Modeling a powertrain in Simscape in a modular vehicle component model library

Stuttgart, 24.09.2015, MBtech, Jörn Bader
Contents

- Introduction – initial situation
  - Driving performance and consumption simulation - Overview
  - Initial powertrain in Simulink

- Switch to a powertrain in Simscape
  - Motivation
  - Short example

- Results and Conclusion
  - Powertrain in Simscape
  - Simulation results and further work
Introduction – initial situation

Driving performance and consumption simulation - Overview
Model structure and additional details

**Model structure:**
- Simulation of powertrain concepts
- Focus on powertrain losses
- No monitoring of powertrain vibrations
- Longitudinal dynamics (one wheel)

**Additional details:**
- ~ 1000 Subsystems with ~ 1000 parameters
- < 1 minute for a NEDC simulation
- > 100 users in several departments with different requirements
Typical results

- Consumption values in l/100km
- Driving performance (e.g. acceleration 0 – 100 kph in seconds)
- Torque flow over time
- Rotation speed over time
Introduction – initial situation

Initial powertrain in Simulink
Powertrain in Simulink – initial structure (schematic)

- Torque ($tq$) flow from Engine to Vehicle body
- Rotation speed ($n$) calculation from Vehicle body back to Engine
- Only one speed calculation in Vehicle body
- Static torque in powertrain
- Dynamic torque calculated at wheel

\[ \dot{\omega} = \frac{tq}{J} \]

\[ n_{whl} = \int \dot{\omega} \]
### Powertrain in Simulink – library structure

- **Library structure**
- **Generic models**
- **Vehicle data for vehicle fleets**
- **Automatic configuration for each vehicle**
  - Model changes
  - Parameter changes

<table>
<thead>
<tr>
<th>Library structure</th>
<th>EngineLib</th>
<th>ClutchLib</th>
<th>EMLib</th>
</tr>
</thead>
<tbody>
<tr>
<td>Engine 1</td>
<td>Clutch 1</td>
<td>Electric Motor 1</td>
<td></td>
</tr>
<tr>
<td>Engine 2</td>
<td>Clutch 2</td>
<td>Electric Motor 2</td>
<td></td>
</tr>
<tr>
<td>Clutch 3</td>
<td></td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

**Example Diagram:**

- Engine
- Clutch
- Electric Motor

**Connections:**

- Engine to Clutch (tq, n)
- Clutch to Electric Motor (tq, n)
Switch to a powertrain in Simscape

Motivation
Motivation for a switch to Simscape

General difficulties for models in Simulink

- Enhancements get a lot more complicated for complex models
- Modifications are hard to accomplish among several developers

Modeling benefits of Simscape

- Simple component interface
  - Less problems with signal names
  - Easier reuse of components
- Physically correct structured modeling
- Easier understanding for users
- Difficult to understand for users
- Lots of adjustments for consistent signal names
Motivation for a switch to Simscape

Numerical benefits of Simscape

- Calculation of dynamic torque throughout the whole powertrain
- Numerically more stable calculation of dynamic torque
  - One system of equations for physical model parts
  - Implicit solver
- Local iterations for physical model parts
- Backward calculation possible
Motivation for a switch to Simscape

Dynamic Torque – detailed view

Simulink problems:

- Dynamic Torque calculated through \( t_q = J \cdot \dot{\omega} \Rightarrow n = \frac{30}{\pi} \int \dot{\omega} \)

- Calculation in every component requires at least 7 integrators

- Calculation explicit / fixed step \(\rightarrow\) high errors or small step size needed

Simscape:

- Implicit solver for stiff parts of the model \(\rightarrow\) More stable calculation of dynamic torque

- Local iterations to minimize simulation errors
Switch to a powertrain in Simscape

Short example
Transformation example – model of a differential

Equations:

\[ n_{Dftl\ln} = i_{Dftl} \cdot n_{Whl} \]

\[ tq_{DftlOut} = \left( tq_{Dftl\ln} - tq_{DftlLoss} \right) \cdot i_{Dftl} \]
Transformation example – model of a differential

```matlab
component differential

inputs
i = { 4, '1' }; % Ratio: left
tqLoss = {0, 'N*m'}; % tqLoss: left
end

nodes
I = foundation.mechanical.rotational.rotational; % I: left
O = foundation.mechanical.rotational.rotational; % O: right
end

parameters
end

variables
i = { 0, 'N*m' }; % i
I.out = { 0, 'N*m' }; % I.out
end

function setup
through( t.in, I.t, [] );
through( t.out, [], O.t );
% Parameter range checking
if i == 0
    pm_error('simscape:NotZero', 'Gear ratio')
end

equations

  t.out == i * (t.in - tqLoss);
  I.w == i * O.w;
end
```

Equations

Simscape
Results and Conclusion

Powertrain in Simscape
Powertrain in Simscape (schematic)

- Physical model parts in Simscape
- One physical network
- Engines as torque sources
- Torque losses calculated in Simulink
- Control parts in Simulink

![Schematic diagram of powertrain model in Simscape](image_url)
Powertrain in Simscape – library structure

- Library structure
- Generic models
- Vehicle data for vehicle fleets
- Automatic configuration for each vehicle
  - Model changes
  - Parameter changes

Diagram shows:
- EngineLib
  - Engine 1
  - Engine 2
- ClutchLib
  - Clutch 1
    - SL
    - SSC
- EMLib
  - Electric Motor 1
  - Electric Motor 2
Results and Conclusion

Simulation results and further work
Correct calculation of dynamic torque

- Vehicle acceleration
- Torque at gearbox output
- Total torque in Simscape including dynamic torque ✔
Results achieved with Simscape

Improved model structure and streamlined modeling process

- Physically more correct powertrain model ✔
- Simple component interface
  - Less problems with signal names ✔
  - Easier reuse of components ✔

Improved accuracy and numerical stability

- Calculation of dynamic torque throughout the whole powertrain ✔
- Numerically more stable calculation of dynamic torque ✔
Conclusion and further work

- Further redesign of the existing model library in Simscape
- Building of new models in Simscape
- Rollout of the Simscape library for all active users
- Backward calculations in Simscape
Thank you