rel,1} \\
\dot{v}_{rel,2} &= p_5 \cdot v_{rel,2} + \dot{d}_{rel} - \dot{v}_{rel,2} \\
\dot{F}_1 &= p_6 \cdot c_1 \cdot x_{rel,1} + p_7 \cdot \dot{d}_1 + \dot{v}_{rel,1} \\
\dot{F}_2 &= p_8 \cdot c_2 \cdot x_{rel,2} + p_9 \cdot \dot{d}_2 + \dot{v}_{rel,2} \\
\end{align*}
\]

**Reduced Set of Equations**

\[
\begin{align*}
\dot{x}_M &= c_2 \cdot x_M \\
\dot{v}_M &= p_3 \cdot v_M - p_4 \cdot \dot{v}_M \\
\dot{F}_1 &= p_6 \cdot c_1 \cdot x_M + p_7 \cdot \dot{d}_1 + \dot{v}_M \\
\dot{F}_2 &= p_8 \cdot c_2 \cdot x_M + p_9 \cdot \dot{d}_2 + \dot{v}_M \\
\end{align*}
\]

**Algorithm Flow:**

1. **Label Terms**
2. **Rank Reductions**
3. **Apply and evaluate next Reduction**
4. **Undo Reduction**

**Decision Points:**
- Is Error in Tolerance?
  - Yes: **Done**
  - No: **Is # of FLOPs ≤ required?**
    - Yes: **Done**
    - No: **Apply and evaluate next Reduction**

© Robert Bosch GmbH 2016. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
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
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

- 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 $\rightarrow$ FLOPS per step < $10^5$

Applied methods

- One-step ranking
- Term reduction

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: $0.005 \text{ m}^2/\text{s}^2$

- Applied methods
  - Residual ranking
  - Term reduction

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.
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

```plaintext
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](http://www.openmodelica.org) [2]
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](http://www.openmodelica.org) [2]
Equation-based Model Reduction in OpenModelica Compiler
Implementation in OpenModelica

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

```plaintext
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

\[
\begin{align*}
x_s1 &= x_m1; \\
x_s2 &= x_m2; \\
v_s1 &= v_m1; \\
v_s2 &= v_m2; \\
f1 &= c1*x_s1 + d1*v_s1; \\
f2 &= c2*x_s2 + d2*v_s2; \\
\end{align*}
\]

\[
\begin{align*}
\text{der}(x_m1) &= v_m1; \\
\text{der}(v_m1) &= (f2-f1)/m1; \\
\text{der}(x_m2) &= v_m2; \\
\text{der}(v_m2) &= (f1-f2)/m2;
\end{align*}
\]
end Test;
```

▶ Assumptions:
- \(d, T\) small
- \(m_{M1} \ll m_{M2}\)

▶ Manually reduced set of equations:

\[
\begin{align*}
x_{M1} &= \frac{c_{S2} \cdot x_{M2}}{c_{S1} + c_{S2}} \\
x_{rel,S2} &= x_{M2} - x_{M1} \\
F_{S2} &= c_{S2} \cdot x_{rel,S2} \\
\dot{x}_{M2} &= v_{M2} \\
\dot{v}_{M2} &= \frac{1}{m_{M2}} (F_{ext} - F_{S2})
\end{align*}
\]
### Equation-based Model Reduction in OpenModelica Compiler

**Ranking Results**

<table>
<thead>
<tr>
<th>Reduction</th>
<th>Labels to zero</th>
<th>(v_{M2}) error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>8</td>
<td>10,7966</td>
</tr>
<tr>
<td>Step 2</td>
<td>11</td>
<td>3,42727</td>
</tr>
<tr>
<td>Step 3</td>
<td>12</td>
<td>3,42727</td>
</tr>
<tr>
<td>Step 4</td>
<td>14</td>
<td>3,57779</td>
</tr>
<tr>
<td>Step 5</td>
<td>15</td>
<td>10,7966</td>
</tr>
<tr>
<td>Step 6</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Step 7</td>
<td>0,2</td>
<td>0,494477</td>
</tr>
<tr>
<td>Step 8</td>
<td>0,5</td>
<td>0</td>
</tr>
<tr>
<td>Step 9</td>
<td>0,5,6</td>
<td>2,80696</td>
</tr>
<tr>
<td>Step10</td>
<td>0,5,10</td>
<td>10,7966</td>
</tr>
<tr>
<td>Step11</td>
<td>0,5,13</td>
<td>0,106507</td>
</tr>
<tr>
<td>Step12</td>
<td>0,5,3</td>
<td>2,8069</td>
</tr>
<tr>
<td>Step13</td>
<td>0,5,4</td>
<td>2,8069</td>
</tr>
<tr>
<td>Step14</td>
<td>0,5,7</td>
<td>2,15583</td>
</tr>
<tr>
<td>Step15</td>
<td>0,5,6</td>
<td>2,8069</td>
</tr>
<tr>
<td>Step16</td>
<td>0,5,1</td>
<td>10,7966</td>
</tr>
</tbody>
</table>

- Only reductions with small impact are accepted.
- Deviation of \(v_{M2}\) remains in error bound.

- Term reduction works.

- Residual ranking is poor.
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*.

---

### Impact of Ranking on Overall Performance

<table>
<thead>
<tr>
<th></th>
<th>Perfect Ranking</th>
<th>Residual Ranking</th>
<th>No Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ranking Effort</td>
<td>460 ms</td>
<td>230 ms</td>
<td>-</td>
</tr>
<tr>
<td>Reduction Effort</td>
<td>140 ms</td>
<td>130 ms</td>
<td>450 ms</td>
</tr>
<tr>
<td>Overall Effort</td>
<td>600 ms</td>
<td>360 ms</td>
<td>450 ms</td>
</tr>
<tr>
<td>Error</td>
<td>0.528</td>
<td>0.103</td>
<td>0.528</td>
</tr>
<tr>
<td>Speed-up</td>
<td>1.13</td>
<td>1.04</td>
<td>1.13</td>
</tr>
</tbody>
</table>
Equation-based Model Reduction in OpenModelica Compiler

Reduction Results

Implemented reduction:

\[
\begin{align*}
    x_{rel,S_1} &= x_M_1 \\
    x_{rel,S_2} &= x_M_2 - x_M_1 \\
    v_{rel,S_1} &= v_M_1 \\
    v_{rel,S_2} &= v_{M_2} - v_{M_1} \\
    F_{S_1} &= c_{S_1} \cdot x_{rel,S_1} + d_{S_1} \cdot v_{rel,S_1} \\
    F_{S_2} &= c_{S_2} \cdot x_{rel,S_2} + d_{S_2} \cdot v_{rel,S_2} \\
    \dot{x}_M_1 &= v_{M_1} \\
    \dot{v}_M_1 &= \frac{1}{m_{M_1}} (F_{S_2} - F_{S_1}) \\
    \dot{x}_M_2 &= v_{M_2} \\
    \dot{v}_M_2 &= \frac{1}{m_{M_2}} (F_{ext} - F_{S_2})
\end{align*}
\]

- 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.

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Implemented Reduction</th>
<th>Theoretical Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abs. Error $v_{M_2}$</td>
<td>0</td>
<td>0.103</td>
<td>0.528</td>
</tr>
<tr>
<td>Speed-up (evaluation of RHS)</td>
<td>1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>Speed-up (simulation time dassl)</td>
<td>1</td>
<td>0.6</td>
<td>2.9</td>
</tr>
</tbody>
</table>
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)?

---

**Scenario (Excitation) Model** → **Plant Model** → **Cost Function** → **Labeling Reduction**
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 \(^3,^4\).

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.

\[
\sum_{i=1}^{n} a_i \frac{d^ny(t)}{dt^n} = \sum_{i=1}^{m} b_i \frac{d^m u(t)}{dt^m}, \quad n \geq m
\]
Linear (LIN)

\[
\sum_{i=1}^{n} a_i \frac{d^ny(t)}{dt^n} = \sum_{i=1}^{m} b_i \frac{d^m u(t)}{dt^m} + f(y(t), u(t)), \quad n \geq m
\]
Linear dynamics and static nonlinearity (STANLIN)

\[
\sum_{i=1}^{n} a_i \frac{d^ny(t)}{dt^n} = \sum_{i=1}^{m} b_i \frac{d^m u(t)}{dt^m} + f(y(t), u(t)) + g \left( \frac{dy(t)}{dt} \right), \quad n \geq m
\]
Static nonlinearity and friction, linear rest-dynamics (FRICSTANLIN)

\[
f^{\left( y(t), \ldots, \frac{d^ny(t)}{dt^n}, \ldots, \frac{d^n u(t)}{dt^n} \right)} = 0, \quad n \geq m
\]
Fully nonlinear dynamics (NLIN)
Thank you for your attention.

Oliver Lenord¹, Niklas Worschech², Fatemeh Davoudi³, Umut Durak³, Sven Hartmann³

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