Interface-Jacobian Based Co-Simulation Algorithm (IJCSA)

Hilding Elmqvist
Hans Olsson
Karl Wernersson

Content

- Cosimulation with variable step size
- Stability using Interface Jacobian
- Step Size Control
- Possibility for parallelization
- Example: Robot with 8 FMUs

Sub-model equations

$$\dot{x}(t) = f(x(t), u(t))$$

Differential equation

$$y = g(x(t))$$

$$x(t_n) = x_n$$

Initial condition for macro step(n)

$$u(t) = u_n + \frac{(u_{n+1} - u_n)}{(t_{n+1} - t_n)}(t - t_n)$$

Assumed ramped input given u_{n+1}

$$y_{n+1} = y(t_{n+1})$$

Output at end-point of macro step

Solution can be written as follows when u(t) is given as above
(F is the evolution operator for one macro step, approximated by solver+model equations, f):

$$x_{n+1} = F(x_n, u_n, u_{n+1})$$

$$y_{n+1} = g(F(x_n, u_n, u_{n+1}))$$

Including direct term:

$$y_{n+1} = g(F(x_n, u_n, u_{n+1}), u_{n+1})$$

Coupled systems

$$y^i_{n+1} = g^i(F^i(x^i_n, u^i_n, u^i_{n+1}))$$

Superscript is sub-system number, subscript macro step

Two systems coupled with all inputs and outputs

$$y^1_{n+1} = g^1(F^1(x^1_n, u^1_n, u^1_{n+1}))$$
$$y^2_{n+1} = g^2(F^2(x^2_n, u^2_n, u^2_{n+1}))$$
$$y^2_{n+1} = u^1_{n+1}$$
$$y^1_{n+1} = u^2_{n+1}$$

Solve as a non-linear system of equations; or just use predictor-corrector without iterations

$$\begin{bmatrix} I & -\frac{\partial g^2}{\partial x^2} \cdot \frac{\partial F^2}{\partial u^2} \\ -\frac{\partial g^1}{\partial x^1} \cdot \frac{\partial F^1}{\partial u^1} & I \end{bmatrix} \cdot \begin{bmatrix} \Delta u^1 \\ \Delta u^2 \end{bmatrix} = \begin{bmatrix} g^2(F^2(,,u^2)) - u^1 \\ g^1(F^1(,,u^1)) - u^2 \end{bmatrix}$$

Stabilization

- Input ramp **implicitly** defined, gives stability
- Interface Jacobian for efficient solution

$$\frac{\partial g^i}{\partial x^i} \cdot \frac{\partial F^i}{\partial u^i}$$

  - FMU provides parts: $\frac{\partial g^i}{\partial x^i}$ $\frac{\partial f^i}{\partial x^i}$ $\frac{\partial f^i}{\partial u^i}$
  - Master co-ordinates
  - Approximated $\frac{\partial F^i}{\partial u^i}$
  - Approximate solution of nonlinear system
  - Smaller dimension of nonlinear system – independent of size(x^i)
- Algebraic loops handled
- Less communication if Interface-Jacobian would be included in FMI

Step Size Control

- Make macro step and 2 half macro steps
- Usual step-size control
  - Based on local order (k) for input
  - Limiters (θ_{max}, θ_{min}), and safety factor (α).
- Error estimated using **Richardson-extrapolation**
  - RMS (Root mean-square) error
  - Mixed absolute (Atol) and relative (Rtol) tolerance

$$H_{new} = \min\left\{\theta_{\max}, \max\left\{\theta_{\min}, \alpha\left(\frac{1}{err}\right)^{\frac{1}{k+1}}\right\}\right\} H_{old}$$

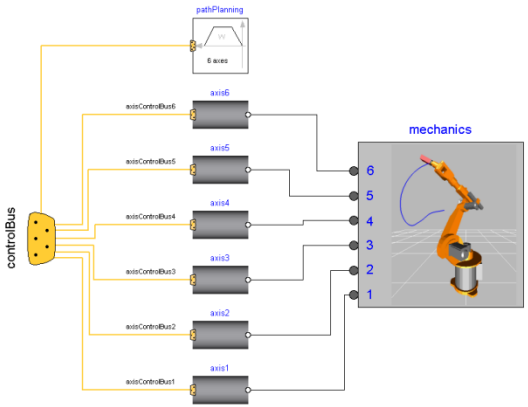
$$err = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(\frac{\frac{Y_j - \hat{Y}_j}{2^{k+1}-1}}{Atol_j + Rtol_j\left|Y_j\right|}\right)^2}$$

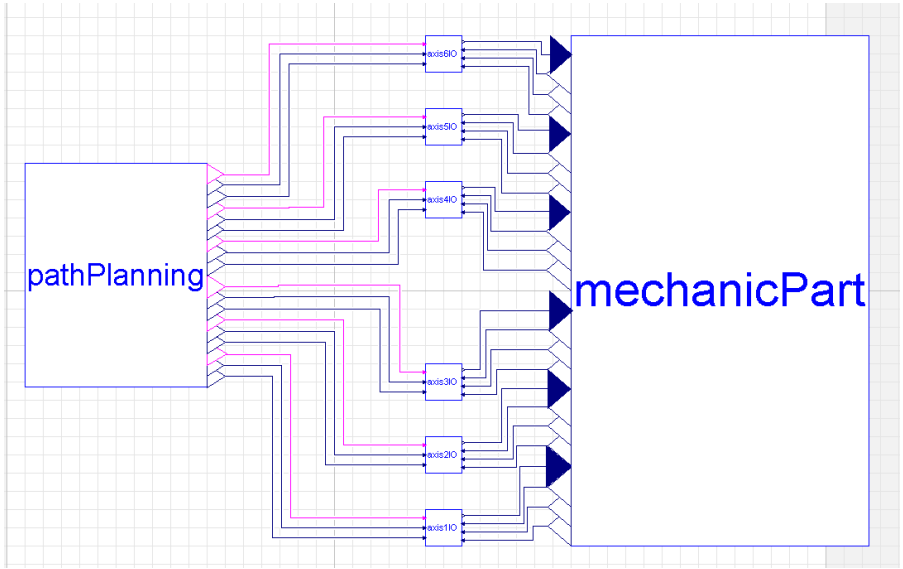
Possibility for parallelization

- Assume 4 FMUs
- Possible speed-up: about 8

Example: MSL Robot partitionend into 8 FMUs

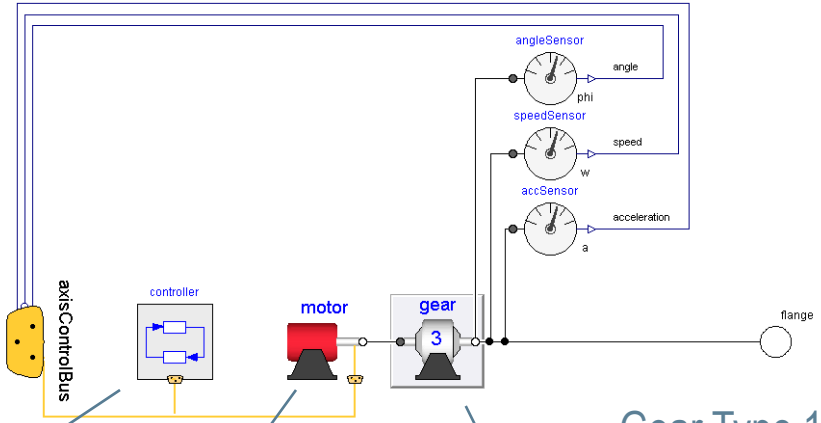
Acausal couplings and algebraic loops between mechanical and three axes

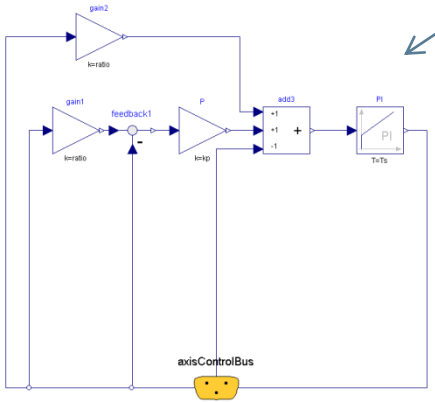


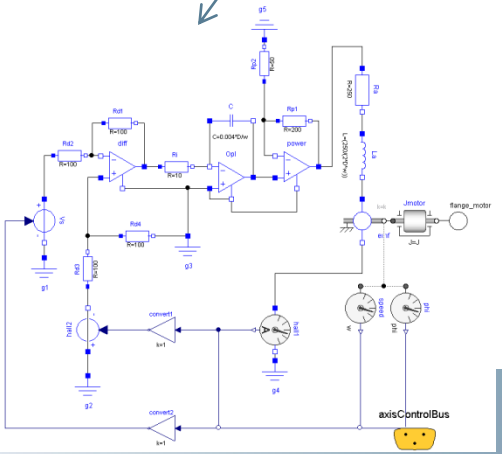


Axes

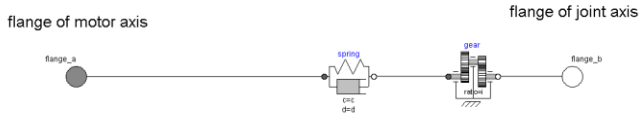
- Friction removed
- Waiting for Hybrid Cosimulation



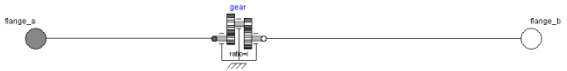




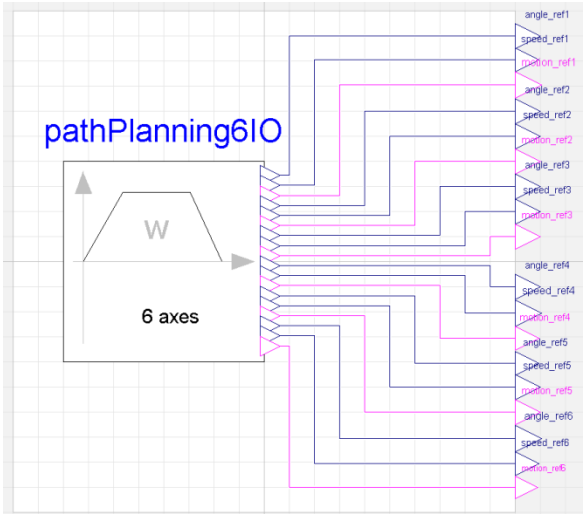
Gear Type 1 no algebraic loop

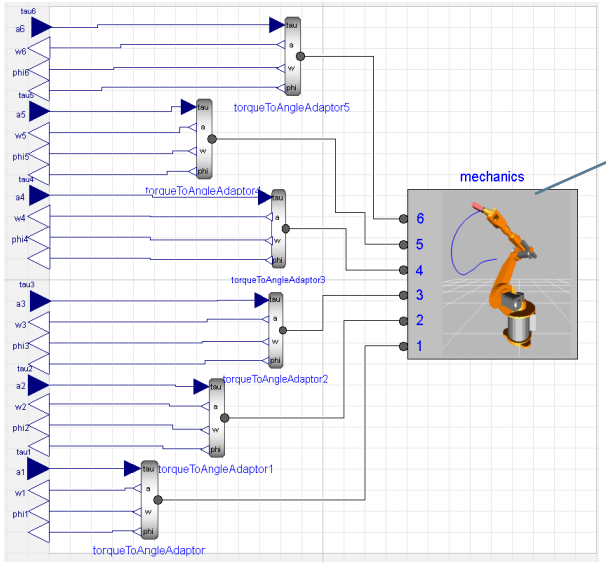


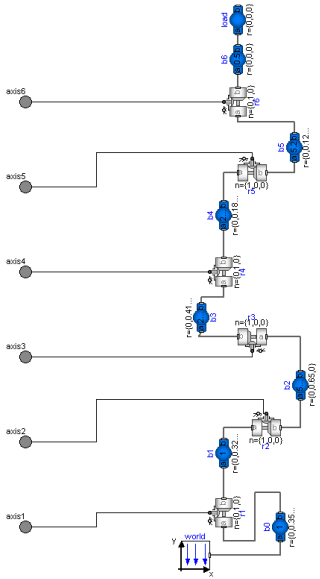
Gear Type 2 algebraic loop



Path Planing and mechanical

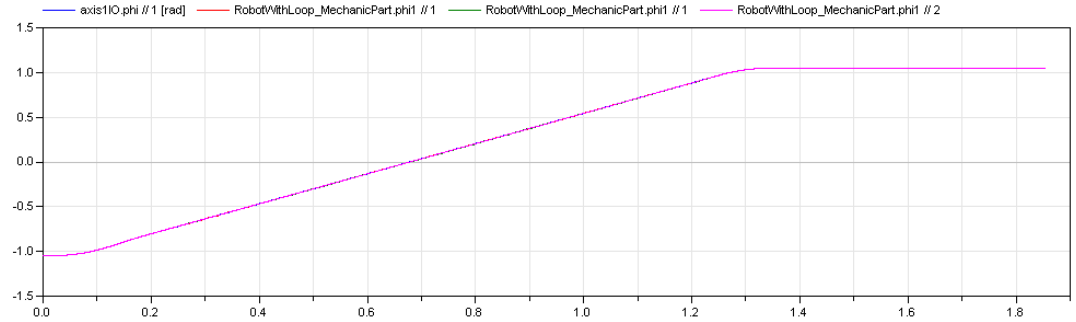




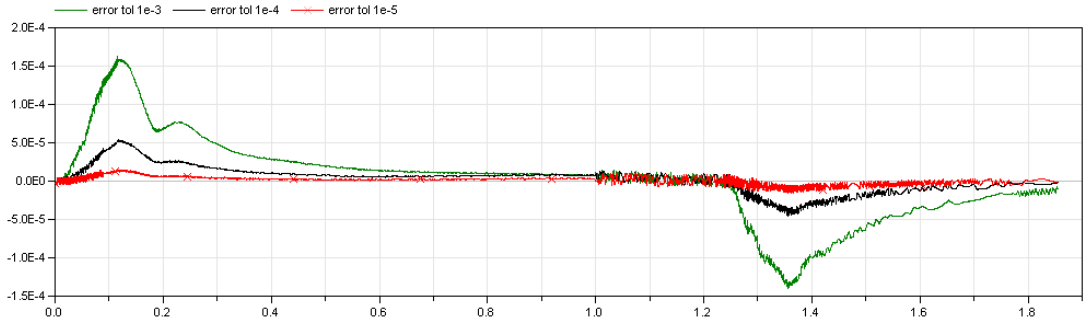


Result of co-simulation of the 8 FMUs

Phi1

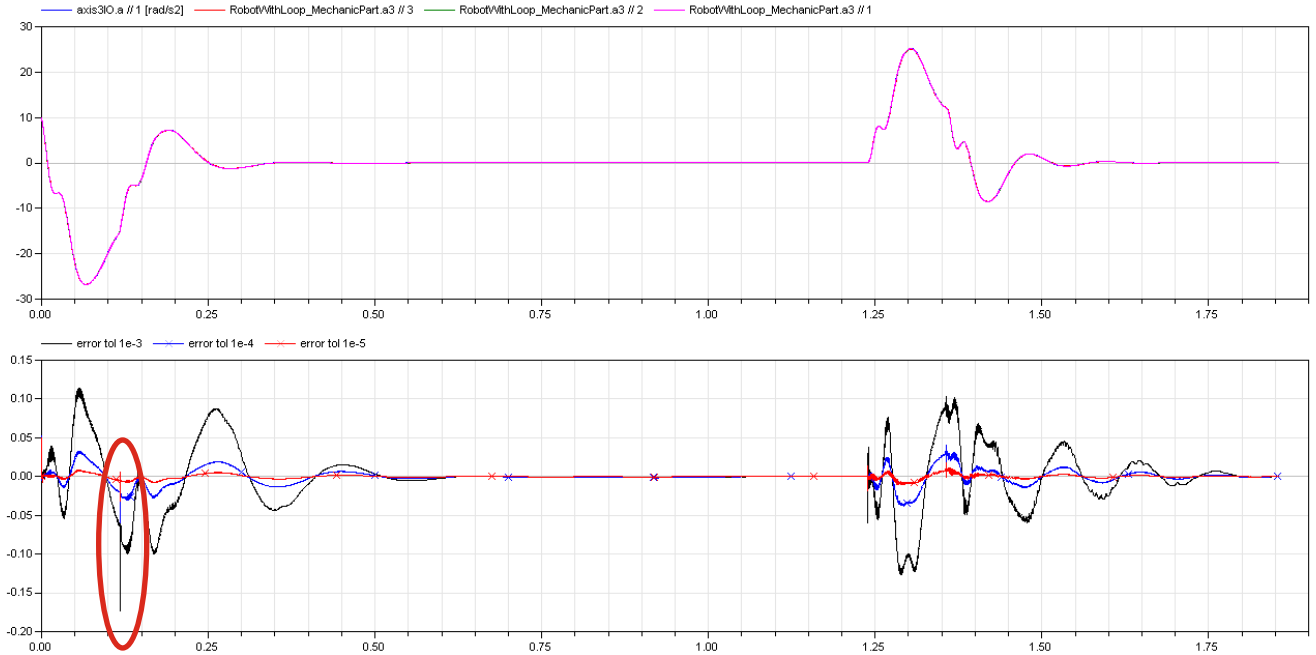


Error



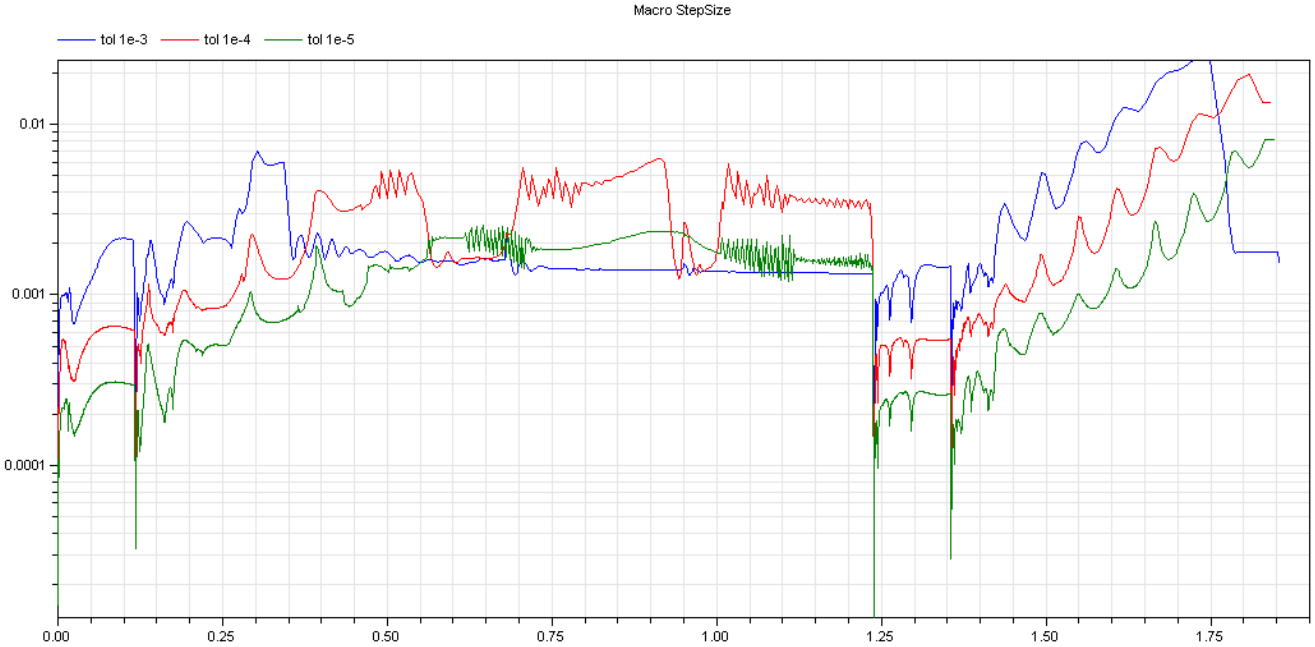
Result of co-simulation of the 8 FMUs

a3 (direct dependency variable)



Result of co-simulation of the 8 FMU's

Macro step size



Reference

- **Interface Jacobian-based Co-Simulation**
  - S. Sicklinger, V. Belsky, B. Engelmann, H. Elmqvist, H. Olsson, R. Wüchner, K.-U. Bletzinger
  - International Journal for Numerical Methods in Engineering
  - March 3, 2014
- http://onlinelibrary.wiley.com/doi/10.1002/nme.4637/references

Needs

- Structured acausal ports
- Event handling – HybridCosimulation
- Optional: Interface Jacobian