Modelica® Tutorial for Beginners

Exercises with Dymola

Bernhard Bachmann
University of Applied Sciences
Bielefeld

based on material from Hilding Elmqvist and Martin Otter

Outline

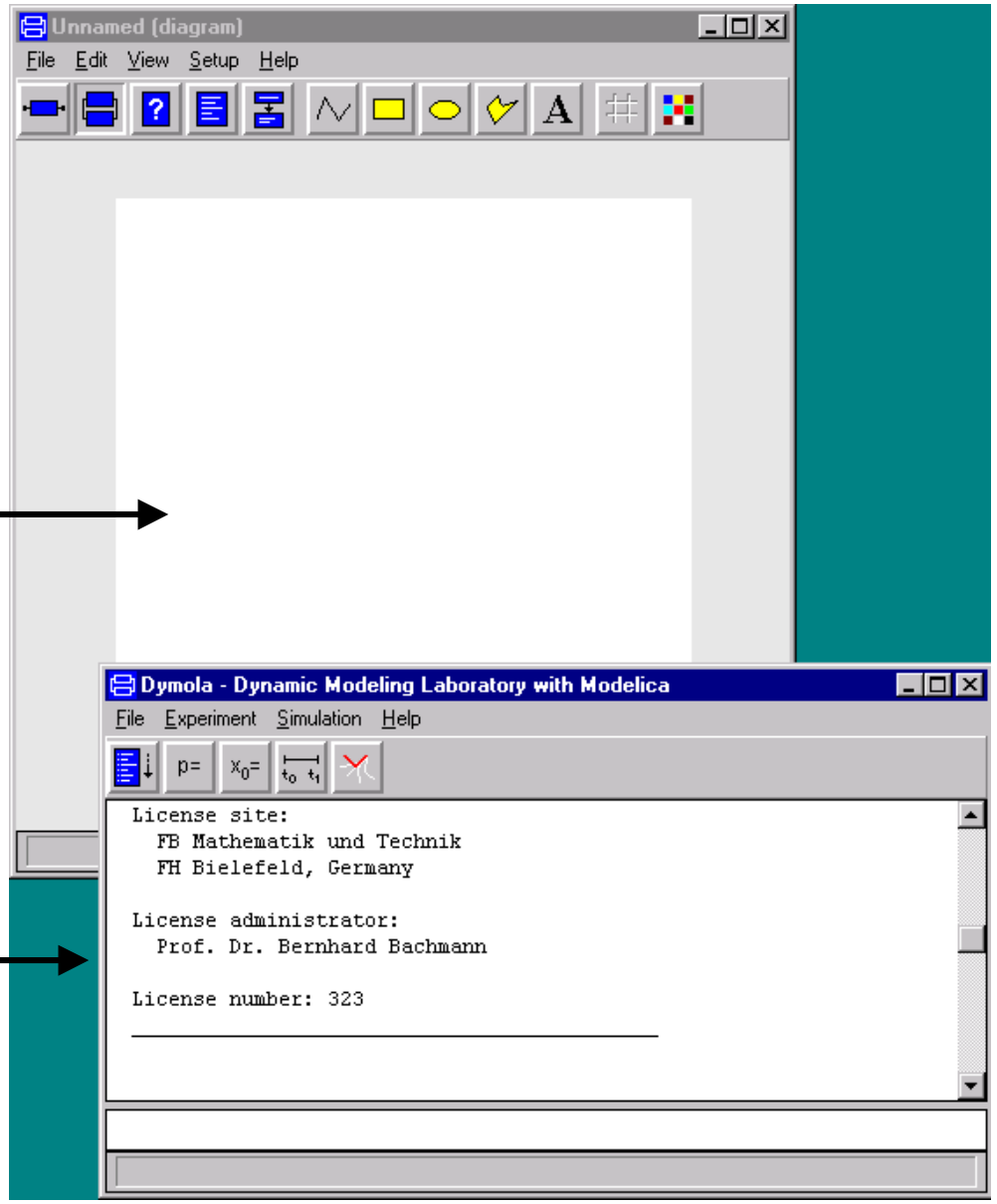
- Getting started with Dymola
- Exercises
  - Using the Modelica standard library
    - 1. Simple drive train
    - 2. Animation of a pendulum
  - Writing simple models (own equations)
    - 3. Acceleration of mass
    - 4. Different model variants of a pendulum
  - Definition of arrays of components
    - 5. 1-dim heat flow
  - Hybrid systems
    - 6. Automatic gear box
    - 7. Bouncing ball
    - 8. Hysteresis Block

Getting Started with Dymola

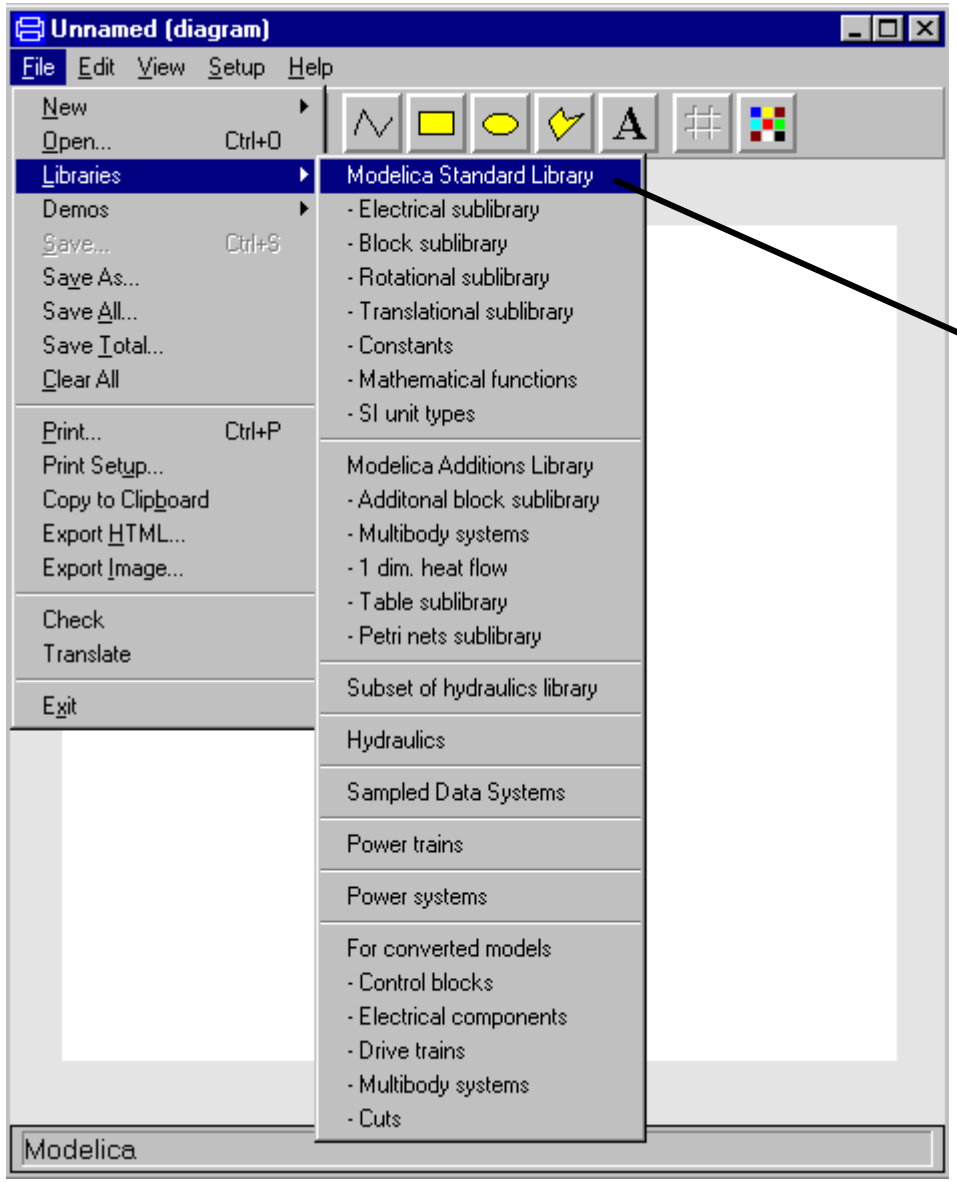
1. Start Dymola
(2 windows appear)

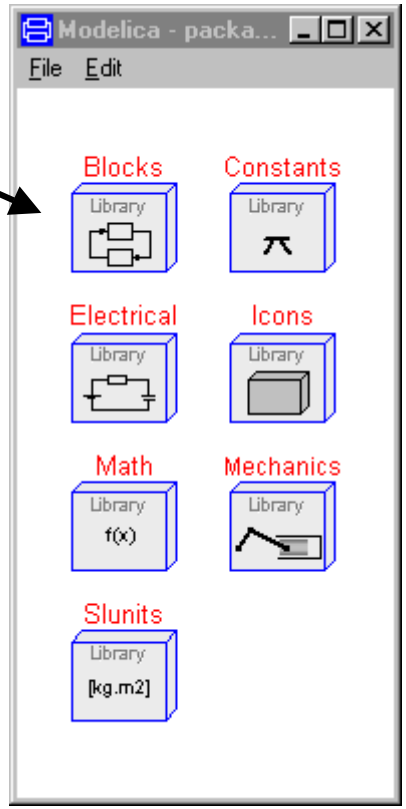
Define model as
composition diagram

simulate here



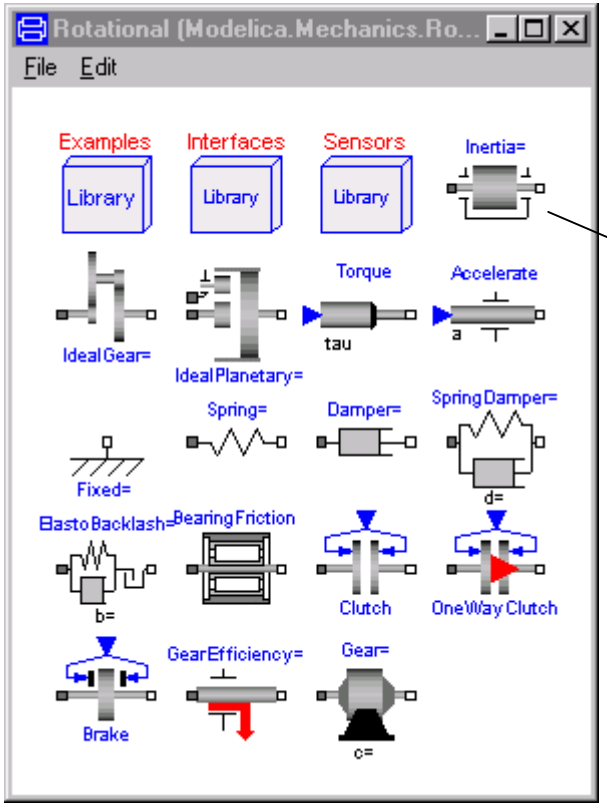
Open library windows

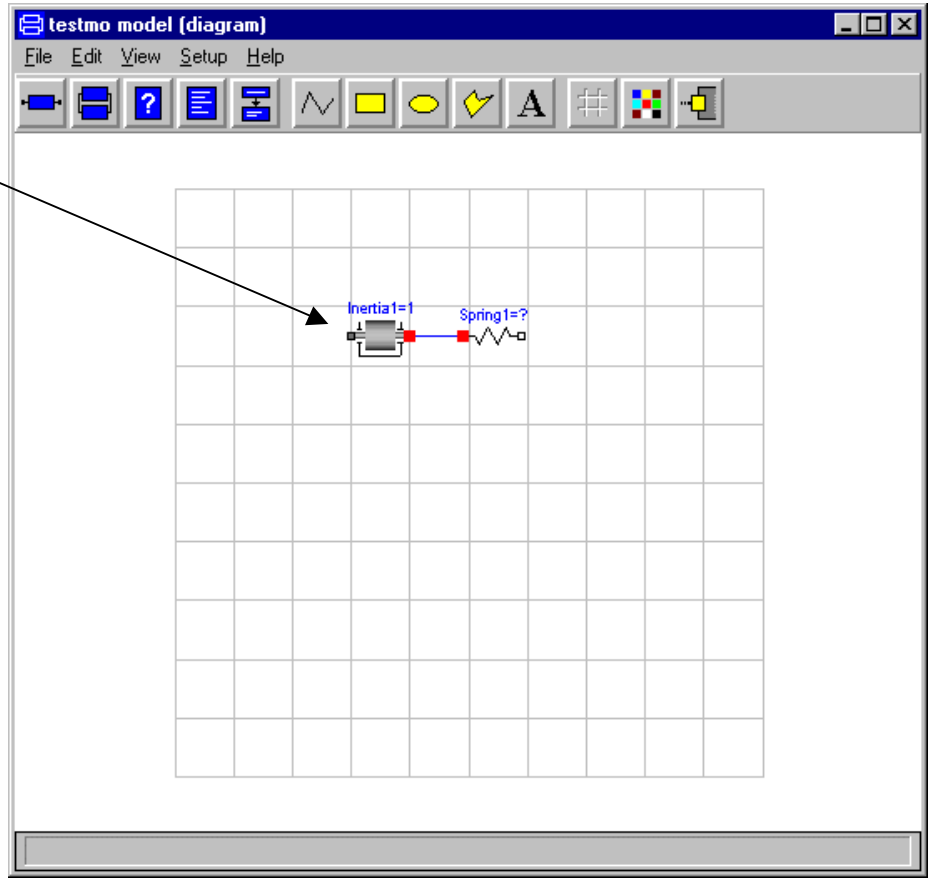




Library window

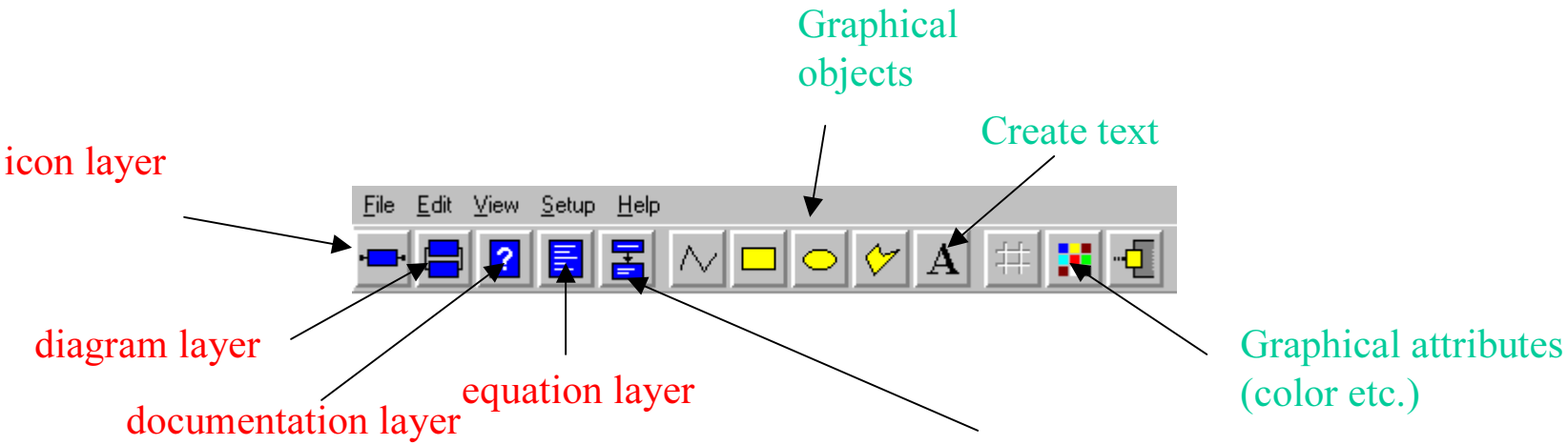
Drag & Drop modeling

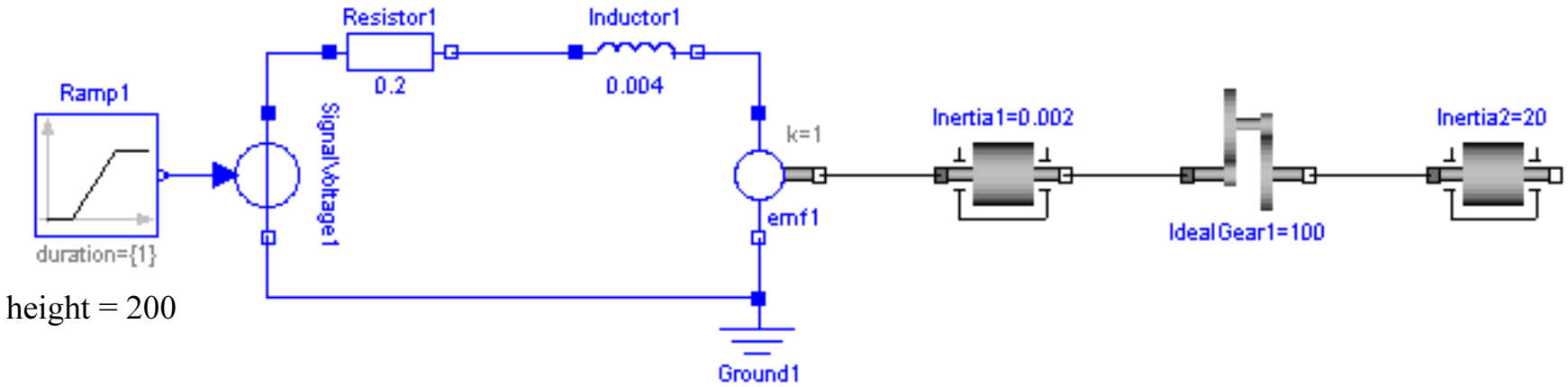




Modelica-Bibliothek:
Modelica.Mechanics.Rotational

Build up model using libraries



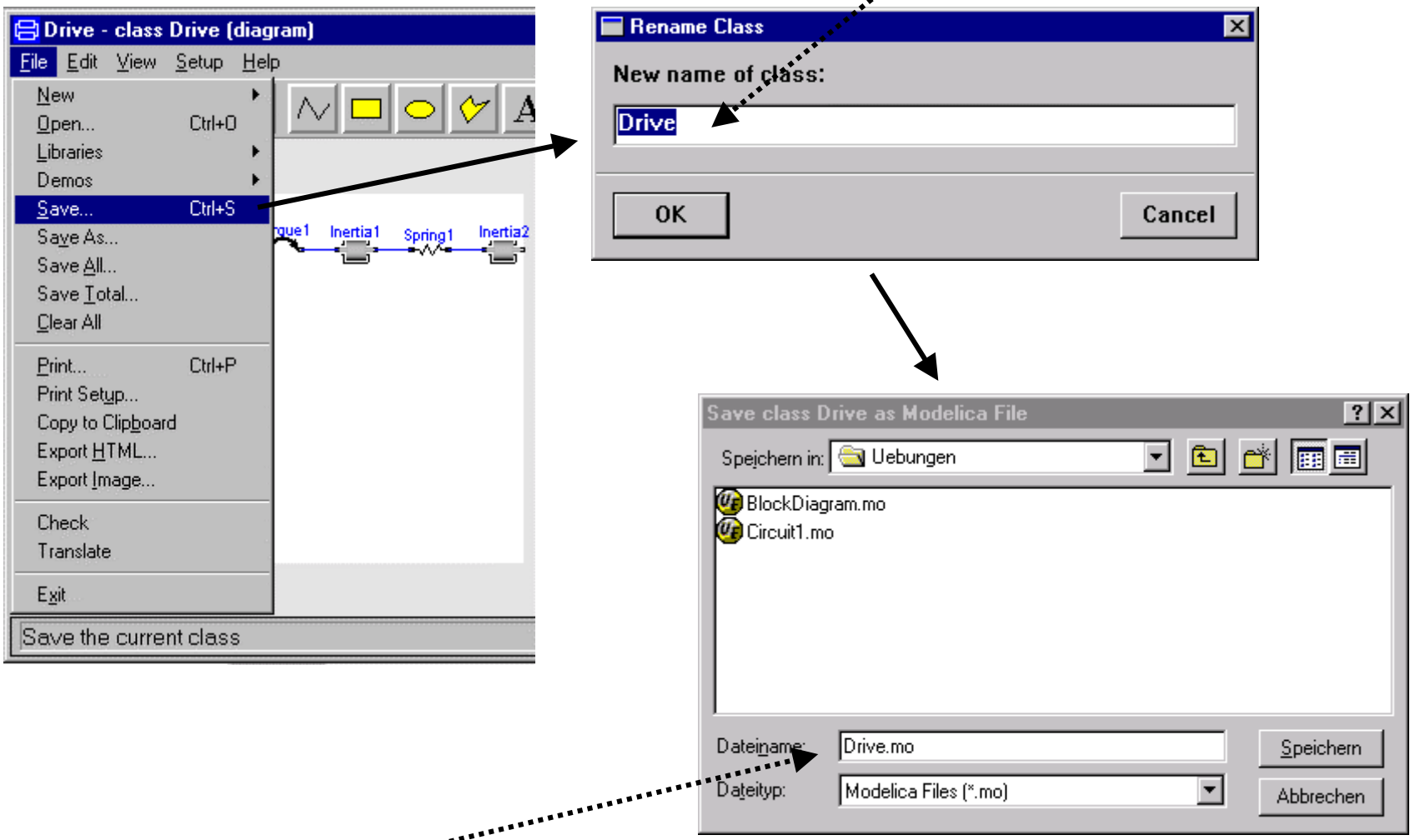


height = 200

diagram layer

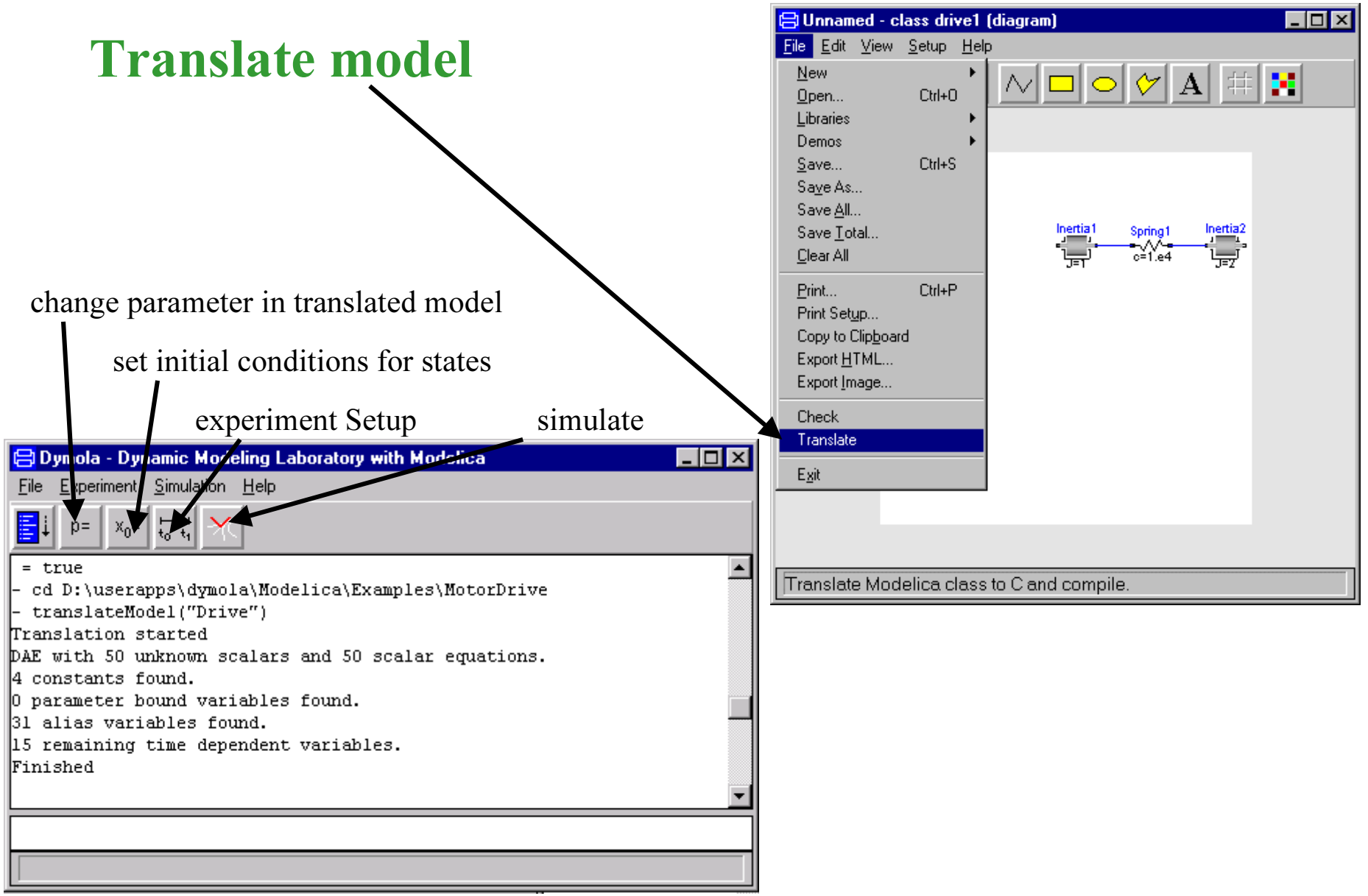
Save model

Model name (letters, digits, ...)

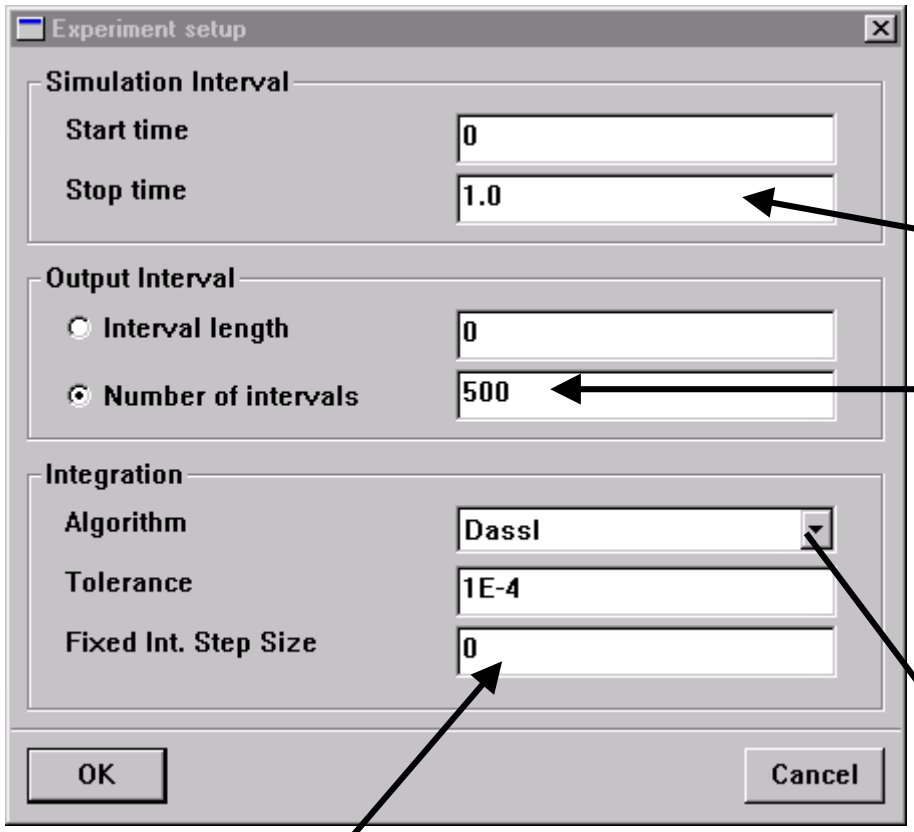


File name (Default: **<Model name>.mo**)

Translate model

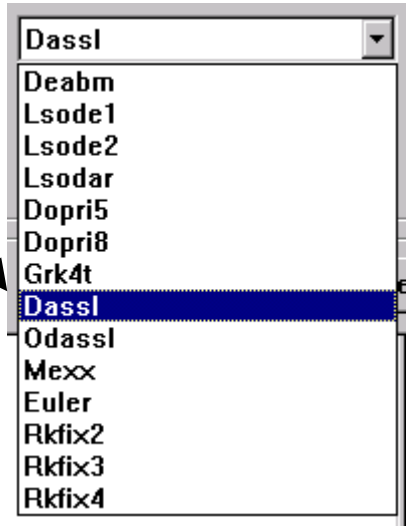


Set integration parameters in experiment setup



Final simulation time

Number of output points

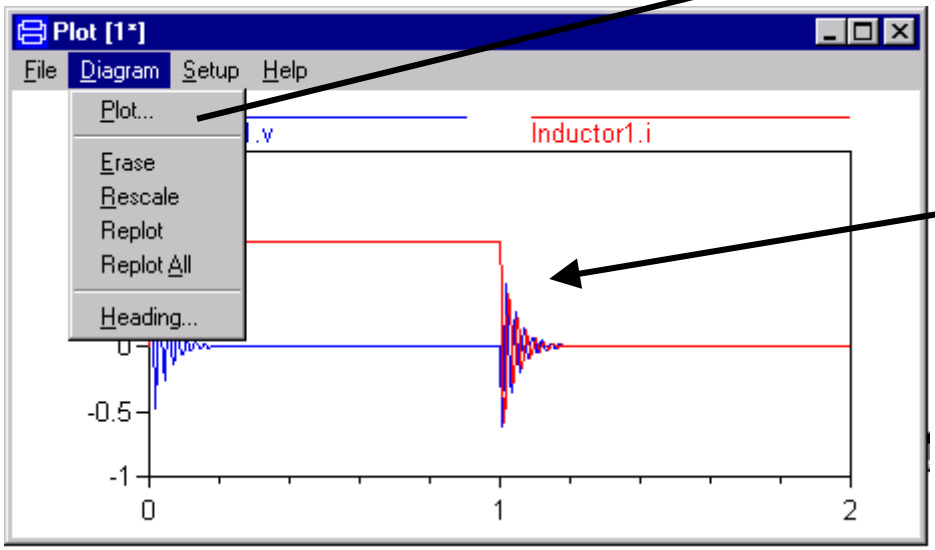


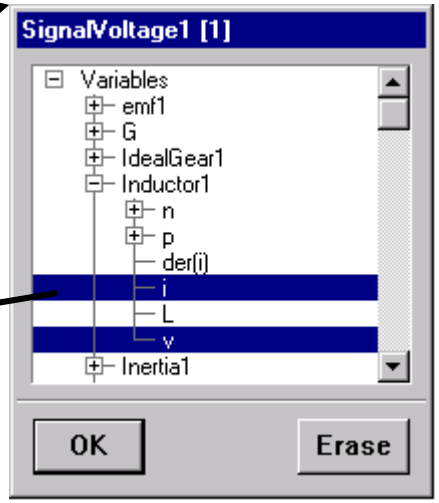
integration routine

Only for integration routines with fixed step size (Euler/Rkfix2/3/4)

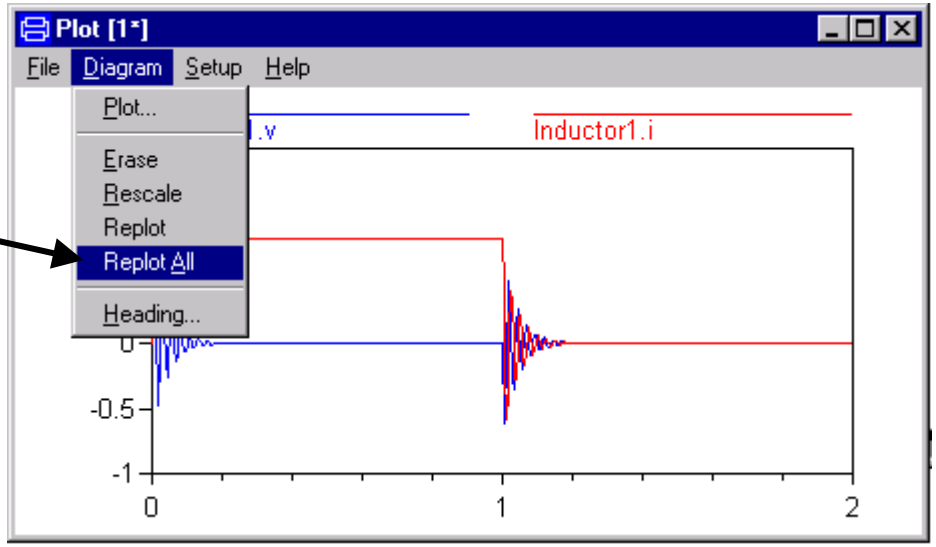
Simulieren!

Plot results

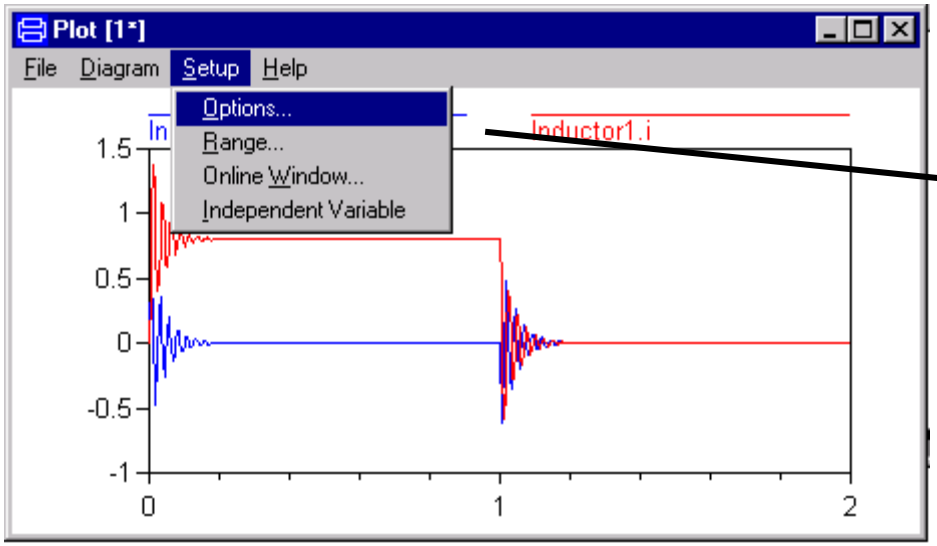


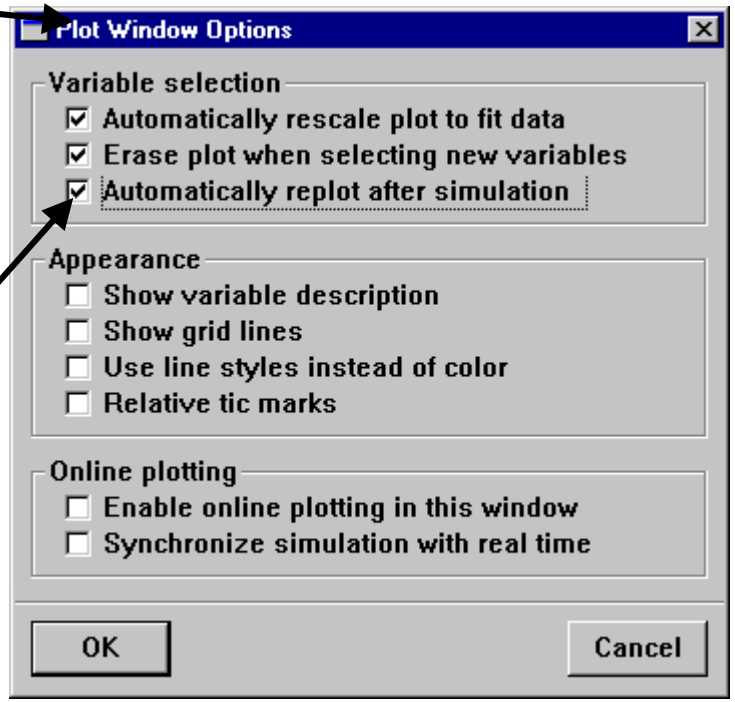


After a second simulation the previous variable selection can be reused



Set Options

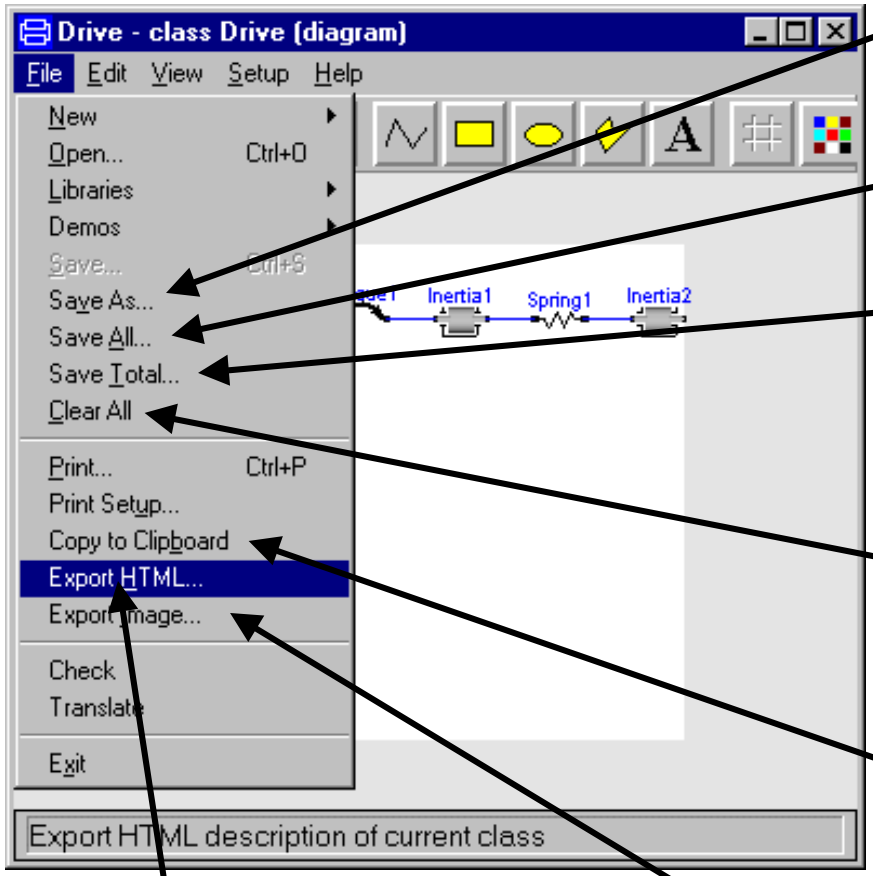




Even better:

use “Re-plot all” as default

Some useful menus of Dymola



Save As ...
Save model under a new name

Save All ...
Save all open models

Save Total ...
Save model including all used components from all libraries in one file

Clear All
clear all models in Dymola (re-start)

Copy to Clipboard
store picture of model in windows metafile format on clipboard. (for re-use in PowerPoint or Word)

Export HTML ...
Generate complete documentation of the model in HTML-format

Export Image ...
Store picture of model in gif format

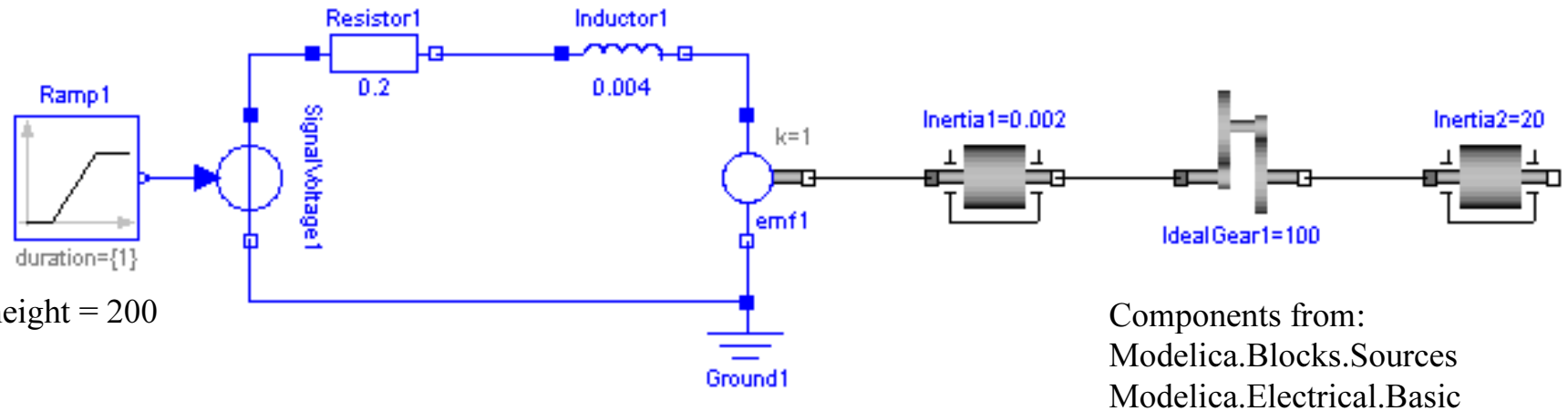
Exercise 1: Simple drive train

Goal:

Build model and simulate using only components from available libraries (useful for complete beginners)

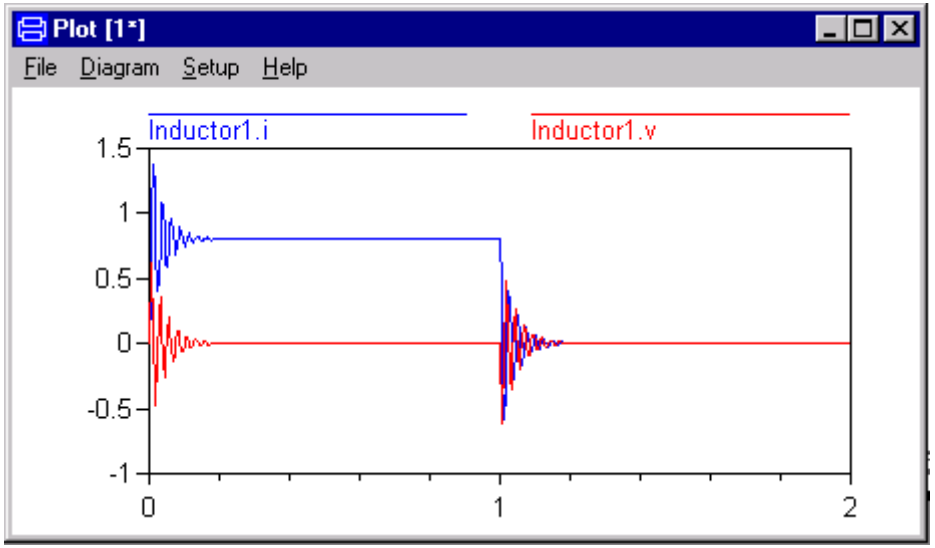
Task:

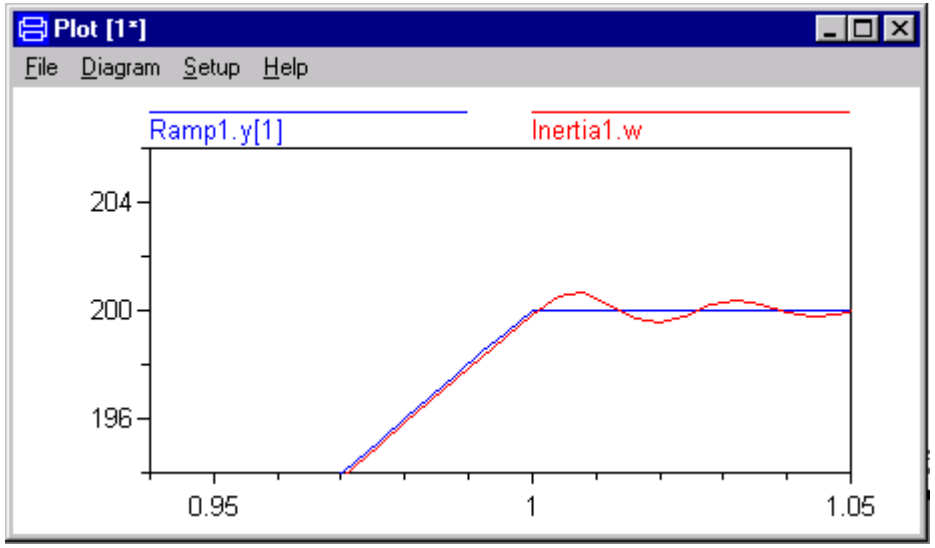
Model and simulate the following simple drive train
(dc-motor with load and ideal gear-box)
Simulate for 2 s
Try different values for the load inertia (inertia2 = 10 .. 30)



Components from:
Modelica.Blocks.Sources
Modelica.Electrical.Basic
Modelica.Mechanics.Rotational

Exercise 1: Result plot

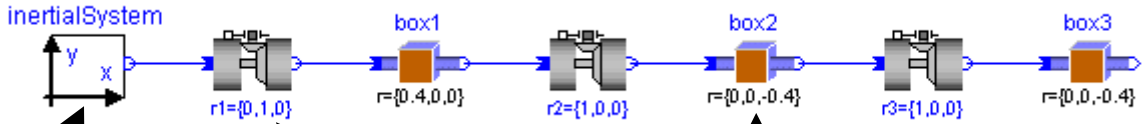




Exercise 2: Futura pendulum

Task:

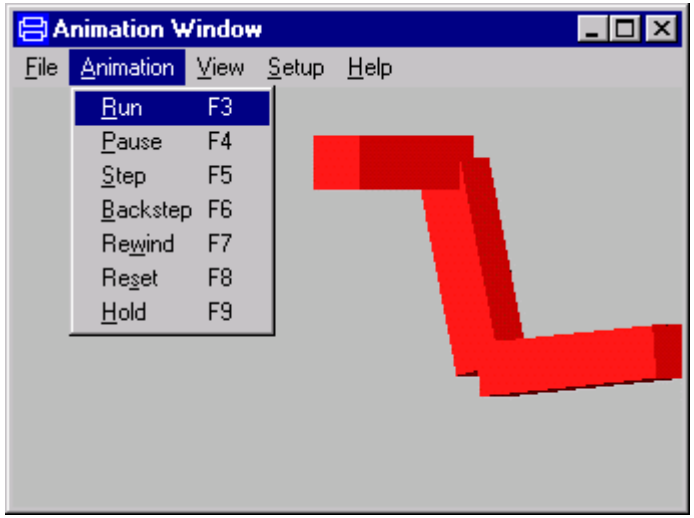
Model, simulate and animate the futura pendulum example using the **ModelicaAdditions.MultiBody** library. Simulate for 5 s



Inertial system
(Parts.InertialSystem)

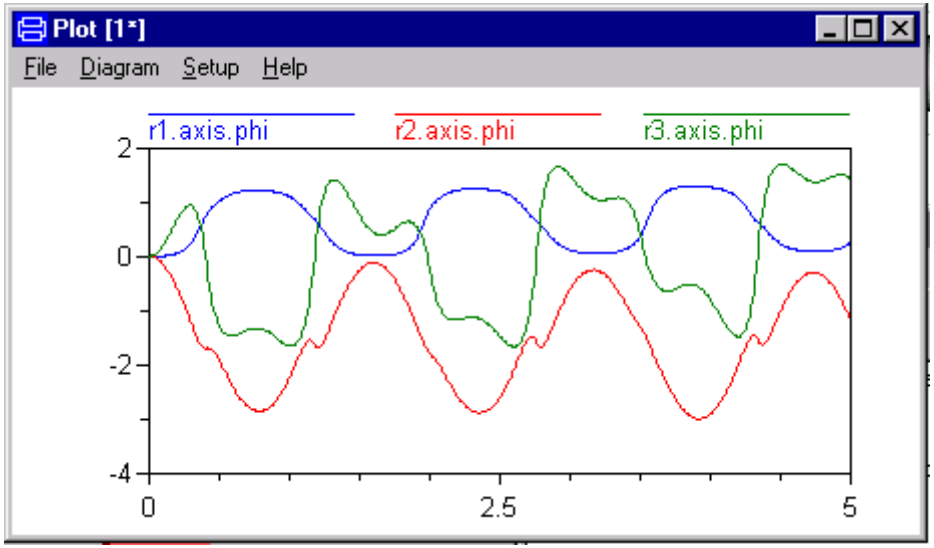
Swivel joint
(Joints.Revolute)

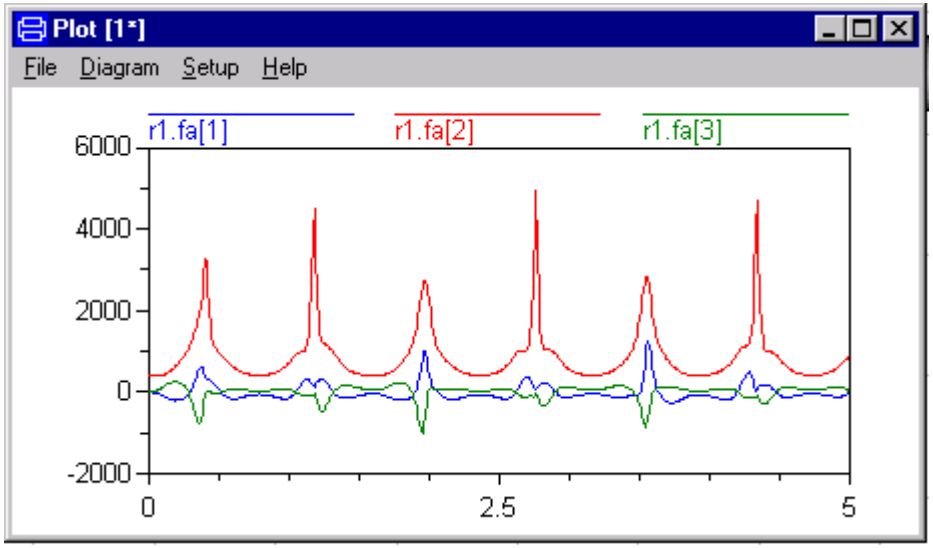
Quader
(Parts.BoxBody)



Animation of
the Simulation Results

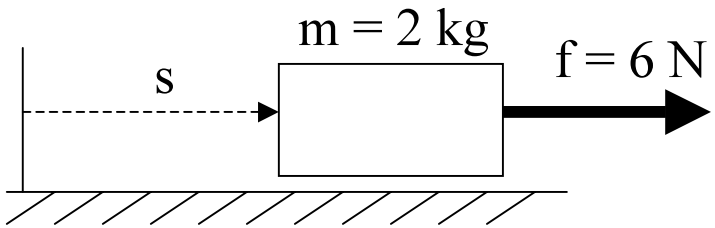
Exercise 2: Result plot





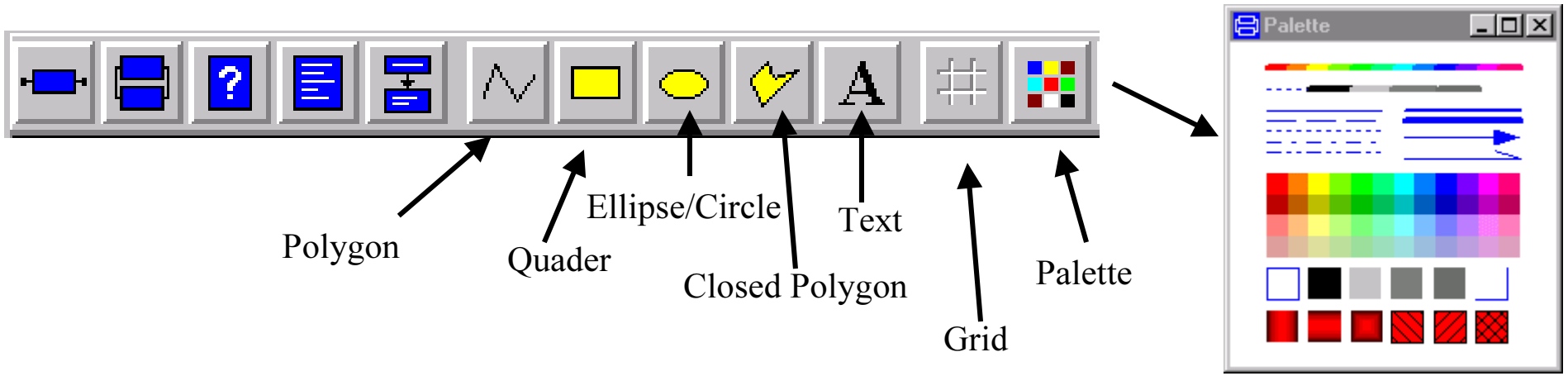
Exercise 3: Acceleration of mass

Task:
Model and simulate variant 3 of the accelerated mass example (see Tutorial)

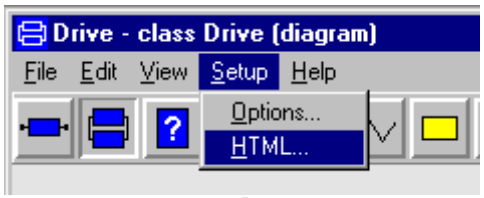


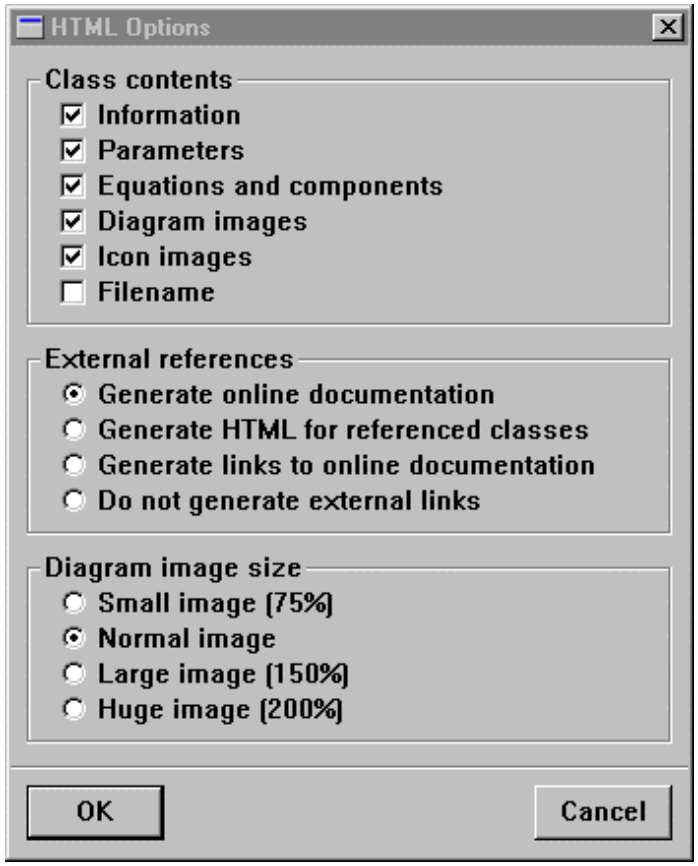
$$m \cdot \ddot{s} = f$$

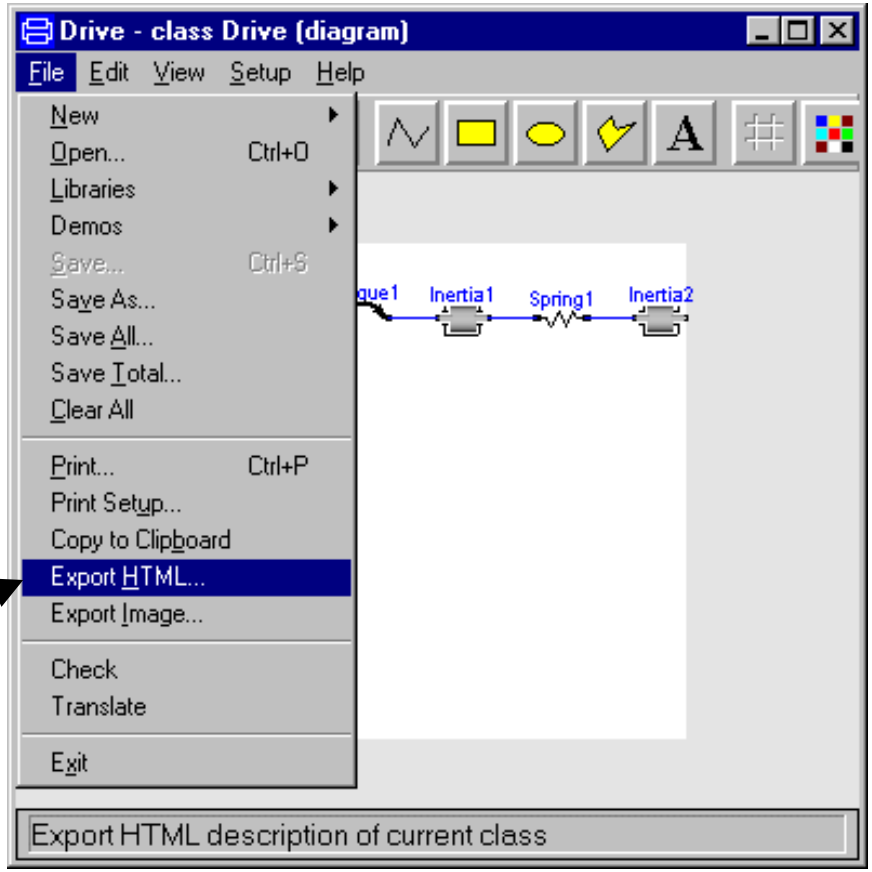
create a **picture** of the model within the **diagram layer** and
create an **HTML-documentation** of the complete model



Exercise 3: Create HTML-documentation





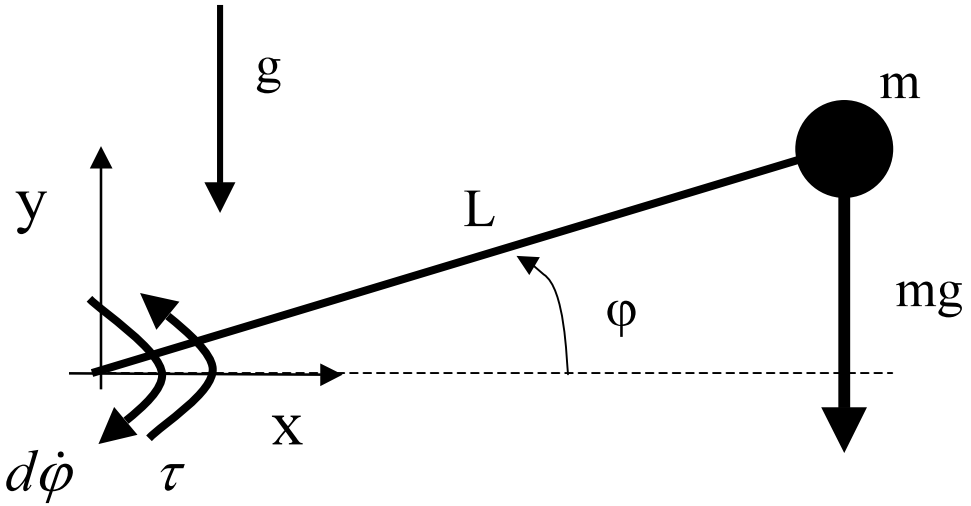


HTML-documentation is stored within the local help-directory.

Exercise 4: Different model variants of a pendulum

Task 1:

Create two different model versions of a robot arm (= Pendulum)
(a) using the component library ModelicaAdditions.**MultiBody**
(b) writing the **equations** by hand.
Simulate and check the results of both models!



Equation at the center of rotation: $$J \cdot \ddot{\varphi} = -m \cdot g \cdot L \cdot \cos(\varphi) - d \cdot \dot{\varphi} + \tau$$

Exercise 4:a) Pendulum - with ModelicaAdditions.MultiBody

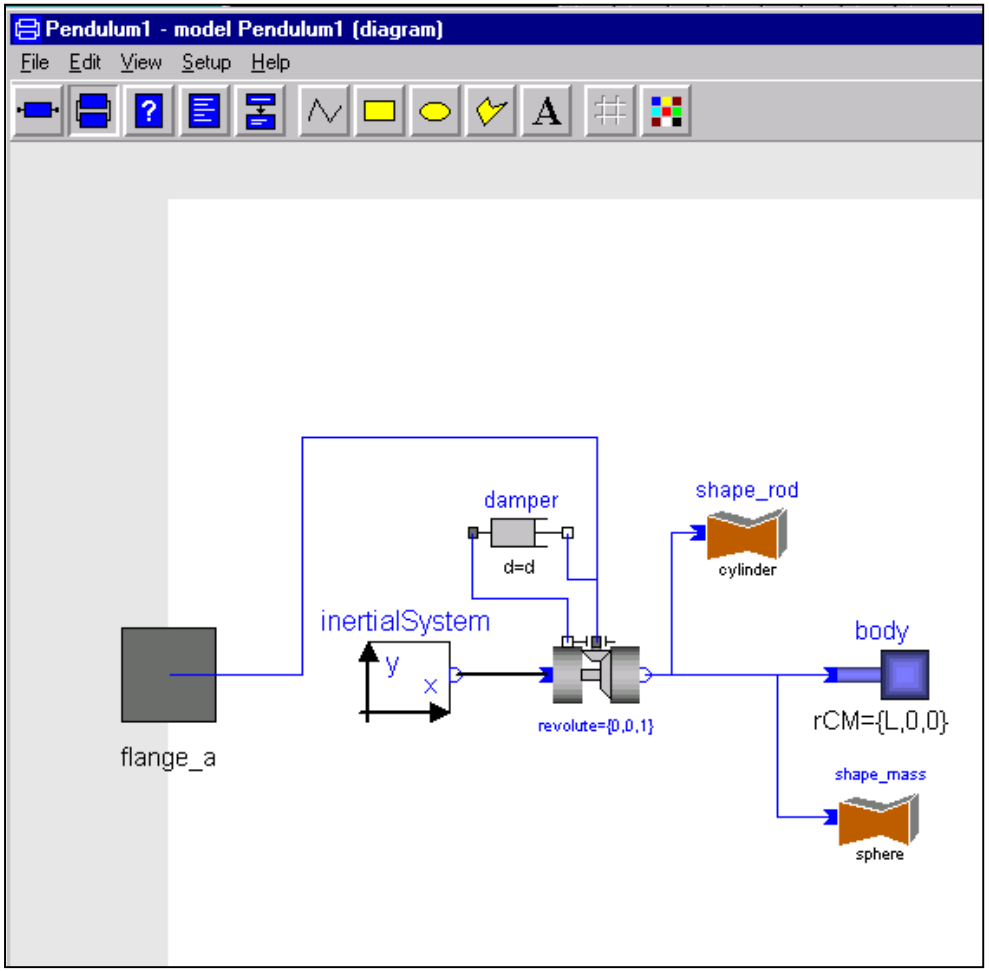
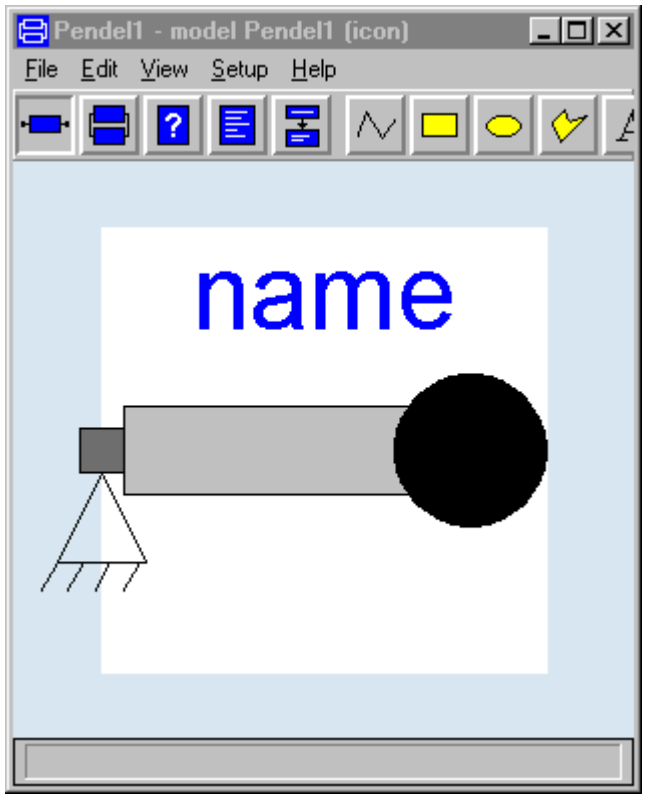
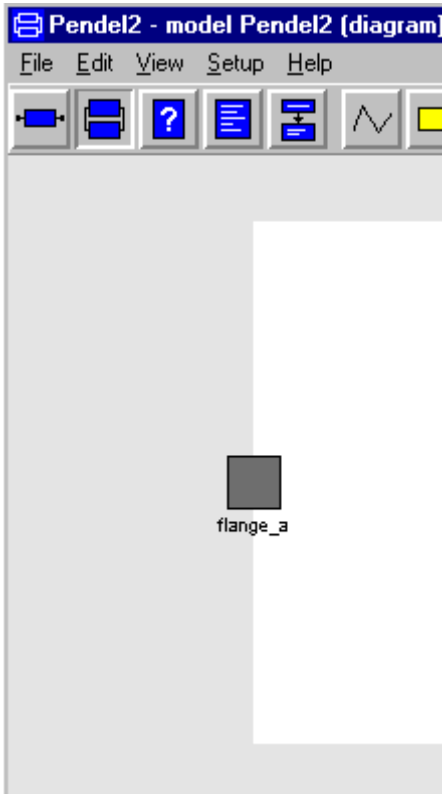


diagram layer



icon layer

Exercise 4:b) Pendulum – equations by hand



Pendel2 - model Pendel2 (equation)

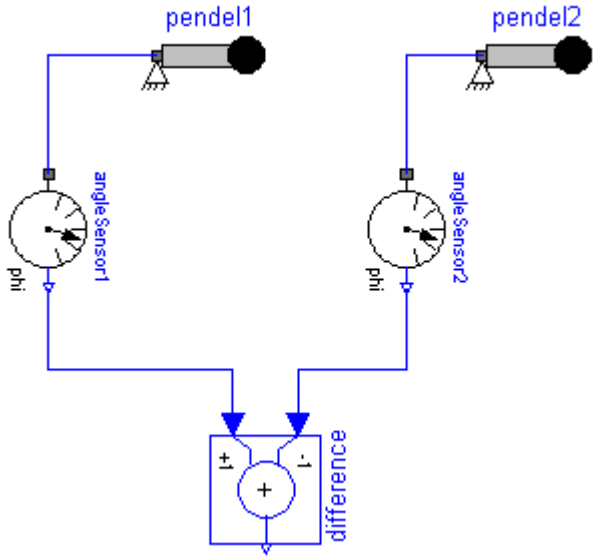
File Edit View Setup Help

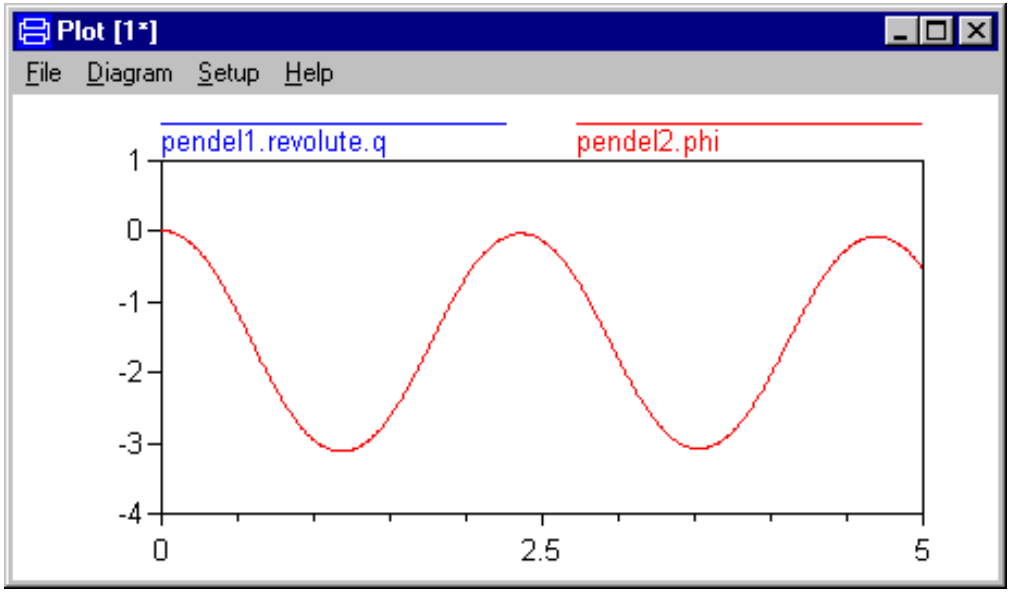
```
package SI = Modelica.SIunits ;
parameter SI.Length L=1 "Stablaenge";
parameter SI.Mass m=5 "Masse des Massenpunkts";
parameter SI.Damping d=0.1 "Daempfung im Gelenk";
parameter SI.Acceleration g=9.81 "Gravitationsbeschleunigu
SI.Angle phi "Rotationswinkel vom Pendel";
SI.AngularVelocity w "Winkelgeschwindigkeit vom Pendel";
```

```
equation
  phi = flange_a.phi;
  w = der(phi);
  m*L*L*der(w) + d*w + m*g*L*cos(phi) = flange_a.tau;
```

Exercise 4: Comparison of the two models

Compare the simulation results of both pendulum models (angle, etc.)

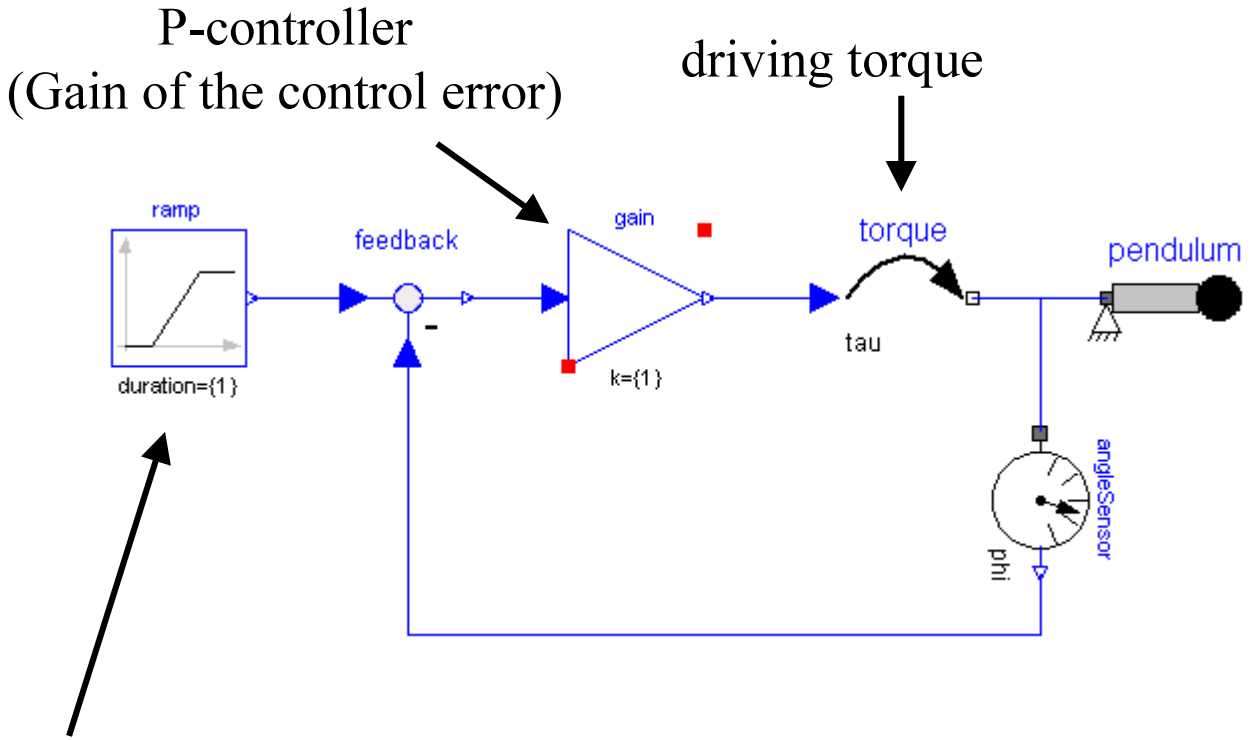




Exercise 4: Simple control system of the robot arm

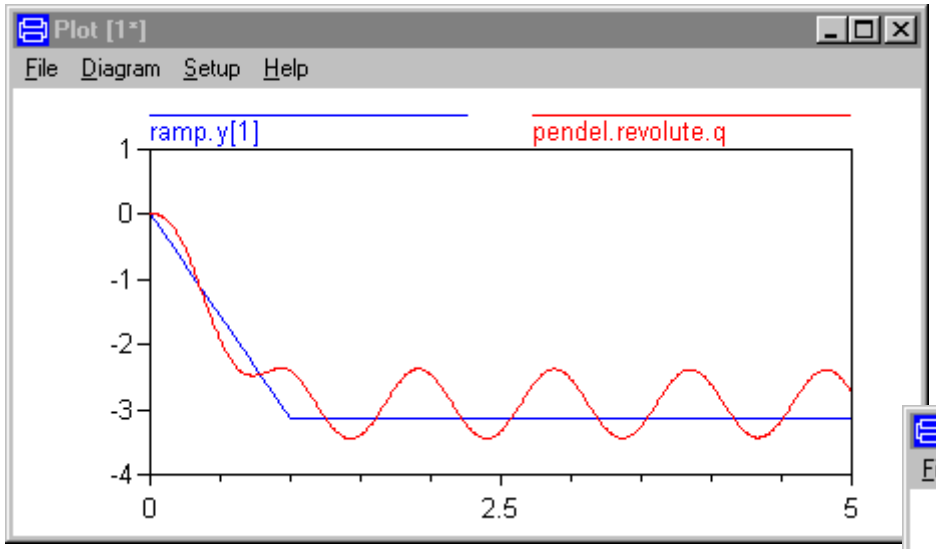
Task 2:

Build up a simple control system to easily change the angle of the robot arm:



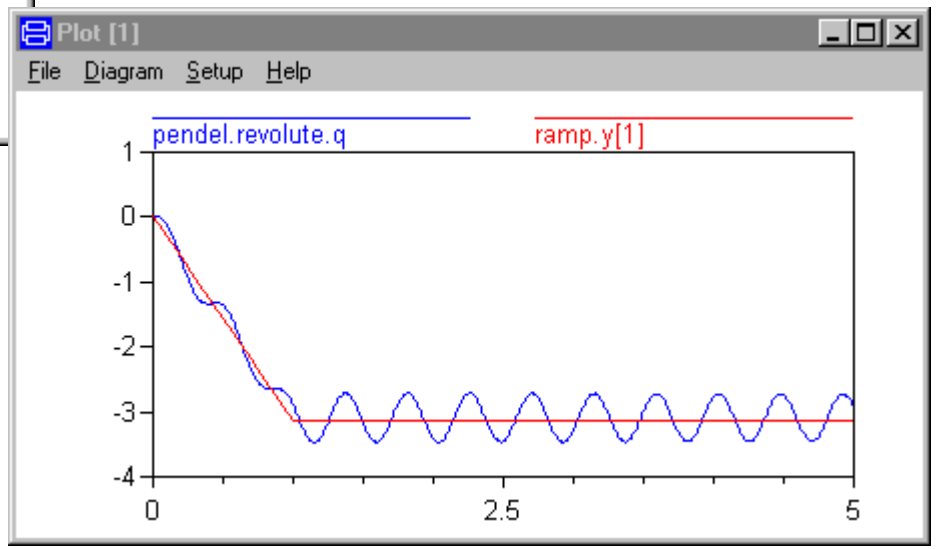
Exercise 4: Simple control system of the robot arm

Try different values for the gain of the controller
for example: gain.k = 1, 10, 100, 1000, 200



k = 200

k = 1000



Control structure is to simple!

Exercise 5: 1-dimensional heat flow

Build up a library of components to model 1-dimensional heat flow (`package SI=Modelica.SIunits`):

```
connector HeatCut_a
   SI.Temperature   T;
   flow SI.HeatFLux q;
end HeatCut_a;
```

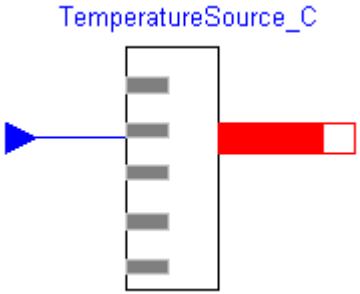
```
connector HeatCut_b
  SI.Temperature T;
  flow SI.HeatFLux q;
end HeatCut_b;
```


```
model HeatResistance „only heat transport"
    parameter SI.Area   A
    parameter SI.Length L
    parameter SI.ThermalConductivity lambda;
    HeatCut_a a; HeatCut_b b ;
equation
    a.q = lambda*A/L*(a.T - b.T);
    0   = a.q + b.q;
end HeatResistance
```

Exercise 5: HeatLib-library

HeatCapacitance

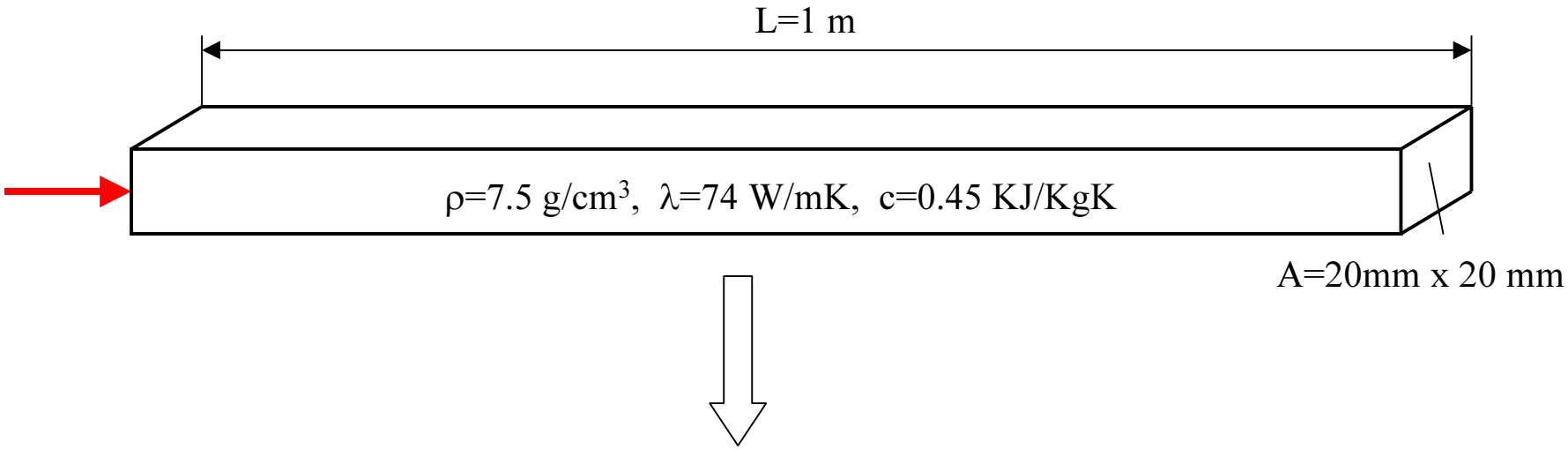
```
model HeatCapacitance „only heat storage"
  parameter SI.Mass m;
  parameter SI.SpecificHeatCapacity c;
  parameter SI.Temp_C T0 "initial Temp.";
  HeatCut_a a(T(start=T0-Modelica.Constants.T_zero));
equation
  a.q = c*m*der(a.T);
end HeatCapacitance;
```



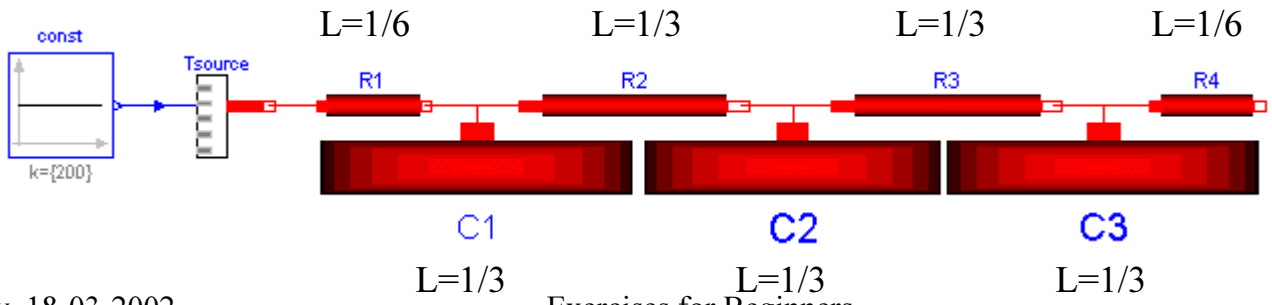

```
model TemperatureSource_C „fixed temperature"
  Modelica.Blocks.Interfaces.InPort inPort(n=1);
  HeatCut_b b;
equation
  b.T = inPort.signal[1]- Modelica.Constants.T_zero;
end TemperatureSource_C;
```

Exercise 5: Modeling problem

A completely isolated rod shall be modeled which has a temperature of 20^0 Celsius everywhere. At the left end, suddenly the temperature is raised to a fixed temperature of 200^0 Celsius:

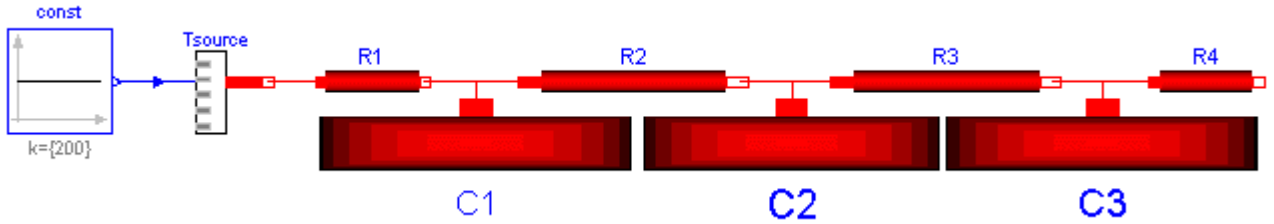


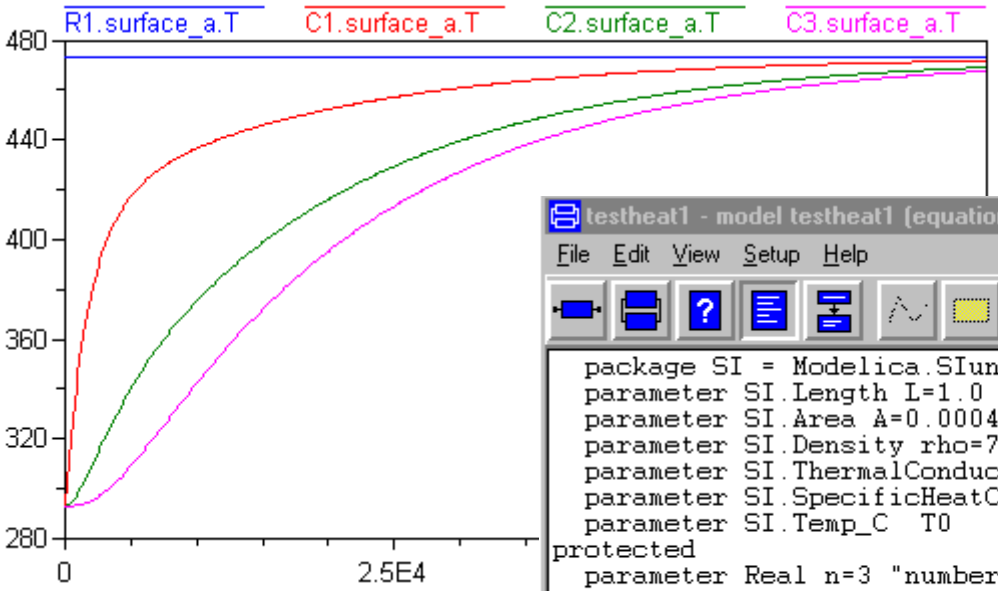
discretized rod in base elements of your library



Exercise 5: Modeling with single elements

Task 1: Build up and simulate the rod with single elements (n=3).





Important: Introduce reasonable parameters

testheat1 - model testheat1 (equation)

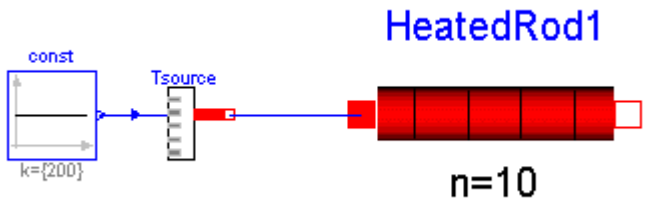
File Edit View Setup Help

```
  package SI = Modelica.SIunits;
  parameter SI.Length L=1.0 "Length of rod";
  parameter SI.Area A=0.0004 "Area of rod";
  parameter SI.Density rho=7.5*1000 "Density of rod material";
  parameter SI.ThermalConductivity lambda=74 "Thermal conductivity of material";
  parameter SI.SpecificHeatCapacity c=450 "spec. heat capacity of material";
  parameter SI.Temp_C  T0    =20     "initial temperature";
protected
  parameter Real n=3 "number of elements";
  parameter SI.Length Lelem=L/n "length of a heat capacity element";
  parameter SI.Mass mElem=A*Lelem*rho "mass of a heat capacity element";
```

Exercise 5: Modeling with component arrays

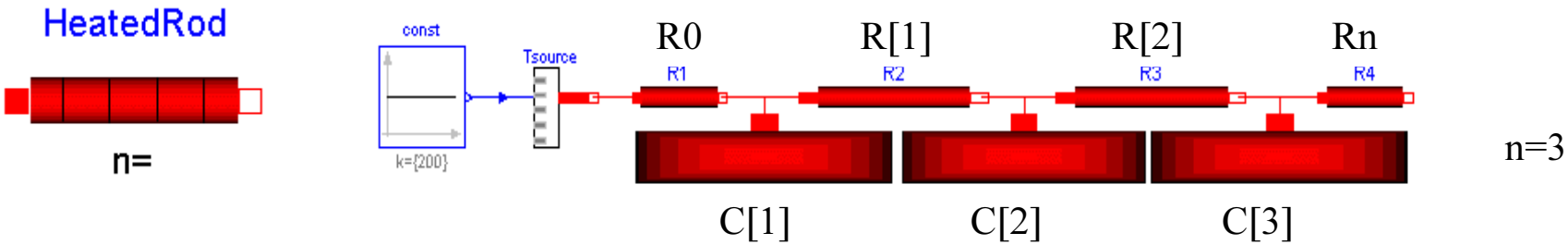
Usage of **single elements** has the **disadvantage**, that it is inconvenient to change to a **finer discretization**.

Better: use **component arrays**

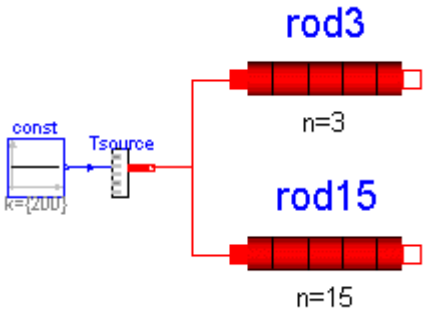


Task 2:

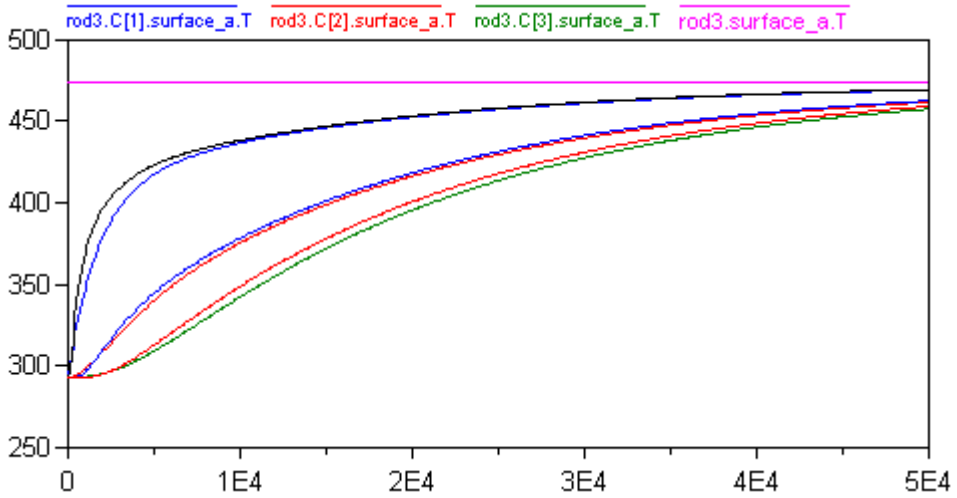
Model the rod with component arrays, such that the number n of discretized elements can be changed with one parameter. Compare simulations for n=3 and n=10. Simulate for 5e4 s.



Exercise 5: Result plot



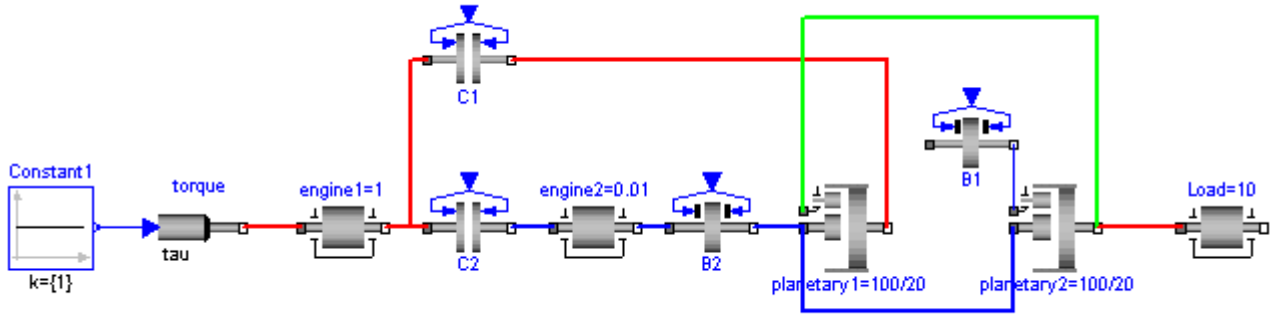
compare **n=3** with **n=15**
(at n=15
rod3.C[1] und rod15.C[3],
rod3.C[2] und rod15.C[8],
rod3.C[3] und rod15.C[12]
are at the same place).



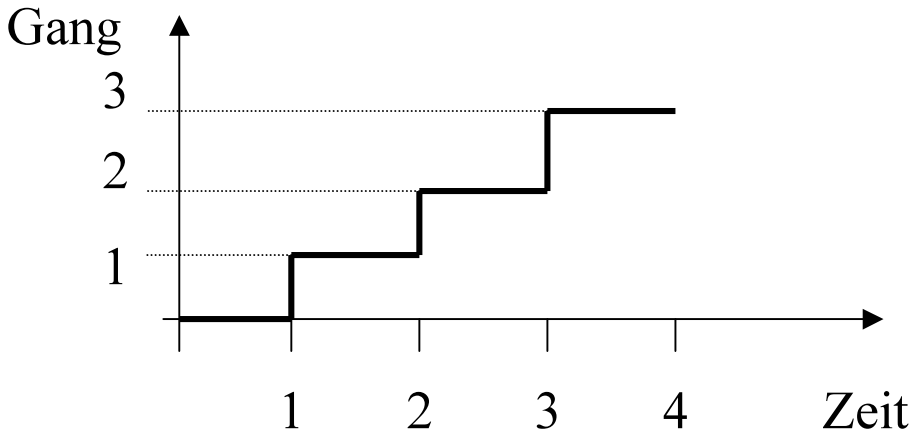
Exercise 6: Simple automatic gearbox

Task 1:

Build up the following model of a simple automatic gearbox and design a simple control for the clutches C1, C2 and brakes B1, B2, and simulate the gear shift from the figure at the lower, right part (shift from gear 1 to gear 3). Use the default value for the friction characteristics for all clutches and brakes.

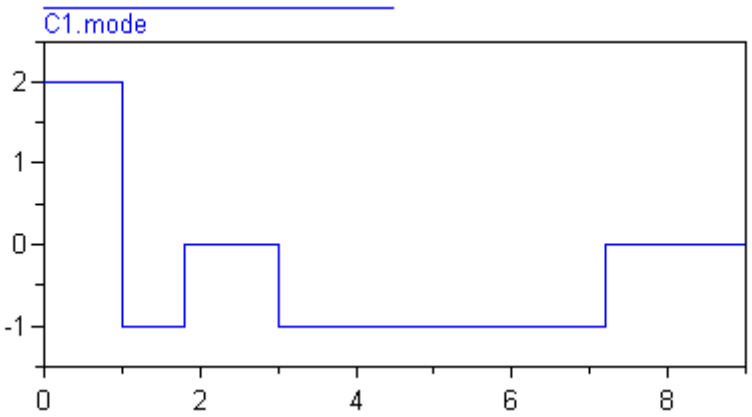


Gang	C1	C2	B1	B2
0				
1	on		on	
2	on			on
3	on	on		



Exercise 6: Fixing an appropriate contact pressure

Task 2: Determine an appropriate contact pressure of the Clutch.



Clutch.**mode**

= **2**: clutch is **not active**

= **1**: **forward sliding**

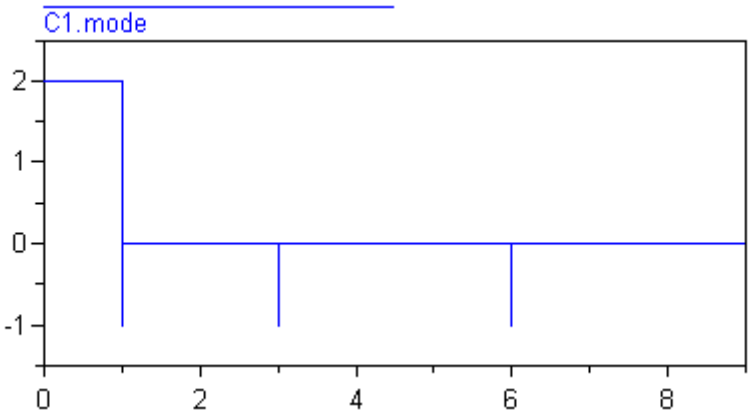
= **0**: **stuck** (no relative motion)

= **-1**: **backward sliding**

k=200

Clutch slips a lot

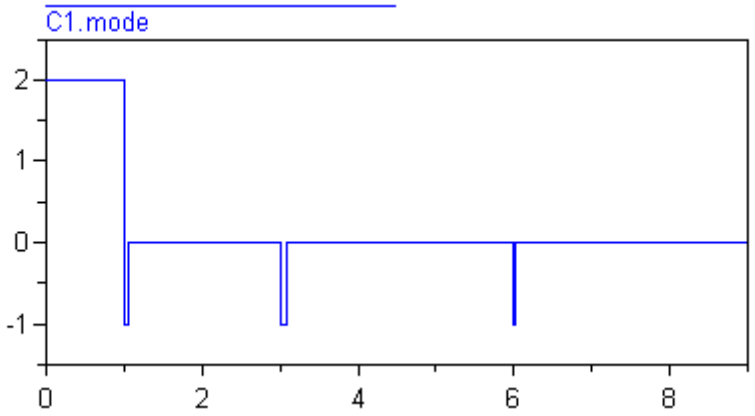
(heavy losses and wastage)



k=10000

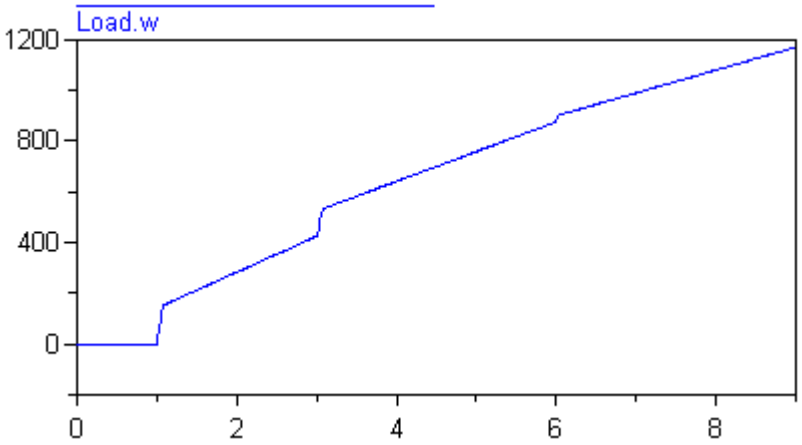
Clutch does not slip at all
(jerky changing gears)

Exercise 6: Fixing an appropriate contact pressure



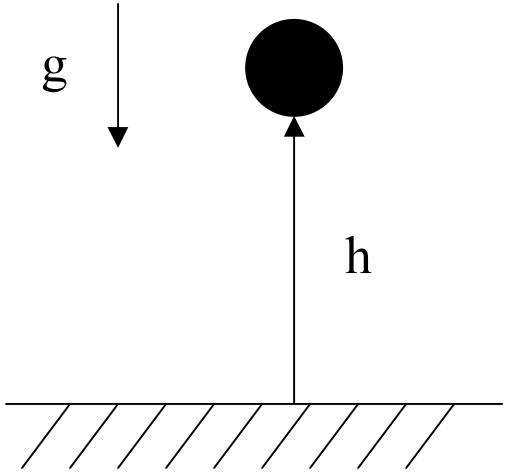
k = 2000

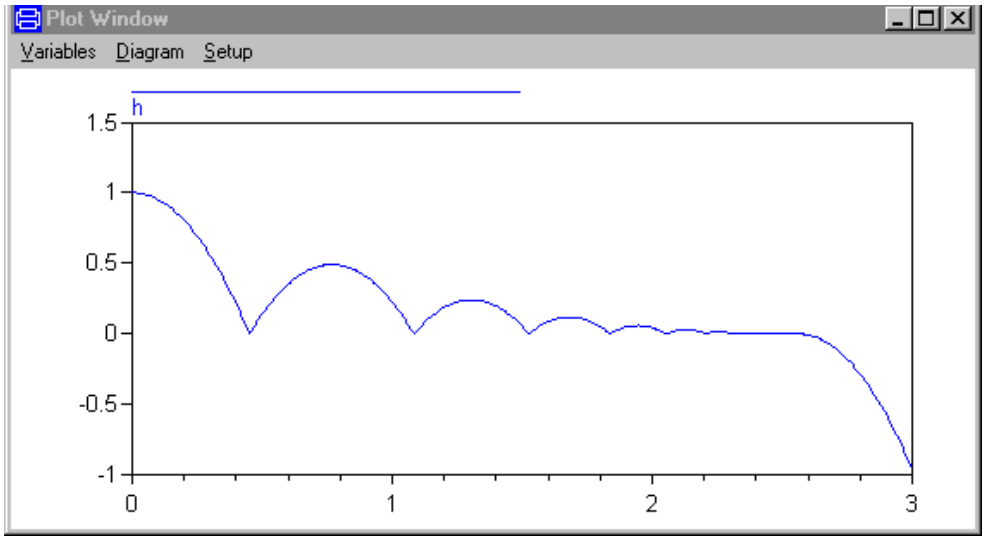
good compromise



Drehzahl von Last

Exercise 7: Bouncing ball






```
model bouncingBall
  parameter Real e=0.7;
  parameter Real g=9.81;
  Real h(start=1);
  Real v;
equation
  der(h) = v;
  der(v) = -g;

  when h <= 0 then
    reinit(v, -e*pre(v));
  end when;
end bouncingBall;
```

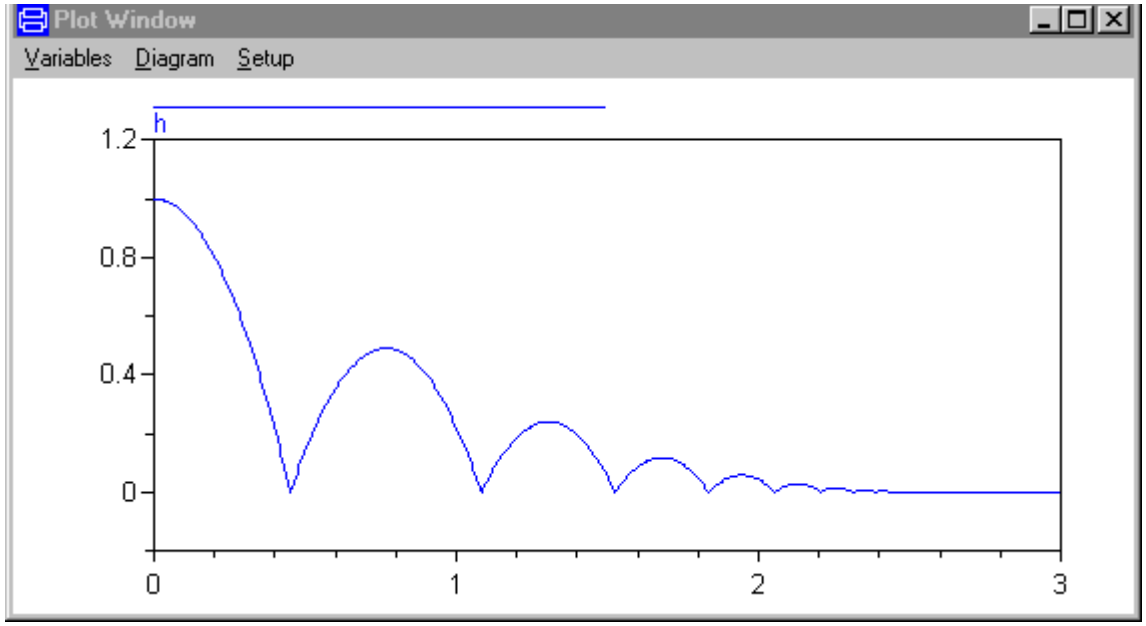
Exercise 7: Bouncing ball

Task:

(a) Reproduce the simulation result of page 75 (see Tutorial)

(b) Modify the model of the bouncing ball, so that the ball finally stays on the ground.

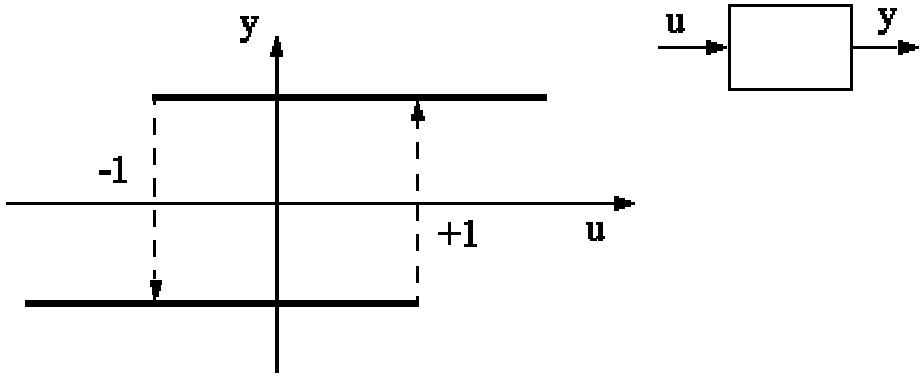
Idea: Change the model structure, if the velocity after the impulse is too small. (= Acceleration of the ball vanishes).



Exercise 8: Modeling of an Hysteresis

Task:

Build up a Modelica model of the following Hysteresis block. Use the **pre**-operator as memory to detect which branch of the characteristic curve has been calculated in the past.



Test the Hysteresis-block with a sinus-signal u=sin(w*time)