Teaching Modelica for Mathematicians and Engineers

Modelica Educational Workshop
Berlin

Bernhard Bachmann
University of Applied Sciences
Bielefeld
Outline

- Past Teaching Experience
  - Mathematicians
    - precognition, course objectives
  - Course Details
    - theoretical content, tools, exercises

- Discussion on Future Teaching Options
  - Mathematicians and Engineers
    - master students having less mathematical background
  - Tools and concepts
    - adaptation of previous course
Past Teaching (1999 – 2008)

- **Course attendees (diploma-study in mathematics) [ca. 15]**
  - good mathematical background
    - analysis and linear algebra topics
    - optimization (linear, nonlinear problems)
    - numerical methods (no ODEs)
    - theory on ordinary differential equations
    - familiar tools: Maple, Matlab (no Simulink)
    - basic programming knowledge (C/C++)
  - basic engineering background
    - simple mathematical modeling of physical components
      mechanics, electrical systems (static)

- **Course objectives**
  - engineering aspects
    - component and library development in Modelica
  - mathematical aspects
    - understand symbolic transformations and numerical issues
Past Teaching (1999 – 2008)

- Course details
  - 4 semester periods per week (13 weeks)
    - theory and practical exercises
  - learning by doing (small projects)
  - tools
    - Matlab-Simulink (1st project)
    - Dymola (Modelica projects)
  - exam
    - practical
      - projects (development and explanation)
    - theory
      - mathematical and modeling aspects
Past Teaching (1999 – 2008)

- 1st course: Basic understanding of the principles using a simple electrical system
  - modeling
    - develop the DAE representation
      - abstract mathematical view
    - understand numerical integration (Euler method)
    - sort the equation system (find causality)

  - simulation and implementation
    - using Matlab
    - using Simulink
    - using Dymola (flat representation)

\[
\begin{align*}
  u &= A \cdot \sin(2 \pi f t + \phi) \\
  u &= u_1 + u_2, \ u = u_3 + u_4 \\
  i &= i_1 + i_2 \\
  u_1 &= R_i_1, \ u_3 = R_2 i_2, \\
  C \frac{du_2}{dt} &= i_1, \ L \frac{di_2}{dt} = u_4
\end{align*}
\]
Past Teaching (1999 – 2008)

- 1st course project
  - basic understanding of the principles using a simple mechanical system
  - modeling
    - develop the DAE representation
    - understand numerical integration (Euler method)
    - sort the equation system (find causality)
  - simulation and implementation
    - using Matlab
    - using Simulink
    - using Dymola

\[
\begin{align*}
\tau_1 &= u, \quad \omega_1 = \dot{\varphi}_1 \\
J_1 \cdot \dot{\varphi}_1 &= \tau_1 + \tau_2 \\
\tau_2 &= c \cdot (\varphi_2 - \varphi_1) \\
0 &= \tau_2 + \tau_3, \quad \omega_2 = \dot{\varphi}_2 \\
J_2 \cdot \dot{\varphi}_2 &= \tau_3
\end{align*}
\]
Past Teaching (1999 – 2008)

- 2nd course:
  Benefit of using Modelica
  Getting started with Dymola
  - examples from 1st course continued
  - modeling
    - drag and drop
    - library structure
    - find components
  - simulation
    - compile model
    - experiment setup
    - view and compare results

\[ u(t) \]

\[ \phi_1, \tau_1 \]

\[ J_1 \]

\[ \phi_1, \tau_2 \]

\[ c \]

\[ \phi_2, \tau_3 \]

\[ J_2 \]

\[ \phi_2, \tau_4 = 0 \]
Past Teaching (1999 – 2008)

- **2nd course project**
  - build up and simulate different physical systems
    - examples from 1st course
    - drive train
    - triple pendulum
    - ...

![Diagram of a Modelica model involving components like resistor, inductor, and gearing.](image)

height = 200
Past Teaching (1999 – 2008)

- 3rd course: Getting started with Modelica
  - flat Modelica
    - basic keywords
      - model, parameter, equation, basic types, der, …
    - type attributes
      - min, max, units, …
    - type classes
      - library SIunits
  - hierarchical Modelica
    - model and connector classes
    - connect statement
      - basic principles of flow and potential variables
Past Teaching (1999 – 2008)

- 3rd course project
  - implement and simulate a Pendulum model
    - flat representation
      using predefined types (SI units)
    - hierarchical representation (using Multibody library)
      animation
    - build up a simple controller to adjust the angle

\[
J \cdot \ddot{\phi} = -m \cdot g \cdot L \cdot \cos(\phi) - d \cdot \dot{\phi} + \tau
\]
Past Teaching (1999 – 2008)

- **4th course:**
  Introduction to basic control techniques
  - examples
    - with/without feedback
  - Laplace-transformation
  - different mathematical formulations
    - functional description
    - block diagram
    - Step function response
  - standard controller
    - P-, D-, I-, PD-, PI-, PID-controller

- **4th course project**
  - example from 3rd course continued
  - try different controller and compare results
Past Teaching (1999 – 2008)

- **5th course:**
  - Build Libraries in Modelica
    - package concept
      - example using simple electrical component
    - modifier concept
    - build libraries in Dymola
      - icon layer, diagram layer
      - coordinate system
      - connector view
      - parameter settings
- **5th course project**
  - build up a library
    - motor (including control scheme)
    - gear box (including friction elements)
  - adjust control parameter for suitable test cases
Past Teaching (1999 – 2008)

- **6th course:**
  Build Libraries in Modelica
  - general connection concept
    - energy flow, domain specific potential and flow variables
    - discuss practical issues (rotational)
  - multidisciplinary modeling
  - parameter propagation
    - modifications
    - GUI in Dymola

- **6th course project**
  - continue 5th course project
Past Teaching (1999 – 2008)

- 7th course:
  Advanced Modelica
    - class types
      - type, model, block, function, package, connector
    - algorithm versus equations
    - additional keywords
      - input, output, protected
    - matrices
      - definition, element access, operations, inline functions
    - example of general transfer function

\[
\dot{x} = \begin{bmatrix}
-\frac{a_2}{a_1} & -\frac{a_3}{a_1} & -\frac{a_4}{a_1} & -\frac{a_5}{a_1} \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{bmatrix} \cdot x + \begin{bmatrix}
\frac{1}{a_1} \\
0 \\
0 \\
0 \\
\end{bmatrix} \cdot u
\]

\[
y = \begin{bmatrix}
b_2 - a_2 \frac{b_1}{a_1} & b_3 - a_3 \frac{b_1}{a_1} & b_4 - a_4 \frac{b_1}{a_1} & b_5 - a_5 \frac{b_1}{a_1}
\end{bmatrix} \cdot x + \frac{b_1}{a_1} \cdot u
\]

- 7th course project
  - still continue 5th course project
Past Teaching (1999 – 2008)

- **8th course:**
  Symbolic transformation algorithm
  - mathematical DAE representation
  - regular (index 1) problems
  - matching algorithm
  - sorting (Tarjan algorithm)
  - BLT representation of adjacence matrix

- **8th course project**
  - implement the BLT algorithm for random matrices
Past Teaching (1999 – 2008)

- **9th course:**
  Higher Index problems
  - examples (mechanical, electrical)
    - mathematical DAE representation
    - definition of the structural and differential index
  - detect singular set of equations
    - Pantelides algorithm
  - dummy derivative method
  - state selection mechanism
  - initialization of models

- **9th course project**
  - continue 8th course project
Past Teaching (1999 – 2008)

- **10th course:**
  - Advanced Modelica
    - arrays of component
      - for-loop, variable number of connect statements
    - example
      - transmission line model
    - introduce basic heat flow library
- **10th course project**
  - simulate the temperature distribution of an isolated bar
Past Teaching (1999 – 2008)

- **11th course:**
  - Model discontinuities
    - Modelica Standard library
      - digital controller
      - electrical switch or diode (not ideal)
      - clutch and brake model
    - Modelica language elements
      - if-then-else, when, noEvent, smooth, reinit, pre, …
    - symbolic transformation
      - synchronous equation

- **11th course project**
  - implement examples
    - hysteresis function
    - pulse width modulation block
Past Teaching (1999 – 2008)

- 12th course: Model Discontinuities
  - numerical issues
    - stiffness (“not ideal” switch)
    - time versus state events
    - rounding errors
    - event iterations

- 12th course project
  - implement further examples
    - bouncing ball
    - …
Past Teaching (1999 – 2008)

- 12th course: Model Discontinuities
  - numerical issues
    - stiffness (“not ideal” switch)
    - time versus state events
    - rounding errors
    - event iterations

- 12th course project
  - implement further examples
    - bouncing ball
    - …
Past Teaching (1999 – 2008)

- **13th course:**
  - **Model Discontinuities**
    - varying higher index problems
    - examples
      - mechanical, electrical
    - symbolic transformation
      - analyse singularity
      - dummy derivative method

- **13th course project**
  - run and analyse examples
    - introduce dummy derivative terms in branches
Past Teaching (1999 – 2008)

- **13th course:**
  - Model Discontinuities
    - varying higher index problems
    - examples
      - mechanical, electrical
    - symbolic transformation
      - analyse singularity
      - dummy derivative method

- **13th course project**
  - run and analyse examples
    - introduce dummy derivative terms in branches
Discussion on Future Teaching Options

- **Course attendees**
  - master-study in „Optimization and Simulation“
  - mathematicians and engineers
    - mechanical, electrical, mechatronics,…
  - heterogeneous background in mathematics and engineering

- **Course objectives**
  - engineering aspects
    - component and library development in Modelica
  - mathematical aspects
    - understand symbolic transformations and numerical issues

- **Tools**
  - licencing issues
    - **OpenModelica (SimForge)**, Dymola, MapleSim, MathModelica

- **Applications / Projects**