Introduction to the Modelica language and the simulating environment Dymola, part I

Staffan Haugwitz

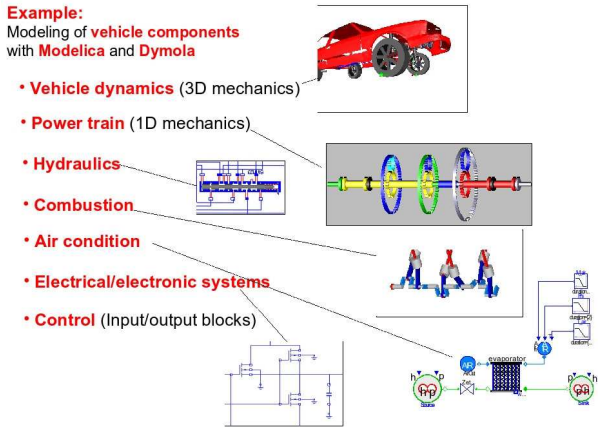
Dept. of Automatic Control
Lund University

Based on material from Modelon

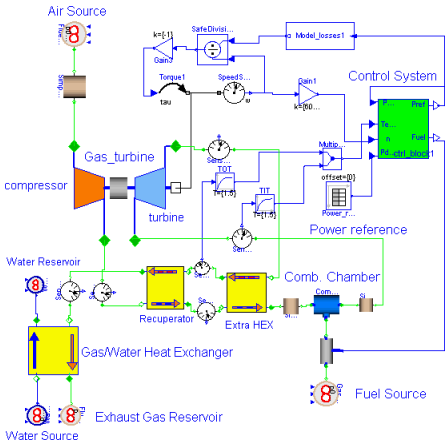
Modelica Lecture - Outline

- Introduction, relation between Modelica and Dymola, applications, overview of Dymola
- Introduction to the Modelica language, syntax and constructs
- Traps and pitfalls (high index problem, stiff systems)

Modelica in Vehicle Modeling



Modelica in Microturbine Systems



Modelica Features

- Declarative equations
  Write the model equations as they are derived from physics, instead of deriving the right hand side.

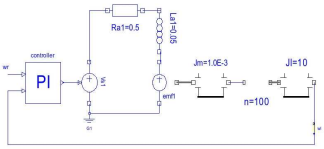
$$C\frac{dP_2}{dt} = F_4 - F_5; \qquad \lambda F_4 = F_1 c_p (T_2 - T_1); \qquad \frac{dP_2}{dt} = \frac{1}{C}\left(\frac{F_1 c_p (T_2 - T_1)}{\lambda} - F_5\right);$$

- Object-oriented modeling language
  (classes, instances, inheritence...)
- Component models
  Easy to build libraries of reusable code. Modelica standard library contains 740 different models, ready to use or modify.

Modelica vs Simulink

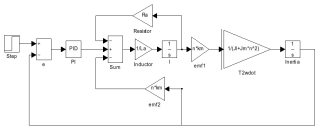
Acausal modeling

Declarative languages just require the developer to define the problem at a higher level and leaves the solution to the simulation tool

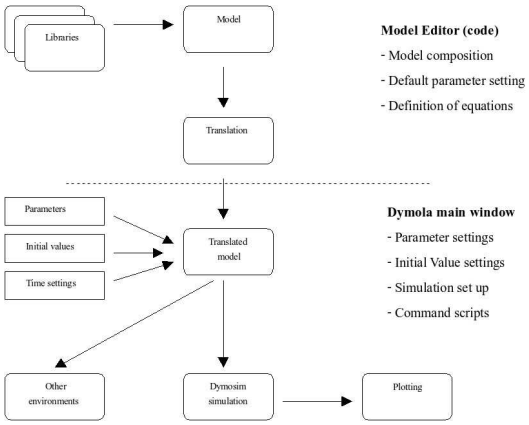


Block-oriented modeling

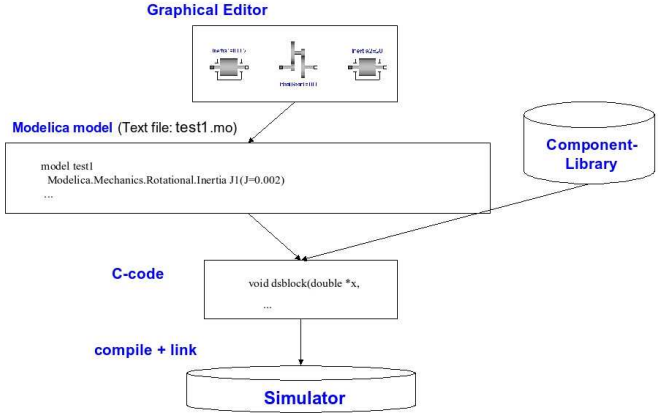
Procedural programmes require the developer to define the order that calculations are to be done in



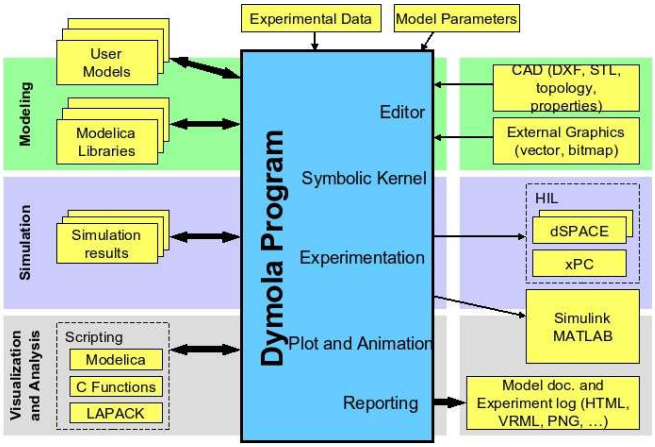
Modelica vs Dymola



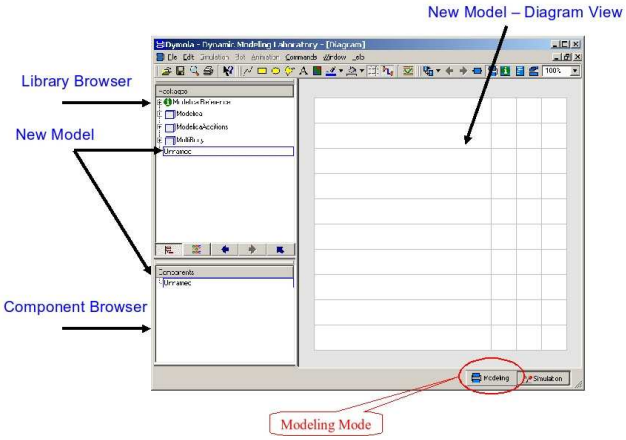
Dymola Overview 1



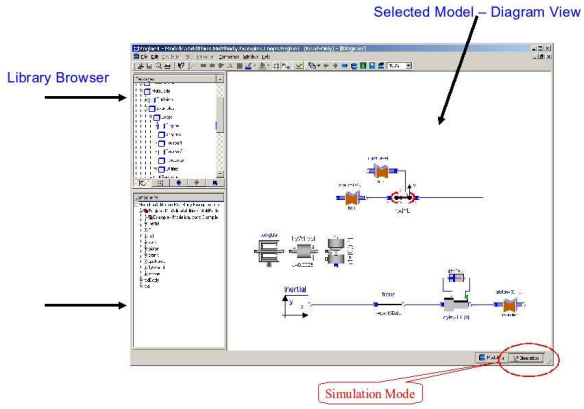
Dymola Overview 2



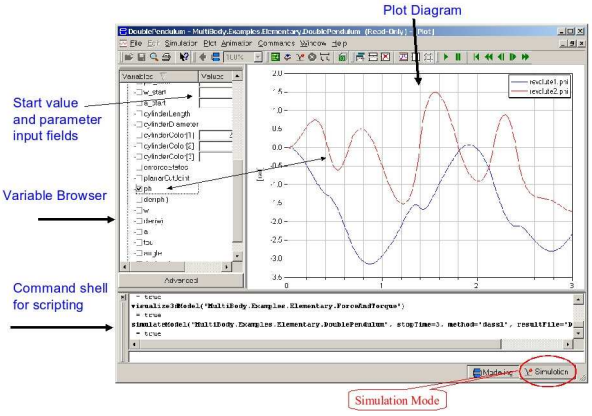
Dymola Introduction 1



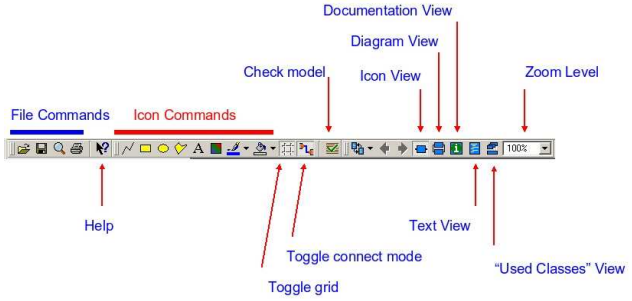
Dymola Introduction 2



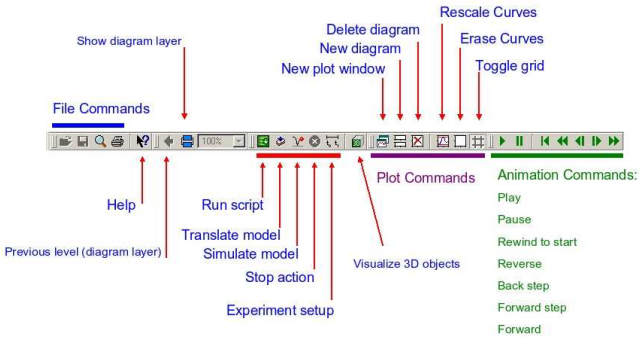
Dymola Introduction 3



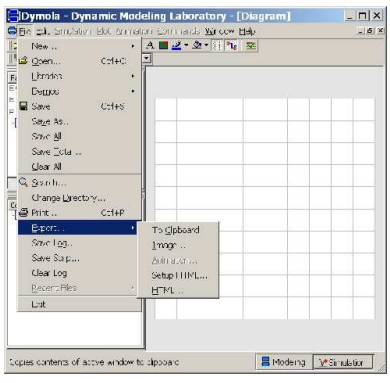
Dymola Introduction 4



Dymola Introduction 5



Dymola Introduction 6



Save As ...
save under a new name

Save Total ...
save including all library models for stand-alone use and support issues

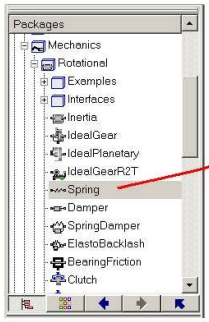
Clear All ...
Clear all models in Dymola (restart)

To Clipboard ...
export as EMF to clipboard

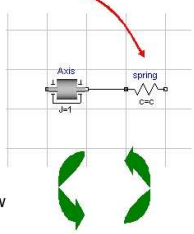
Save Script ...
record actions as macro and save them

HTML ...
Generate html documentation

Dymola Introduction 7



Modelica-Library:
Modelica.Mechanics.Rotational



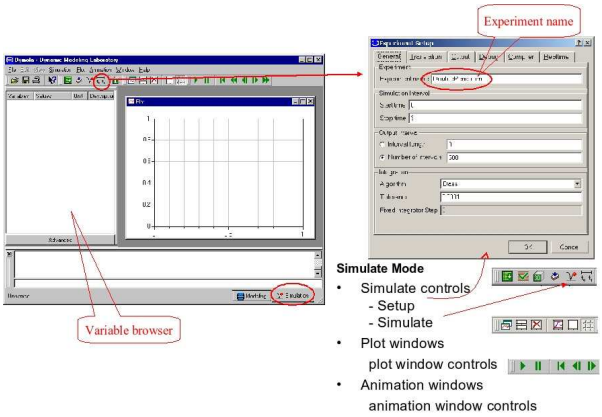
Text View and Graphical View are equivalent

```
model Unnamed
  Modelica.Mechanics.Rotational.Inertia Axis a;
  Modelica.Mechanics.Rotational.Spring spring a;
equation
  connect(Axis.flange_b, spring.flange_a) a;
end Unnamed;
```

Dymola Introduction 8

- Build model from Libraries (Modeling Mode)
  - Advanced: build packages to organize own libraries
  - Advanced: Use Modelica language for own models
- Run model (Simulation Mode)
  - Single run
  - Multiple runs via scripts
  - Optimize
- Post-process results (Simulation Mode)
  - Plot
  - Animate
  - Analyze (scripting, Modelica language, external tools)

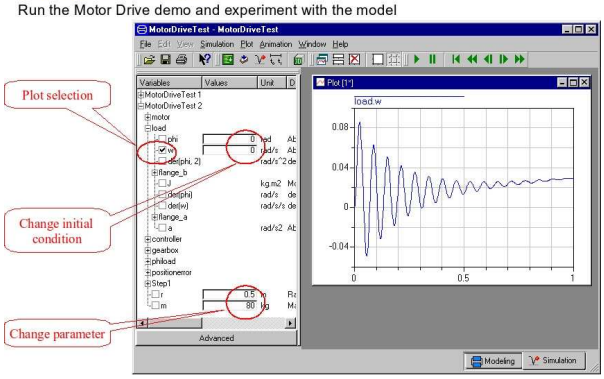
Simulation in Dymola 1



Simulation in Dymola 2

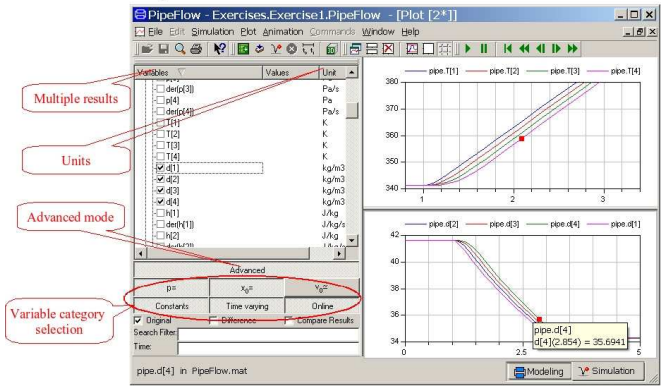
Simulate an existing model

Run the Motor Drive demo and experiment with the model



Simulation in Dymola 3

Advanced experiments



Live Demo 1 - Pendulum

Build model by editing text equations

- Simple planar pendulum
- Two states

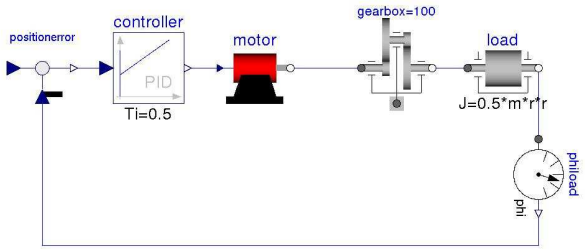
$$\ddot{\theta} = -\frac{g}{L}\sin\theta \qquad (1)$$

```
model SimplePendulum
  constant Real g = 9.81;
  parameter Real L = 1;
  Real Theta(start = 0.1);
  Real ThetaDot;
equation
  ThetaDot = der(Theta);
  der(ThetaDot) = - g/L*sin(Theta);
end SimplePendulum;
```

Live Demo 2 - MotorDrive

Build model by using existing components

- Simple DC motor with ideal gearbox and inertia
- Three states



Dymola features - Linearization

- Linearize model, from “Simulation”-menu, beware linearization is at the start time $t = 0$!
- Simulate model to the desired linearization point
- Set the final state as the initial state with `importInitial();`
- Linearize as usual
- Load file `dslin.mat` into Matlab with e.g. `tloadlin`, many specific matlab routines delivered with Dymola.