Towards an Object-oriented Implementation of von Mises' Motor Calculus Using Modelica

2nd International Workshop on Equation-based Object-oriented Languages and Tools,
Paphos, Cyprus, July 8, 2008

Tobias Zaiczek, Olaf Enge-Rosenblatt

Fraunhofer-Institut Integrierte Schaltungen
Branch Lab Design Automation,
Dresden, Germany

Fraunhofer Institut Integrierte Schaltungen

Contents

Contents

1. Introduction

Current situation

- description of the behaviour of multi-body systems is not an easy task
- Modelica Multibody Standard Library is a well-designed tool
- equations of motions are hard to read and understand

Idea

- usage of motor calculus proposed by Richard von Mises in 1924
- make equations easier to understand

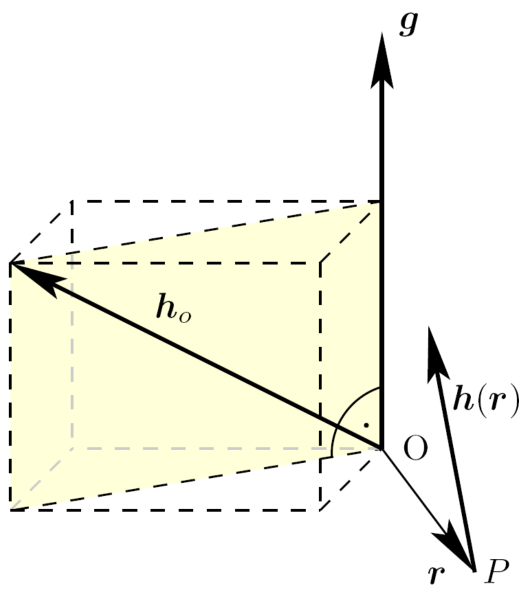
What did we do?

- first phase: implementation of motor calculus by extending Modelica Multibody Standard Library
- approach corresponds with the object-oriented paradigm
- not equation-based to its full sense because of missing operator overloading possibilities

Contents

2. Motor calculus

2.1 Fundamentals



A motor

$$\mathfrak{h} = \begin{pmatrix} \boldsymbol{g} \\ \boldsymbol{h}_o \end{pmatrix}$$

can be represented by an ordered pair of vectors $\boldsymbol{g}$ and $\boldsymbol{h}_o$ defining a vector field in the three-dimensional space:

$$\boldsymbol{h}(\boldsymbol{r}) = \boldsymbol{h}_o + \boldsymbol{g} \times \boldsymbol{r}$$

$\boldsymbol{h}_o$: moment vector at the reference point O

$\boldsymbol{g}$: resultant vector

$\boldsymbol{r}$: position vector for (any) point P

$\boldsymbol{h}$: moment vector for point P

Fraunhofer Institut Integrierte Schaltungen

2. Motor calculus

Fundamental algebraic definitions

addition:

$$\mathfrak{h}_1 + \mathfrak{h}_2 = \begin{pmatrix} \boldsymbol{g}_1 + \boldsymbol{g}_2 \\ \boldsymbol{h}_{o1} + \boldsymbol{h}_{o2} \end{pmatrix}$$

multiplication with a scalar:

$$\alpha\mathfrak{h} = \begin{pmatrix} \alpha\boldsymbol{g} \\ \alpha\boldsymbol{h}_o \end{pmatrix} \qquad \alpha \in \mathbb{R}$$

dot or inner product:

$$(\mathfrak{h}_1, \mathfrak{h}_2) = (\boldsymbol{g}_1, \boldsymbol{h}_{o2}) + (\boldsymbol{g}_2, \boldsymbol{h}_{o1})$$

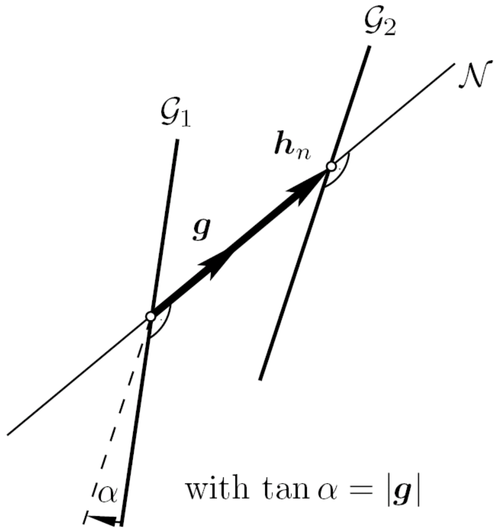
cross or outer product:

$$\mathfrak{h}_1 \times \mathfrak{h}_2 = \begin{pmatrix} \boldsymbol{g}_1 \times \boldsymbol{g}_2 \\ \boldsymbol{g}_1 \times \boldsymbol{h}_{o2} + \boldsymbol{h}_{o1} \times \boldsymbol{g}_2 \end{pmatrix}$$

multiplication with a dyad $\mathfrak{D}$:

$$\mathfrak{D} \circ \mathfrak{h}_1 = \begin{pmatrix} \boldsymbol{D}_{11}\boldsymbol{h}_{o1} + \boldsymbol{D}_{12}\boldsymbol{g}_1 \\ \boldsymbol{D}_{21}\boldsymbol{h}_{o1} + \boldsymbol{D}_{22}\boldsymbol{g}_1 \end{pmatrix}$$

2.2 Geometrical interpretation of motors



- can be represented geometrically by an ordered pair of straight lines $(\mathcal{G}_1, \mathcal{G}_2)$
- all mathematical operations interpretable as geometrical constructions
- $\mathcal{N}$… motor axis = common normal of $\mathcal{G}_1$ and $\mathcal{G}_2$
- $\boldsymbol{h}_n$… moment of the motor on the motor axis, connects $\mathcal{G}_1$ and $\mathcal{G}_2$ along $\mathcal{N}$
- $\boldsymbol{g}$ represents the rotation of $\mathcal{G}_1$ when transferred into $\mathcal{G}_2$
- mapping $(\mathcal{G}_1, \mathcal{G}_2) \mapsto \mathfrak{h}$ is not a one-to-one mapping (motor $\mathfrak{h}$ is invariant w. r. t. translations and rotations of $\mathcal{G}_1$ and $\mathcal{G}_2$ across $\mathcal{N}$)

2.3 Application to rigid bodies

Force motor, velocity motor, momentum motor und inertia dyad

$$\mathfrak{f} = \begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{d}_o \end{pmatrix} \qquad \mathfrak{v} = \begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{v}_o \end{pmatrix} \qquad \mathfrak{p} = \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{l}_o \end{pmatrix} \qquad \mathfrak{M} = \begin{pmatrix} m\boldsymbol{I} & -m\boldsymbol{R_s} \\ m\boldsymbol{R_s} & \boldsymbol{\Theta}_o \end{pmatrix}$$

Basic relations

momentum: $\mathfrak{p} = \mathfrak{M} \circ \mathfrak{v}$

kinetic energy: $T = \frac{1}{2}(\mathfrak{v}, \mathfrak{p}) = \frac{1}{2}(\mathfrak{v}, \mathfrak{M} \circ \mathfrak{v})$

power: $P = (\mathfrak{f}, \mathfrak{v})$

Equations of motion

inertial: $\dot{\mathfrak{p}} = \mathfrak{f}$

body-fixed: $\mathring{\mathfrak{p}} + \mathfrak{v} \times \mathfrak{p} = \mathfrak{f}$

Goals of the Motor Calculus

- description of
  - rigid body movement
  - forces and torques acting on a rigid body
  - momentum and angular momentum

  each by a six-dimensional "vector"

- description independent of reference frame and chosen reference point (geometrical interpretation)

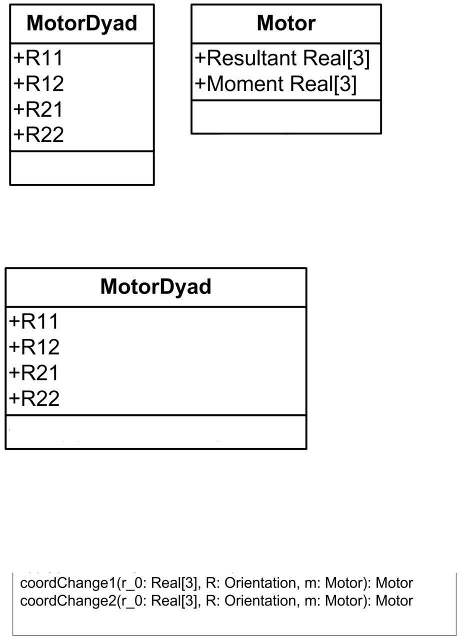
- very clear and simple structure of the fundamental mechanical laws

- formal equivalence to Newton's Second Law

Contents

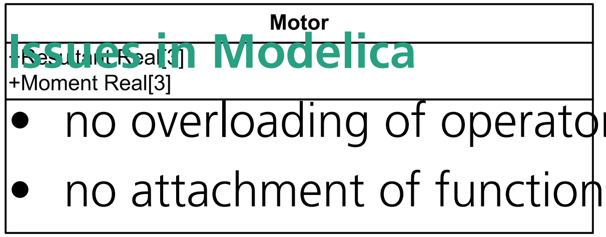
3. Aspects of implementation

3.1 Implementation of a motor library



Objective

- taking advantage of the efficient description in terms of motor calculus in Modelica
- object-oriented implementation of all operations in the class Motor
- specialisation by means of inheritance and polymorphy



Issues in Modelica

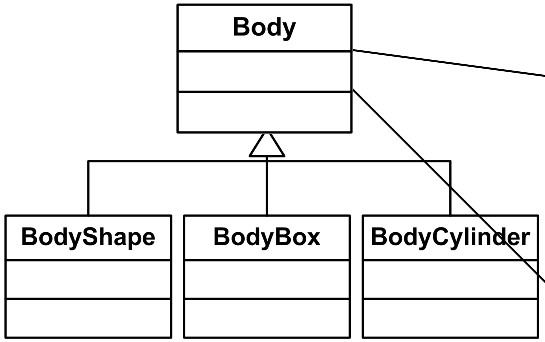
- no overloading of operators or functions
- no attachment of functions to classes

=> Compromise

3.2 Modification of the MultiBody Library

Modification of the class Body

- object-oriented implementation of the equations of motion using the motor calculus



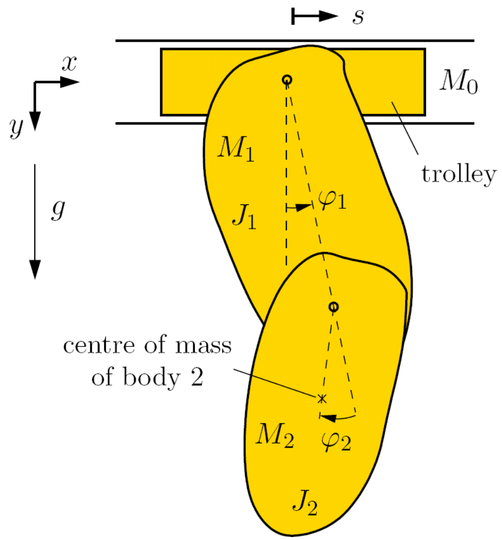
f_a = der(p) + 'x'(v,p) - f_g;

subclasses BodyBox, BodyShape, and BodyCylinder

Contents

4. Examples

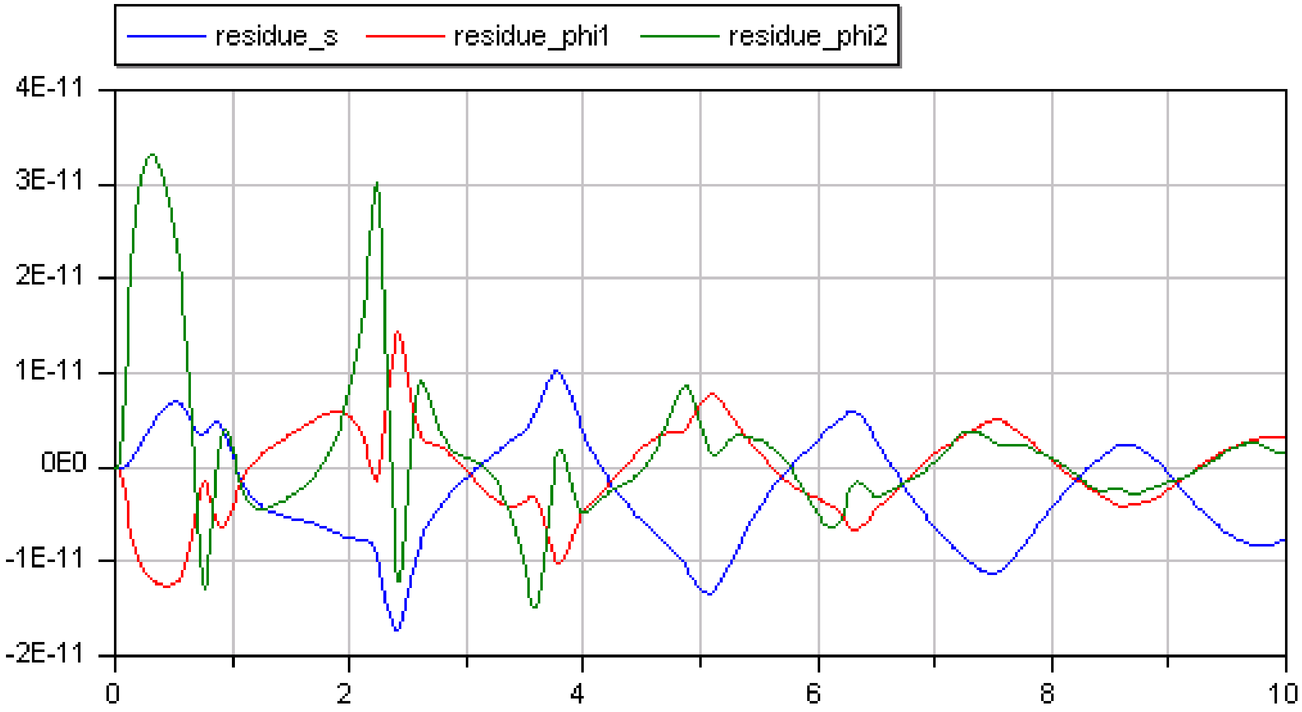
4.1 Damped moveable double pendulum



- Three rigid bodies moving in the Earth's gravitational field
- trolley: mass M_0
  viscose friction (ρ_0)
- 1st pendulum: mass M_1, moment of inertia J_1
  viscose friction (ρ_1)
- 2nd pendulum: mass M_2, moment of inertia J_2
  viscose friction (ρ_2)

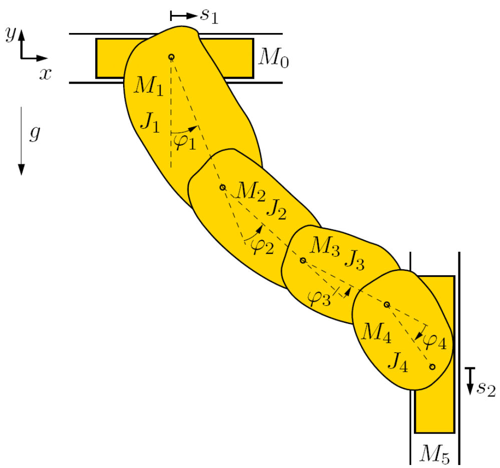
Animation of simulation results

Comparison of simulation results



➢ Errors between both simulation results are sufficiently small and decay for increasing values of the time t

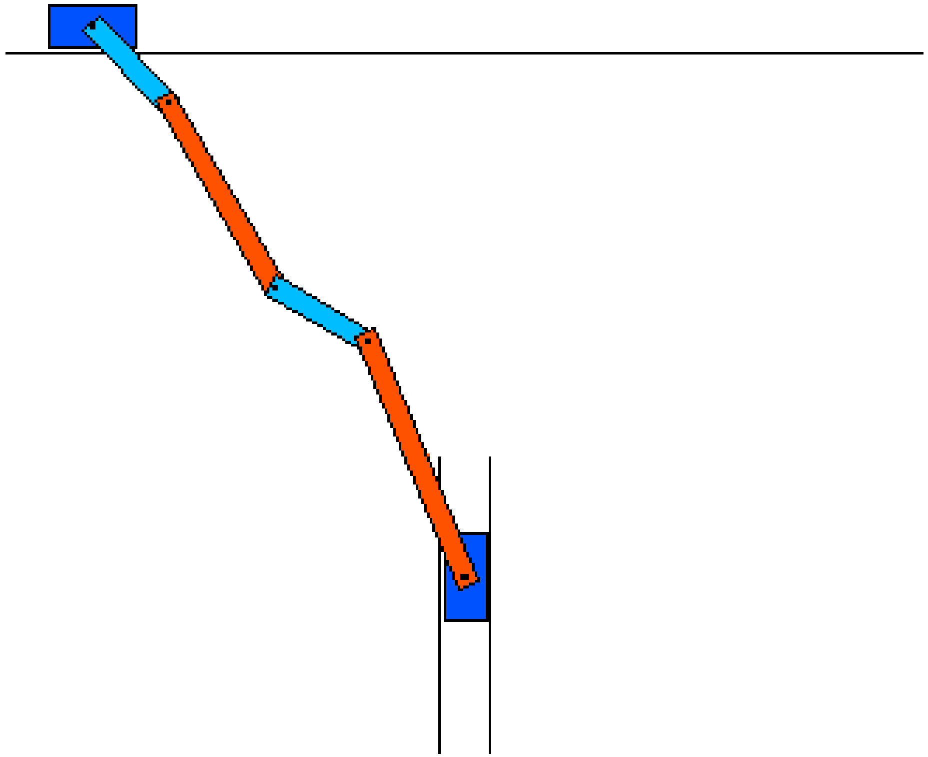
4.2 Damped fourfold pendulum on two movable sliders



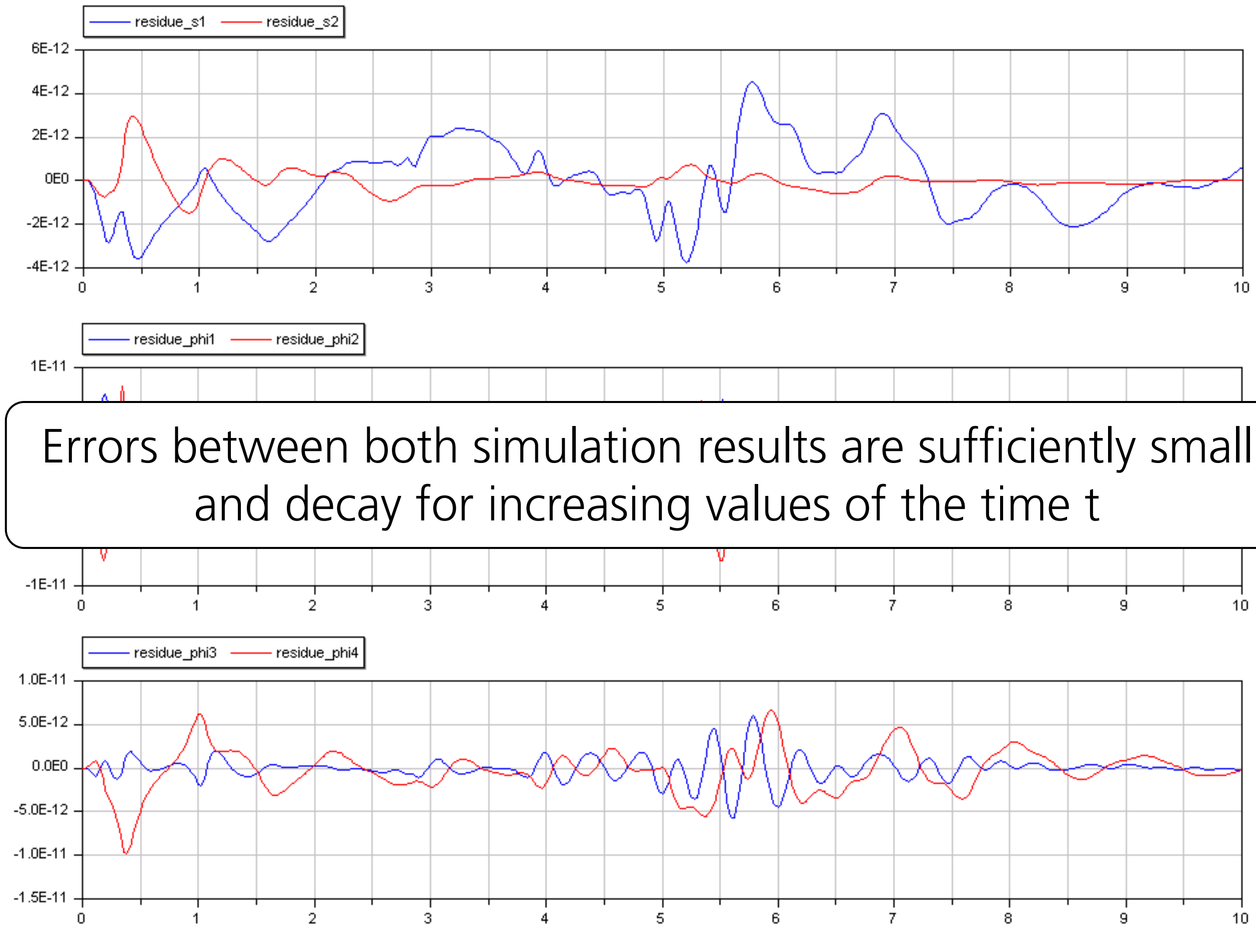
- Six rigid bodies moving in the Earth's gravitational field
- trolleys: masses M_0, M_5 viscose friction (ρ_0 and ρ_5)
- i^th pendulum: mass M_i, moment of inertia J_i viscose friction (ρ_i), $i = 1, \ldots, 4$

➢ closed planar kinematic loop

Animation of simulation results



Comparison of simulation results



Errors between both simulation results are sufficiently small and decay for increasing values of the time t

Contents

5. Summary/Outlook

presented:

- short introduction to von Mises' motor calculus
- implementation of Modelica library for motor calculus
- first simple implementation of the motor calculus within the MultiBody Standard Library
- simulation results for different non-trivial mechanical problems

future tasks:

- more sophisticated MultiBody implementation
- numerical analysis in terms of effectiveness and accuracy

Thank You!