Teaching Modelica® for electrical and mechanical engineers at Technical University of Munich

Martin Otter

Modelica Educational Workshop, 2. April 2009, TU Berlin.

Lecture: Object-Oriented Modeling of Mechatronic Systems

In winter semester at Technical University of Munich,
for electrical and mechanical engineering, since 1997, at
Lehrstuhl für Elektrische Antriebssysteme und Leistungselektronik
(1997-2009: Prof. Schröder; since 2009: Prof. Kennel).

Goal:
Learning the basics of object-oriented modeling at hand of Modelica
with exercises from the electrical, mechanical, control and thermal domain.
Practical exercises performed with Dymola.

Number of students:

2008/2009	102 students
2007/2008	68 students
2006/2007	73 students
2005/2006	85 students
2004/2005	61 students

Organization

- 90 min. lecture once a week
- Exercise once a week. Organized as:
  - 3 times a week, a 90 min. supervised exercise in computer room (20 computers, at most 2 students per computer).
  - I give one of the supervised exercises per week, the other two are given by tutors.
- Exercises are posted on the Web one week beforehand, together with solution (in order that students can check themselves whether it is worth to participate at the supervised exercises).
- Students can install Dymola on their computers and can therefore perform exercises at home
- The students grade is determined in a 60 min. test in the last lecture (some questions are with respect to the exercises).

Overview of Lectures

Continuous systems

object diagrams
differential-algebraic equations (DAE)
transformation to state space form
consistent initial conditions of a DAE
singular DAEs
integration methods

Theory

Discontinuous systems

state and time events
synchronisation of events
hierarchical state diagrams
many ideal switching elements
Reel/Boolean equation systems

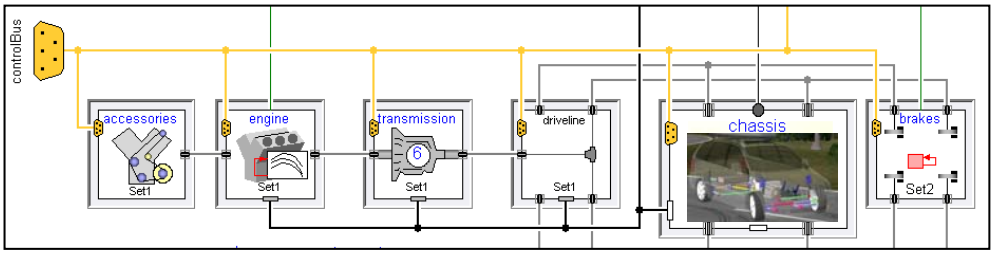
Applications

Modelica (language) and Dymola (program)
electrical systems/motors, drive trains,
3-dim. mechanics, heat transfer,
input/output blocks,
inverse models for control systems,
switching elements (diode, thyristor, friction, ...),
real-time applications

13 Lectures (in 2008/2009)

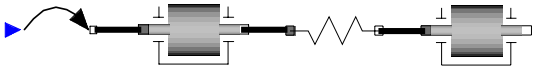
Lecture 1

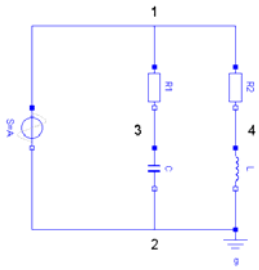
Overview, Object diagrams, Modelica Standard Library, introduction into Dymola



Lecture 2

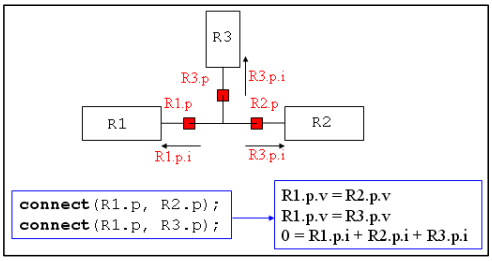
Basics of equation based modeling, potential/flow variables, simple examples.
How many equations ("balanced models")?





Lecture 3

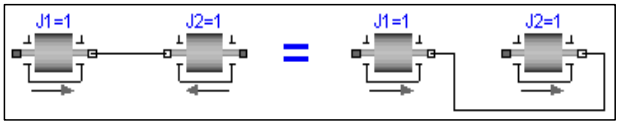
Introduction into the Modelica language (textual/graphical Modelica models), connector design (automatic generation of boundary and of balance equations), examples of simple components (capacitor + inertia)

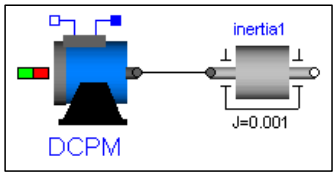


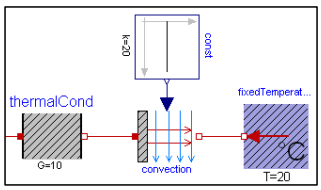

```
model Capacitor
   import SI = Modelica.SIunits;
   parameter SI.Capacitance C;
   SI.Voltage u "Spannungsabfall";
   Pin p, n;
equation
          0 = p.i + n.i;
          u = p.v - n.v;
   C*der(u) = p.i;
end Capacitor;
```

Lecture 4

Connectors and components for drive trains and heat transfer

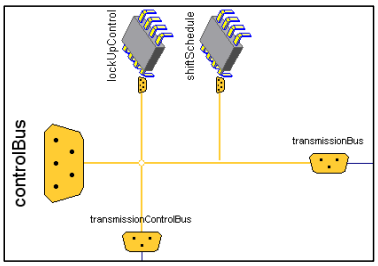


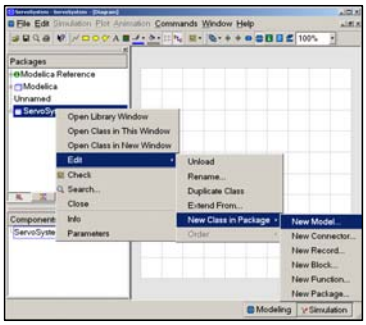




Lecture 5

Connectors for signals and buses, hierarchical connectors, inheritance, packages (rename/copy/unload/new...).

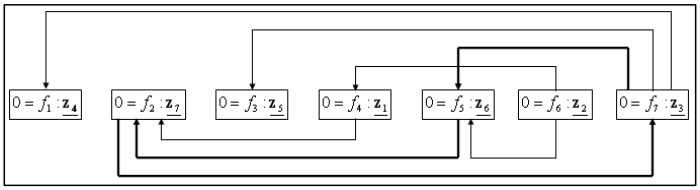


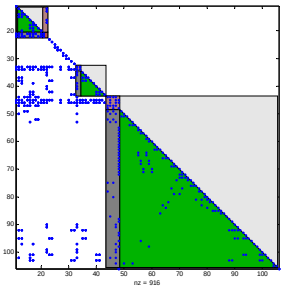


Lecture 6

Sorting, i.e. BLT transformation (example, assignment/Tarjan algorithm), variable substitution, i.e., tearing.

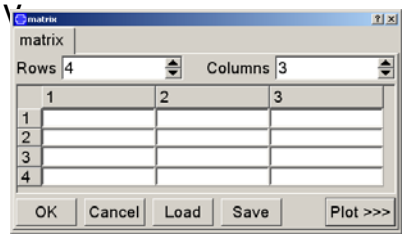
$$0 = f_2(\underline{\mathbf{z}_2})$$
$$0 = f_4(\underline{\mathbf{z}_1}, z_2)$$
$$0 = f_3(z_2, \underline{\mathbf{z}_3}, \underline{\mathbf{z}_5})$$
$$0 = f_5(z_1, \underline{\mathbf{z}_3}, \underline{\mathbf{z}_5})$$
$$0 = f_1(z_3, \underline{\mathbf{z}_4})$$

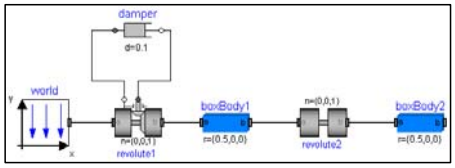




Lecture 7

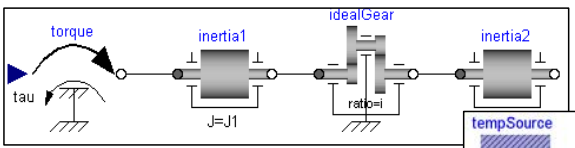
Matrices and arrays in Modelica, interfaces and components for multi-body systems, initialization in Modelica



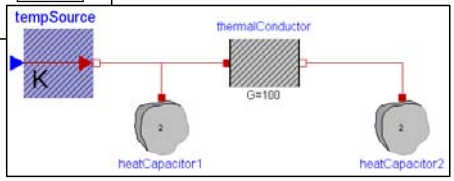


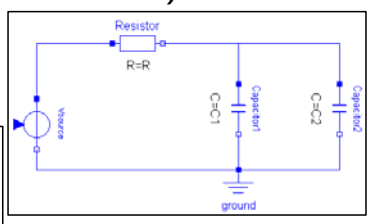
Lecture 8

Singular systems (examples, Pantelides, dummy derivative method)



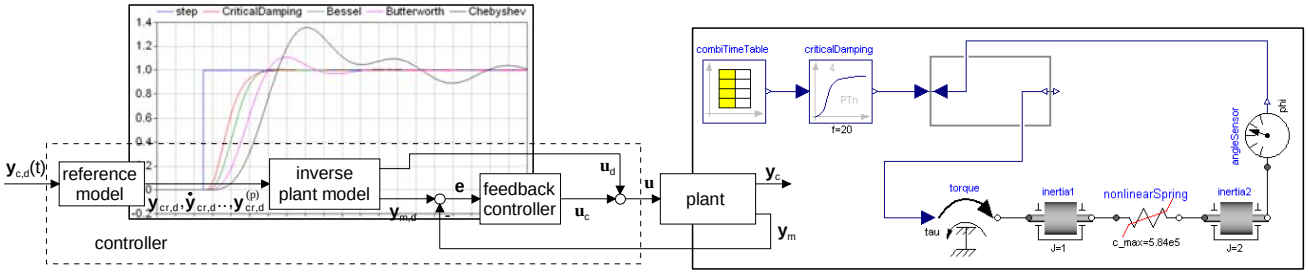
$$\det\left(\frac{\partial \mathbf{f}}{\partial \mathbf{z}}\right) = 0$$





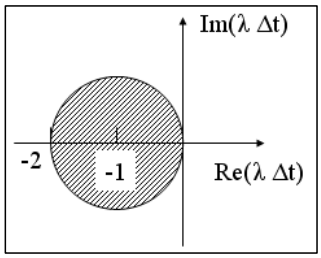
Lecture 9

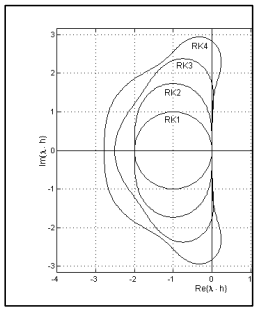
Controller with linear, and non-linear inverse model + filter.



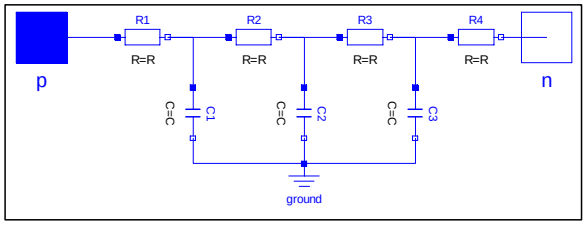
Lecture 10

Integration methods (explicit/implicit Euler, method order, stability region, examples, Dymolas integrators), component arrays.



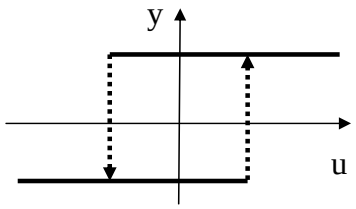


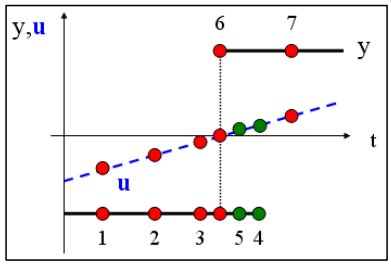
$$\left|\tilde{\mathbf{x}}(t+\Delta t)-\mathbf{x}(t+\Delta t)\right| \le O(\Delta t^{n+1})$$

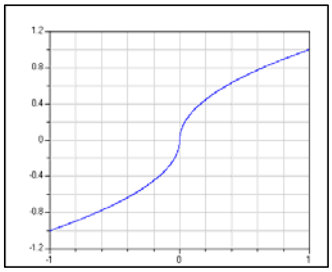


Lecture 11

Discontinuous systems, state/time events, relation triggered events, pre(..), when, smooth(), noEvent(), synchronous equations (sorting).

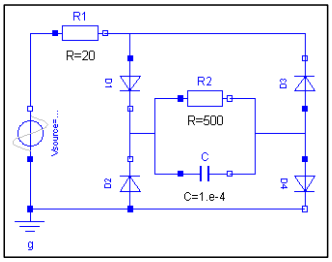


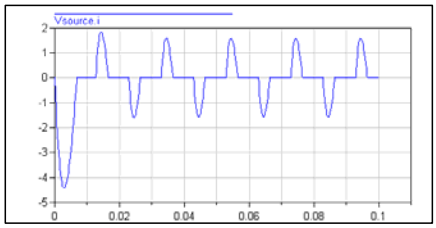


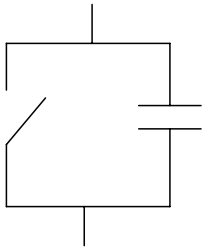


Lecture 12

Variable structure systems (electrical switch, diode, thyristor, parameterized curve description), Real/Boolean equation systems, problems (singular systems)

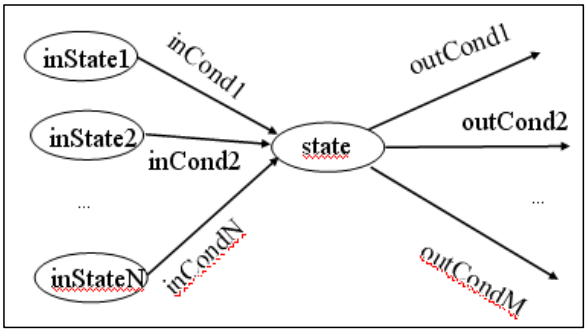


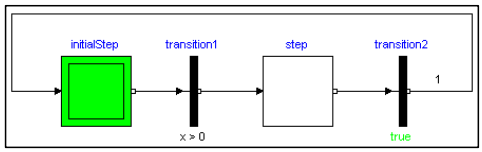


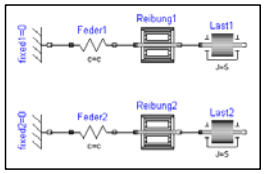


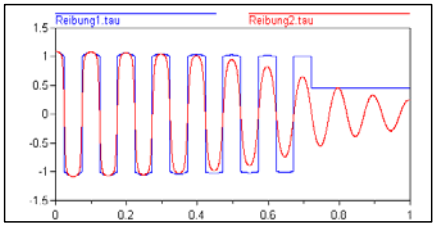
Lecture 13

State diagrams (native implementation, StateGraph), friction





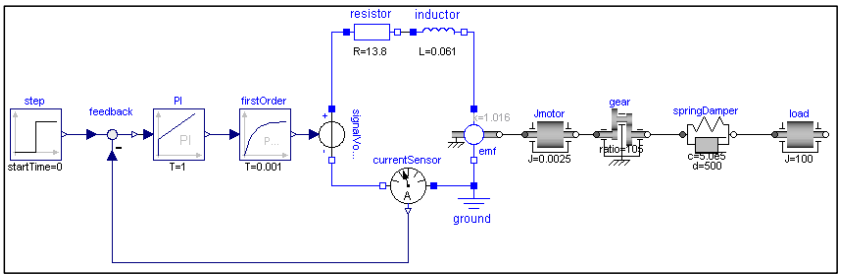




9 Exercises (in 2008/2009)

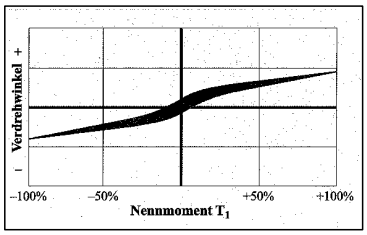
Exercise 1

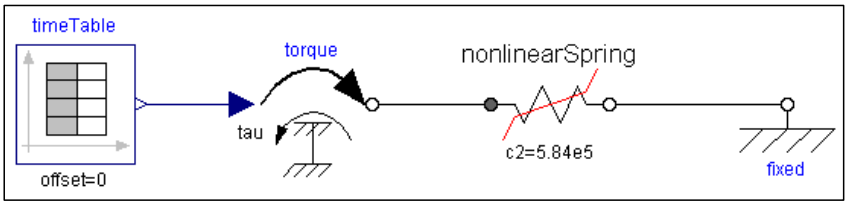
Build electrical motor with current controller + elastic load and tune PI coefficients of current controller manually.



Exercise 2

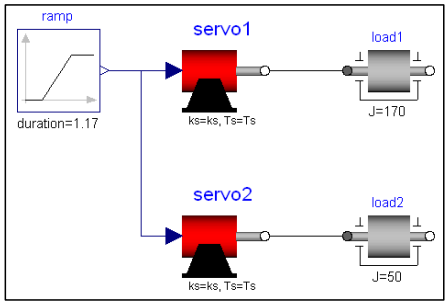
Implement and test non-linear spring characteristic

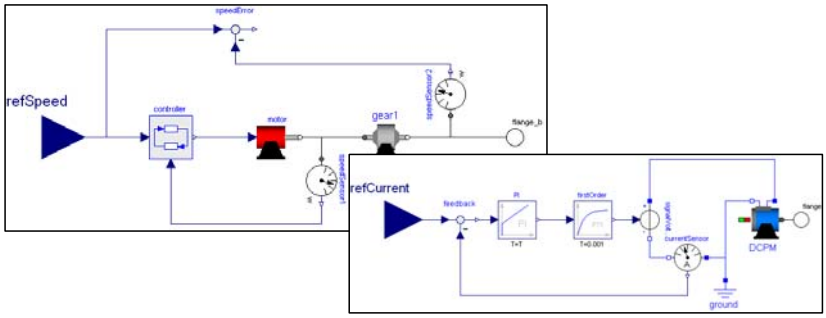




Exercise 3

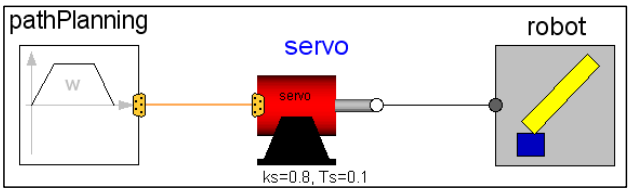
Build and tune servo system for two operating points

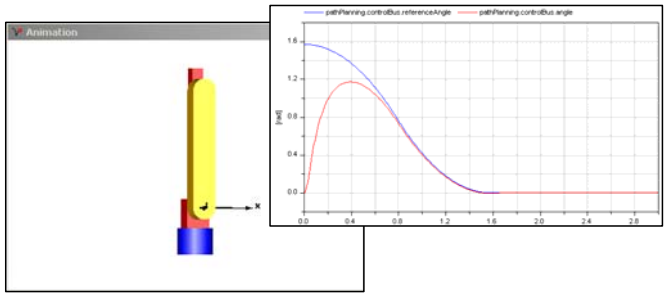




Exercise 4

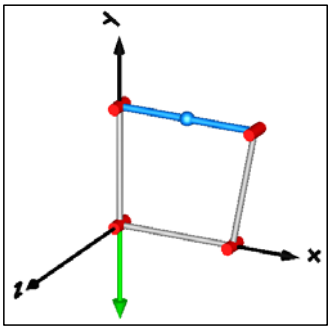
Controlled one-arm robot with bus, controller, path planning, steady-state initialization

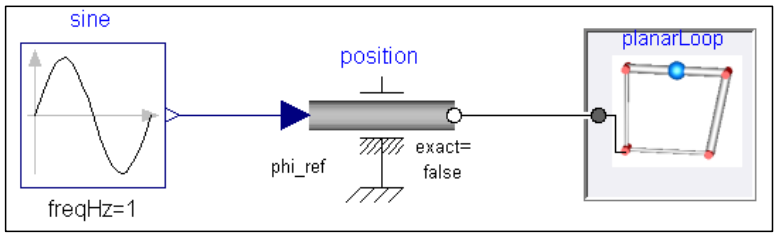




Exercise 5

Four-bar mechanism, different implementations of planar loops, replaceable models



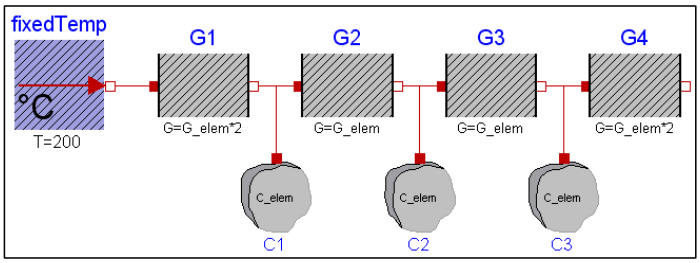


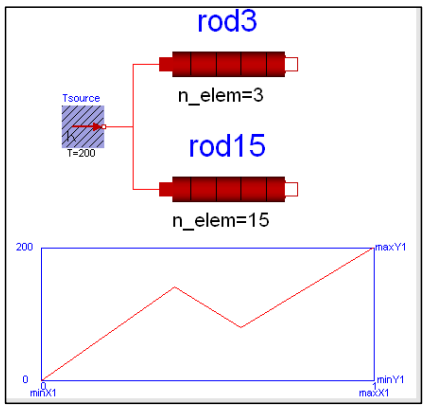
Exercise 6

Nonlinear robot control with inverse model

Exercise 7

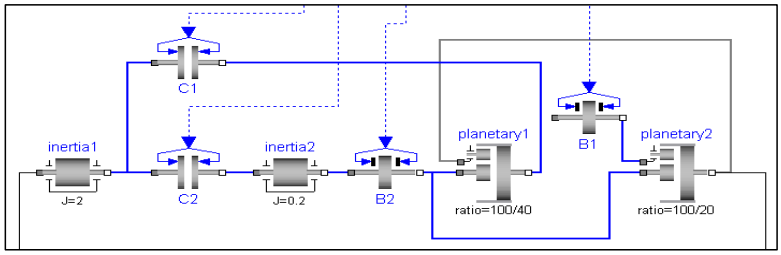
Heat transfer in rod
(implementation with vector of components)





Exercise 8

Build automatic gearbox and design simple ECU (electronic control unit)



gear	C1	C2	B1	B2
0				
1	on		on	
2	on			on
3	on	on		

Exercise 9

Model electrical motor with rectifier, supply source and load