Multibody-System Visualization and Interactive Realtime Simulation of FMUs for OpenModelica based Simulations

Linköping, 01/02/2016
Outline

1. Motivation
2. OMVis - Visualization Tool
3. Interactive Simulator
4. Summary and Outlook
MOTIVATION
Presentation of Results

- numerical values / line plots
- 3D animation
Model Visualization

- get an overall understanding of your model behaviour
- model validation
- communicate more data in a faster, more understandable way
- graphic applications (e.g. simulator)
Simulator Development

Get the most realistic behaviour that is still realtime capable.

- evaluation of FMUs for realtime application
- generic simulator setup
- test different FMUs for realtime ability
  - level of detail
  - optimization, compiler settings
  - parallelization
OMVIS – VISUALIZATION TOOL
Visualization Base Data

**Simultaneous animation**
- scene update during simulation run
- state-of-the-art for commercial tools
- network communication during simulation

**Result file based animation**
- visualization after completed simulation run
- scene description needed

**FMU based animation**
- scene description needed
- not specified in standard
- currently no public solution available (?)
Scene Description

Shapes from:
OMVis - Visualization Tool

Structure

- OpenModelica Compiler
  - Attribute Values: model_res.mat
  - Scene Description: model_visual.xml
  - Interactive Simulation: model.fmu

- XML Parsing: rapidxml.hpp
- FMU Integration: FMI Library
- Input Handling: SDL
- GUI: Qt

- 3D Visualization: OpenSceneGraph
Workflow

OMVis - Visualization Tool

Realtime

Visualization step:
- Get attributes for shapes (fmu, mat)
- Transformation matrix (4x4)
- Update scene

Simulation step:
- Set inputs
- Get derivatives
- Integrate

$t_{vis}$
- $h_{vis}$
- $t_{sim}$
- $h_{sim}$
- $t_{vis} = t_{vis}$
- $t_{sim} = t_{vis}$

Linköping 01/02/2016
Multibody-System Visualization
INTERACTIVE SIMULATOR
Realtime Criteria

- simulation of a time range $\Delta t$ has to be faster than realtime $\Delta t$

$\rightarrow$ deterministic execution time of system computation and time integration:
  - no event iteration (or fixed number of iterations)
  - no nonlinear systems (or fixed number of iterations)
  - no implicit time integration methods (or fixed number of iterations)
  - no order or step size control (fixed step size)

In practice:
  everything is allowed as long as the real time criteria is fulfilled
Encounter Realtime Requirements

**model adaptation:**
- linearization, stiffness, complexity reduction

**system computation:**
- tearing, reshuffling, partial function evaluation, common-sub-expression-elimination, backend-evaluation of linear torn systems
- evaluation of parameters, compute outputs only
- parallelization

**time integration:**
- multirate, multimode, inline integration

**hardware**
Excavator model
Summary

- Implementation of a visualization tool
- either result file visualization or fmu-based, interactive visualization
- scene description XML file, generated by OpenModelica Compiler
- generic simulator set up to evaluate FMUs for interactive simulators
Outlook

- further enhancement of OMC to perform automated realtime optimizations (multirate integration with static partitioning)

- extend OMVis for missing functionalities (missing geometric primitives, modelica visualization lib?)

- FMU visualization standard ?
Thank you for your attention.

»Wissen schafft Brücken.«

Volker Waurich
volker.waurich@tu-dresden.de