Numerical Challenges in continuous-time Co-Simulation

AVL German Simulation Conference, Ulm, 17.10.2018

Dr. Martin Benedikt
VIRTUAL VEHICLE Research Center
Graz, Austria
ViF/AVL Strategic Cooperation

Co-Simulation

Seed Action

Innovation Research @ VIRTUAL VEHICLE

Integrated & Open Development Platform
Connect simulation & test

Connect existing elements within the vehicle development process for early, cost-saving decisions.

Use Cases

Simulation Models

Comparable Results

2004 2010 2011 2014

2018 / Benedikt
AVL German Simulation Conference / Ulm
Why Co-Simulation?

- Many sub-systems impact energy efficiency of aggregate system
- Higher integration levels of multiple mechatronic systems required
- Integration on controls and physics
- Sub-systems modelled in a plurality of legacy authoring tools
- Co-simulation supports the use of legacy tools with its dedicated solvers
- Co-simulation supports the use of IP sensitive simulation solutions

→ Co-simulation environment must be based on least common denominator

Non-iterative co-simulation is the viable industrial solution!
Two perspectives ...

Technical view

- Multi-domain development
- Multi-tool approach
- Multi-vendor
- Dynamic coupling
  - Virtual prototype representation

Mathematical view

Hybrid Electric Vehicle

\[ F(x, y, Dy, D^2y, \ldots, D^n y) = 0 \]

\[ x = (x_1, \ldots, x_m) \]

\[ y = (y_1, \ldots, y_k) \]

- Multi-method
- Multi-solver
- Multi-rate
- Dynamic coupling
  - Coupling error
Continuous-time Co-Simulation Challenge

- ABS Scenario: “Full braking after acceleration to 100 km/h”

→ Significantly shorter braking distance by accurate coupling!
Non-iterative Co-Simulation

Aliasing effects due to sampling

Estimation error due to extrapolation

Discontinuities at coupling time instants
Problem: Aliasing

- Effect of sampling a continuous signal in frequency domain

\[ Y_s(j\omega) = \frac{1}{T_s} Y(j\omega) \]
\[
+ \frac{1}{T_s} \sum_{n=1}^{\infty} Y(j\omega \pm jn\omega_s) \]

- No Aliasing
- Aliasing

sampling time (macro step size)
Problem: Estimation Error

- Modelling of the transfer behaviour of the coupling process

\[
H(s) = \frac{\hat{y}(s)}{y(s)} = 1 - e^{-s\Delta T} \quad (s \Delta T)
\]

\[
H(j\omega) = \frac{2\pi \sin(\omega \Delta T/2)}{\pi \omega} e^{-j\omega \Delta T/2} = \overline{H(\omega)} e^{-j\omega \Delta T/2}
\]

(time-delay introduced)
Solution Approach - NEPCE

\[ y(t) \xrightarrow{\text{Extrapolation step-size } \Delta T} \hat{y}(t) \xrightarrow{\text{Compensation System}} \hat{y}_c(t) \xrightarrow{\epsilon(t)} c(t) \]

**effective bandwidth**
Non-iterative Co-Simulation

Embedding of “anti-aliasing” filter

Embedding of “smoothing” filter

effective bandwidth
Frequency consideration & stiffness

- Bandwidth of coupling signal (system dynamics) determines both, step-size of the underlying solver as well as the macro-step-size.
Figure 1: Excitation of high dynamics from the (zero order hold) extrapolated input signal $w_{bar}(t)$, over the stiff subsystem response $T(t)$ to the corrected output signal $T_c(t)$. 
Motivation

- Hidden co-simulation functionality within available tools
- FMUs generated for integration into 3rd party tools, e.g. GT, Cosimate, etc.
- Demonstration of Model.CONNECT benefits

Example: **Stiff Spring Damper Mass Cascade**

* M. Benedikt, E. Drenth; Relaxing Stiff System Integration by Smoothing Techniques for Non-Iterative Co-Simulation, IUTAM Symposium 2018, Darmstadt
Reference Simulation and FMU integration

1) Reference Model in Simulink
2) Exported as FMU (via FMI.Lab)
3) Integration of FMU’s
   - Matlab R2017b
   - Model.CONNECT
Reference Simulation and FMU integration

Speed Output RMC5 u2_w

- Monolithic Simulation
- FMU Integration MATLAB
- FMU Integration Model.CONNECT

Exact steady state result

Coupling Error
Outlook / up-coming approaches

FMI for “Co-Simulation” (10/2010)

2018 / Benedikt
AVL German Simulation Conference / Ulm 21
„Directional Derivatives“ Utilization

... utilization of linearized subsystems (!) by master algorithms

→ enables linearly-implizit (explizit) master algorithms (internal states needed)
→ enables interface-jacobian-based master algorithms (pre-step stab.)
Directional Derivatives improve stability behaviour in general!

Model.CONNECT’s pre-step algorithm outperforms linear-implicit schemes!

The graphs show the performance comparison between different algorithms for a monolithic solution, ZOH, NEPCE, pre-step stabilization, and linear-implicit stabilization. The x-axis represents time, and the y-axis represents the deviation from a reference value. The plots demonstrate how each method performs over time, with Model.CONNECT’s pre-step algorithm showing superior stability compared to linear-implicit schemes.
Conclusion

- Co-simulation is performed because of rather complex subsystems simulations, inherently representing stiff system simulations.
- ... and, iterating over macro-steps and extracting model-information is practical (almost and till now) impossible.

- Model.CONNECT solutions
  - relaxing stiffness for (model-free) non-iterative schemes
  - out performs classical linear-implicit non-iterative schemes
Thanks for your attention!

Dr. Martin Benedikt
Head of ‘Co-Simulation and Software’
VIRTUAL VEHICLE Research Center, Graz, Austria

martin.benedikt@v2c2.at