

Überblick und Aktuelles zu Dymola



Inhalt

- Dymola
- Modelica
- Modellieren mit Dymola
- Modellbibliotheken
- Simulieren mit Dymola
- Dymola-Simulink-Interface
- Modelica 2.1
- Dymola 5.4

Dymola

- Dymola ist eine Software zur multidisziplinären Modellierung und Simulation
- entwickelt von Dynasim AB in Schweden
- Vertrieb im deutschsprachigen Raum: Bausch-Gall GmbH

BAUSCH-GALL GmbH

- Seit 1987
- Hans Gall und Dr. Ingrid Bausch-Gall seit 1981 bzw. 1983 selbständig
- Vertrieb von Simulationssoftware
- Projektarbeit
- Schulungen
 - Dymola, Spice, Simulink, Matlab

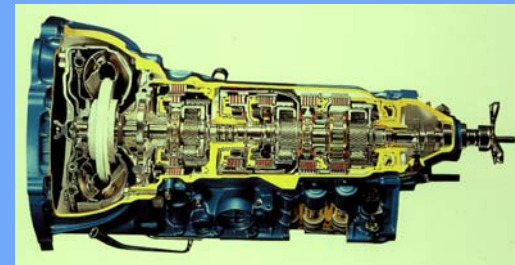
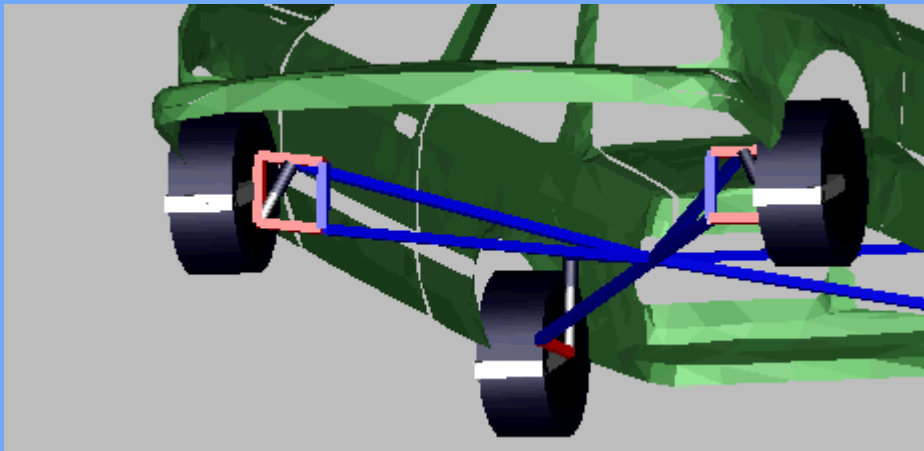
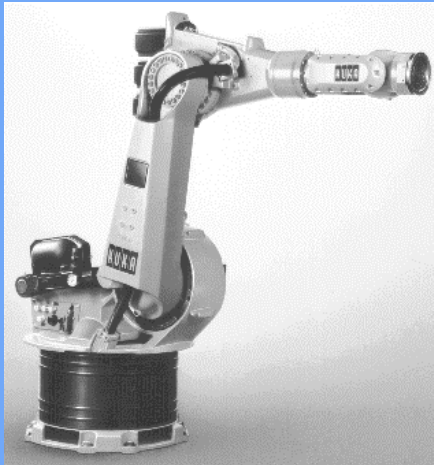
Dymola

- Ziel: einheitliche Modellierung verschiedener Ingenieurdisziplinen
- einige Stichworte
 - “Multi-Engineering modelling”
 - Nichtkausale Modellierung
 - Graphische Modellierungsoberfläche
 - Schnittstelle zu anderen Programmen

Modelica

- neue Methode
 - objektorientiertes Modellieren
 - gleichungsorientiertes Modellieren
- austauschbare, standardisierte formale Sprache
 - zum Modellaustausch
 - Weiterverwendung von Modellierungswissen
- Modelica --- Sprachdefinition
www.modelica.org
- Dymola --- Software zur Modellierung und Simulation mit Modelica
www.dynasim.se

Anwendungsbeispiele



Modelica Design Group

- Zusammenschluß von Simulationsexperten
- ehrenamtlich
- mehrere Treffen jährlich
- Erstellung und Veröffentlichung der Sprachdefinition
- Pflege der Homepage

Ziele der Modelica Design Group

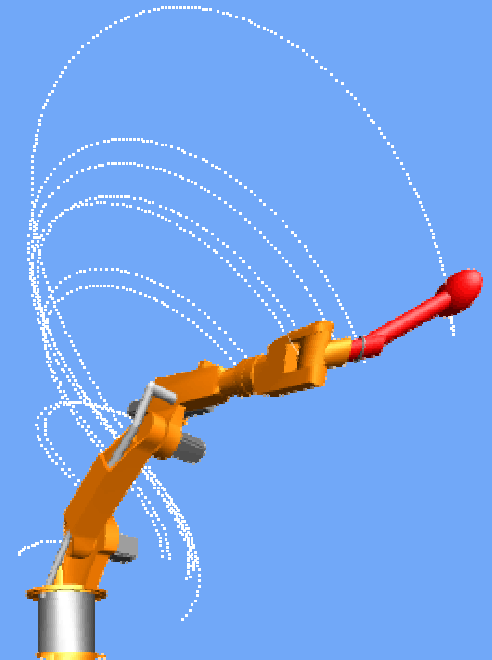
- offene Sprachdefinition
- objektorientierte Sprachen vereinheitlichen
- Wiederverwendbarkeit der Modelle
- Deklarative, d.h. mathematische Gleichungen statt prozedurale Zuweisungen wie in einer Programmiersprache
- Sprache soll effiziente Simulation unterstützen

Modelica - Entwicklungsstand

- Entwurf startete im September 1996
- Version 1.0 im September 1997
- Heute: Version 2.1
- Werkzeuge und Bibliotheken stehen zur Verfügung
- Homepage: www.modelica.org
- regelmäßige Tagungen
- nächste Tagung: März 2005 an der TU HH

Die Firma Dynasim

- Entwickler von Dymola
- gegründet 1992 von Hilding Elmqvist
- Jetzt: 7 Angestellte, davon 5 technisch/wissenschaftlich
- Zusammenarbeit mit Wissenschaftlern
- Kunden weltweit, z.B.: Toyota, Volvo, Rover, Ford, GM
- Deutschland z.B.: ABB, BMW, DaimlerChrysler, Hella, ZFF, Volkswagen, Opel, Universitäten und Forschungsinstitute
- Österreich, z.B.: AVL, Elin, MAGNA STEYR, TU Wien

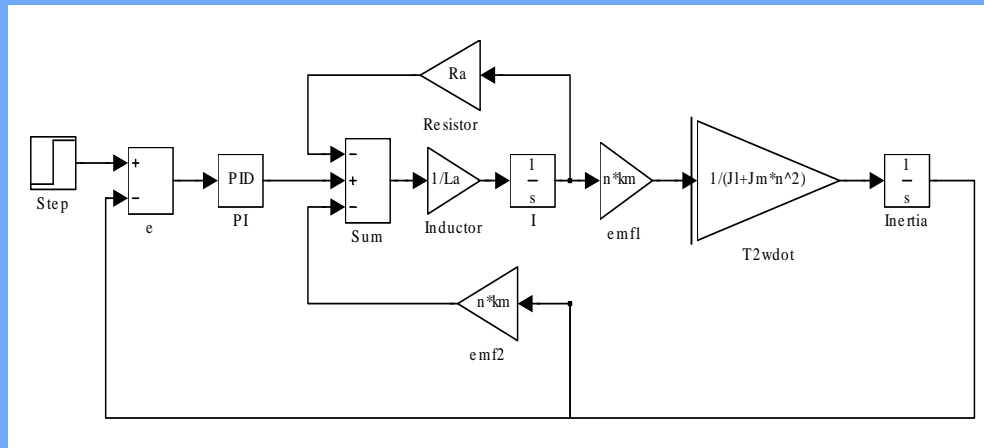
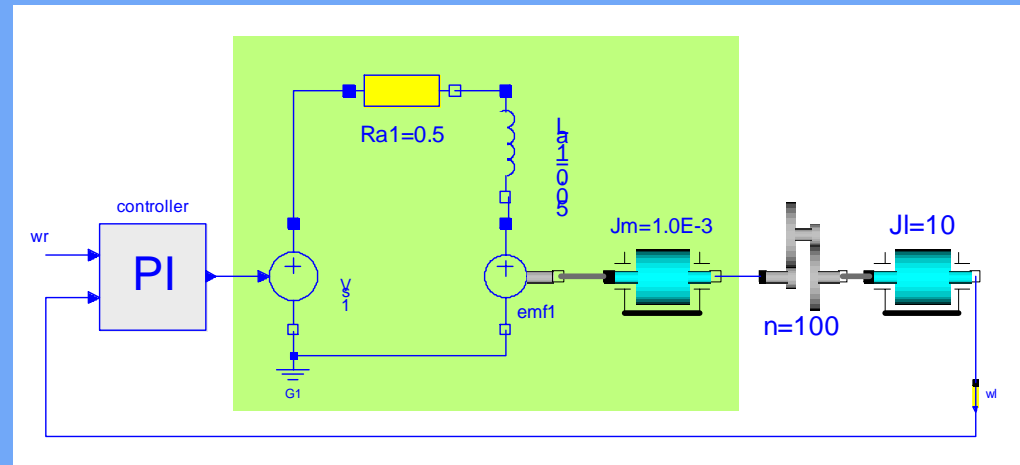


Entwicklungsziele

- Software zur Modellierung komplexer Systeme
- Modelle verschiedener Ingenieurdisziplinen
- Verfügbarkeit der Modelle als Quellcode
- homogenes Simulationsmodell
- effiziente Simulation
- Verkürzung der Entwicklungszeit

Energieflußorientierte Modellierung

Dymola

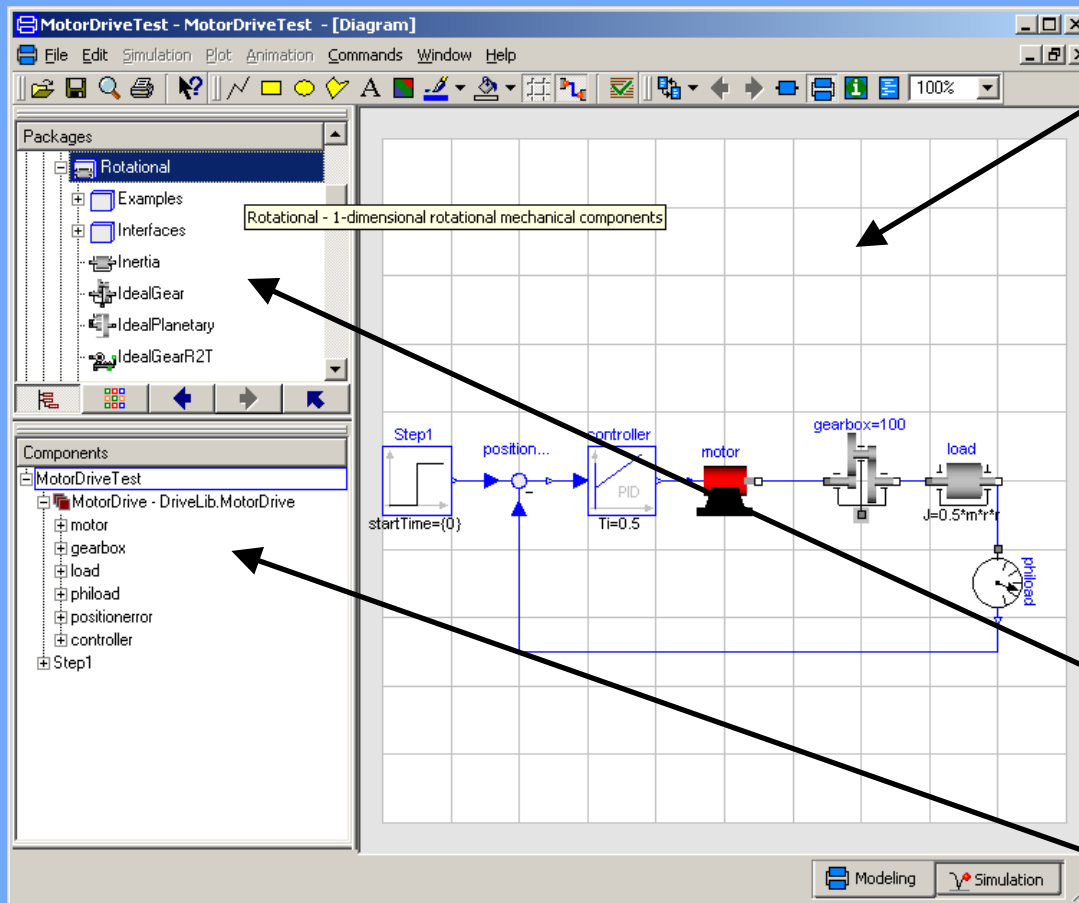


signalflußorientierte Modellierung

Modellierung mit Dymola

- versteht Modelica
- Gleichungen als Modellbasis
- Variablen, Typen, Einheiten (units)
- objektorientierte Struktur der Modelle
Klassen, Konnektoren, Blöcke
- Modellbibliotheken, sog. Packages
- Felder und Matrizen, einschl. Operatoren
- Funktionen und Algorithmen
- symbolische Manipulation der Gleichungen

Dymola-Modellierungsfenster



Modellierungsfenster

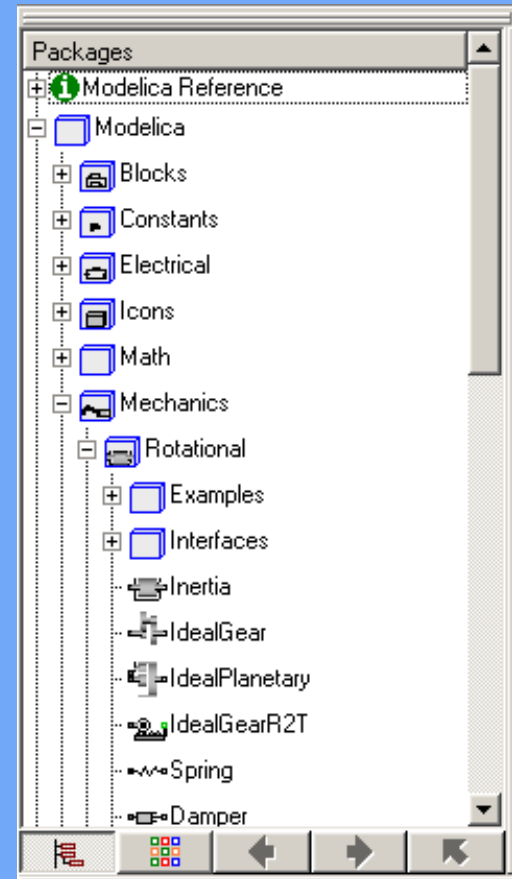
- Icon Layer: Definition der Graphik
- Diagram Layer: Zusammenstellen des Modells
- Documentation Layer
HTML-Dokumentation einfügen
- Equation Layer:
Gleichungen als Modelica-Kode

Model Library Browser

Model Browser

Modelica-Bibliotheken

- Hierarchisch
- Komponenten lassen sich sinnvoll anordnen
- grosse Komponentenvielfalt
- freie Bibliotheken enthalten z.B.:
 - einfache Regelungstechnik (Blöcke)
 - 1D-Mechanik (rotatorisch, translatorisch)
 - analoge und digitale Elektrik
 - Mehrkörpersysteme
 - einfache Pneumatik und Hydraulik
 - Fahrdynamik (VehicleDynamics)

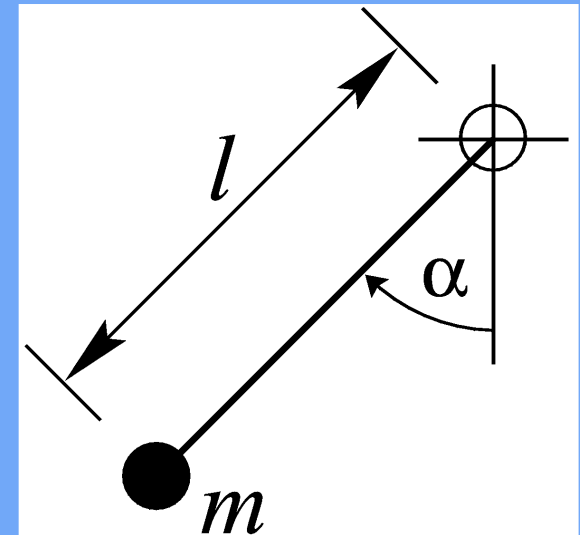


Modelica-Bibliotheken

- kommerzielle Bibliotheken für
 - Antriebsstränge (Powertrain)
 - Pneumatik
 - Hydraulik
 - Klimaanlage (AirCondition)
- weitere freie Bibliotheken
www.modelica.org

Grundlage der Modellbildung in Dymola

- Modellierungsgrundlage:
 - Differentialgleichungssysteme
 - Differenzengleichungen
 - algebraische Gleichungen
- Einfacher Fall: Pendel



$$ml\ddot{\alpha} + mg \sin(\alpha) + dl\dot{\alpha} = 0$$

Dymola-Modell

The screenshot displays the Dymola software interface for editing a Modelica model. The window title is "pendel - pendel - [Modelica Text]". The menu bar includes File, Edit, Simulation, Plot, Animation, Commands, Window, and Help. The toolbar contains various icons for file operations, simulation, and editing. The interface is divided into several panes:

- Packages:** A tree view on the left showing the package hierarchy: Modelica Reference, Modelica, ModelicaAdditions, Unnamed, and pendel (selected).
- Components:** A list in the middle showing the component "pendel" selected.
- Code Editor:** The main area on the right contains the following Modelica code:

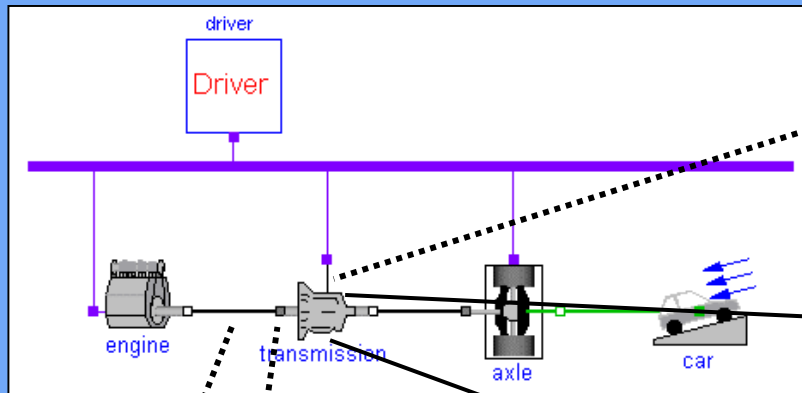

```

model pendel
  parameter Real m=1;
  parameter Real l=0.5;
  parameter Real g=9.81;
  parameter Real d=0.5;
  Real alpha(start=0.1);
  Real w;
equation
  der(alpha) = w;
  m*l*der(w) + m*g*sin(alpha) + d*w = 0;
end pendel;
      
```
- Status Bar:** At the bottom right, it shows "Line: 3", "Modeling" mode, and "Simulation" mode.

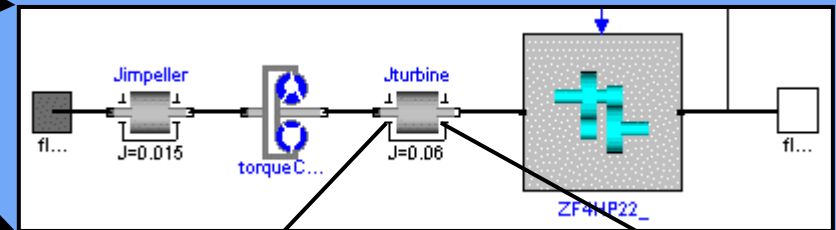
Modellierung

- sog. Objektdiagramme
- jedes Bauteil als abgeschlossene Komponente
- physikalische Konnektoren
- sinnvolle Graphik
- HTML-Dokumentation
- Beispiele:
 - Allgemeines Objektdiagramm
 - Komponente: Rotational Inertia aus der Modelica Library

Objektdiagramme



Komponente



```
equation
w = der(phi);
a = der(w);
J*a = flange_a.tau + flange_b.tau;
```

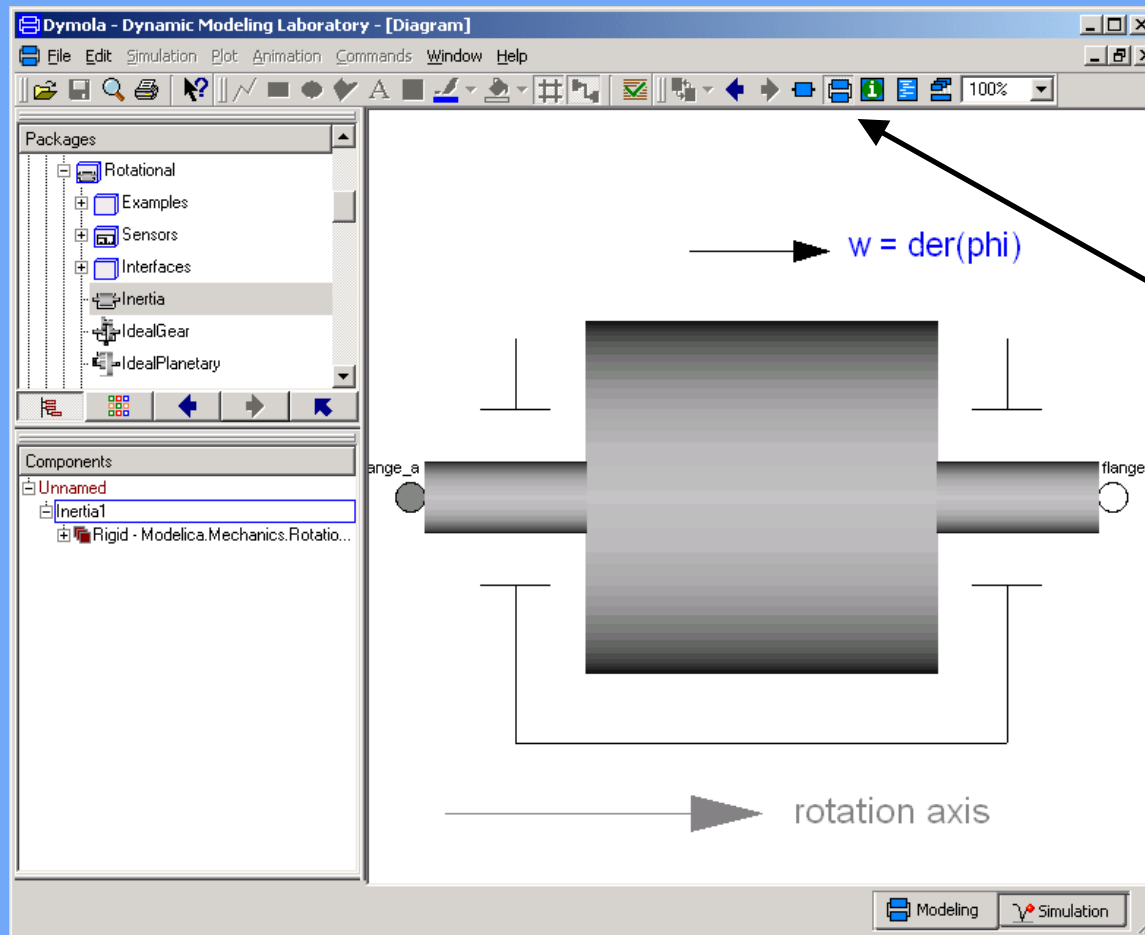
Verbindung

Schnittstelle

Objektdiagramme

- jedes Icon repräsentiert eine physikalische Komponente
- Verbindungen stellen tatsächliche physikalische Verbindungen dar
- jede Komponente wird durch Sub-Komponenten (hierarchisch) oder Gleichungen beschrieben
- Mit **symbolischen** Algorithmen, wird die "high level" Modelica Beschreibung in die Zustandsform transformiert ($\mathbf{dx}/dt = \mathbf{f}(\mathbf{x}, t)$)

Modelica - Rotational - Inertia



Diagram

Analogien für Konnektoren

<u>Fachgebiet</u>	<u>Potential</u>	<u>Fluss</u>
Elektrotechnik	Spannung	Strom
Mechanik, transl.	Position	Kraft
Mechanik, rotat.	Winkel	Drehmoment
Hydraulik	Druck	Volumenstrom
Thermodynamik	Temperatur	Wärmefluß

Grundregeln an Knoten:

Flüsse summieren sich zu Null (Energieerhaltung)

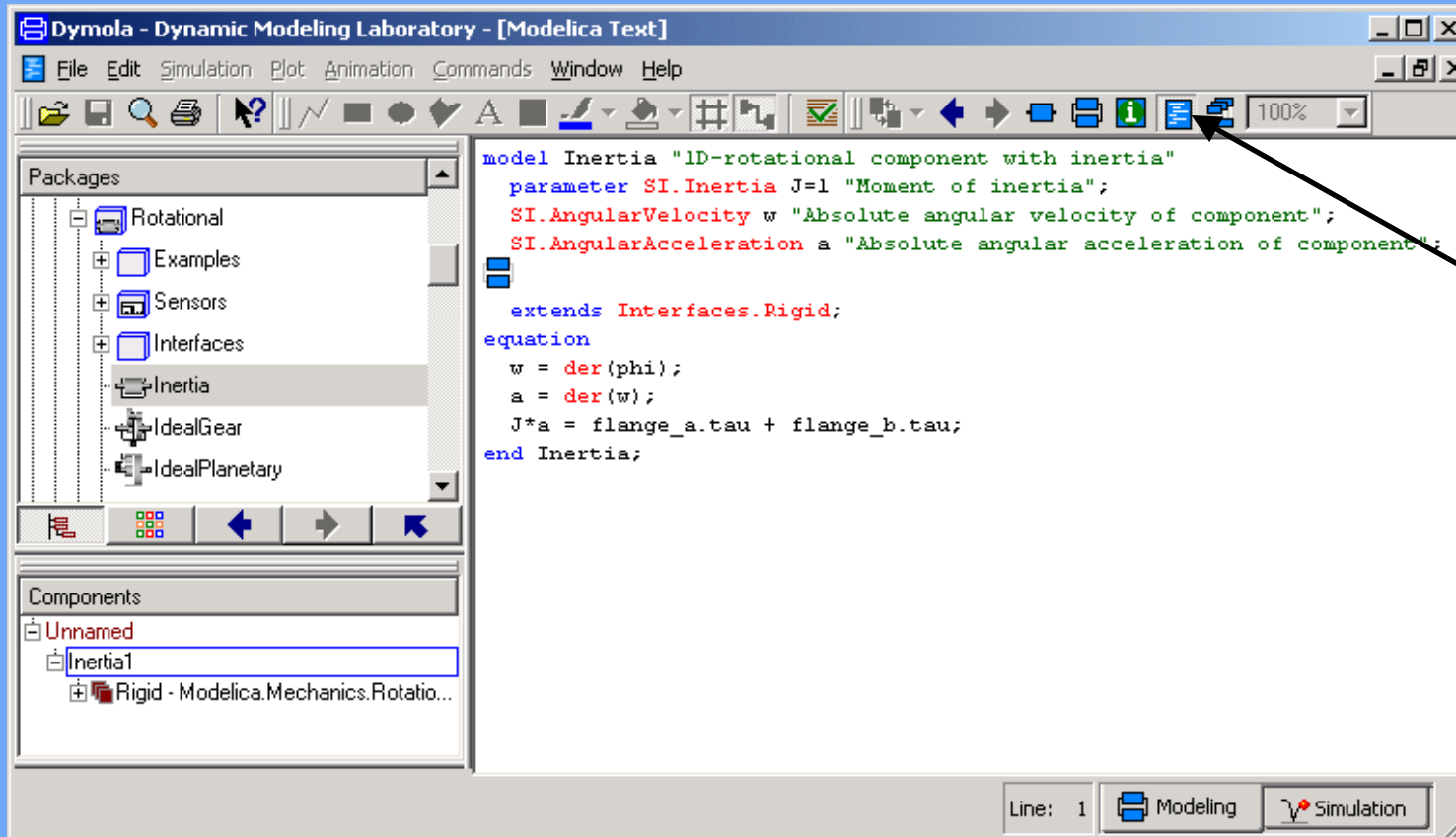
Potentiale haben den gleichen Wert

Konnektoren

```
connector Modelica.Mechanics.Rotational.Interfaces.Flange_a
  "1D rotational flange (filled square icon)"
  SI.Angle phi "Absolute rotation angle of flange";
  flow SI.Torque tau "Cut torque in the flange";
end Flange_a;
```

```
connector Flange_b "1D rotational flange (non-filled square icon)"
  SI.Angle phi "Absolute rotation angle of flange";
  flow SI.Torque tau "Cut torque in the flange";
end Flange_b;
```

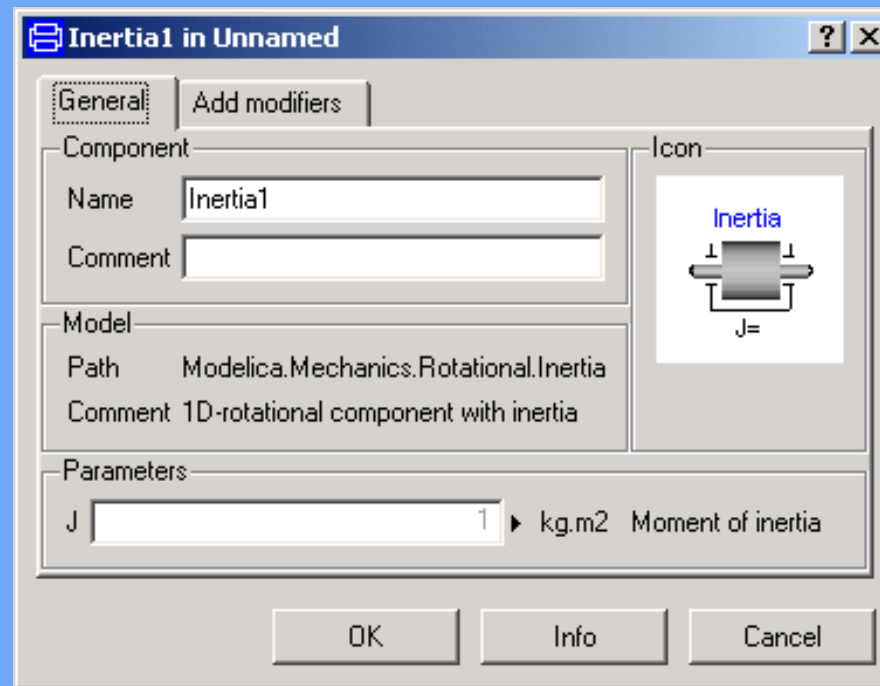
Modelica - Rotational - Inertia



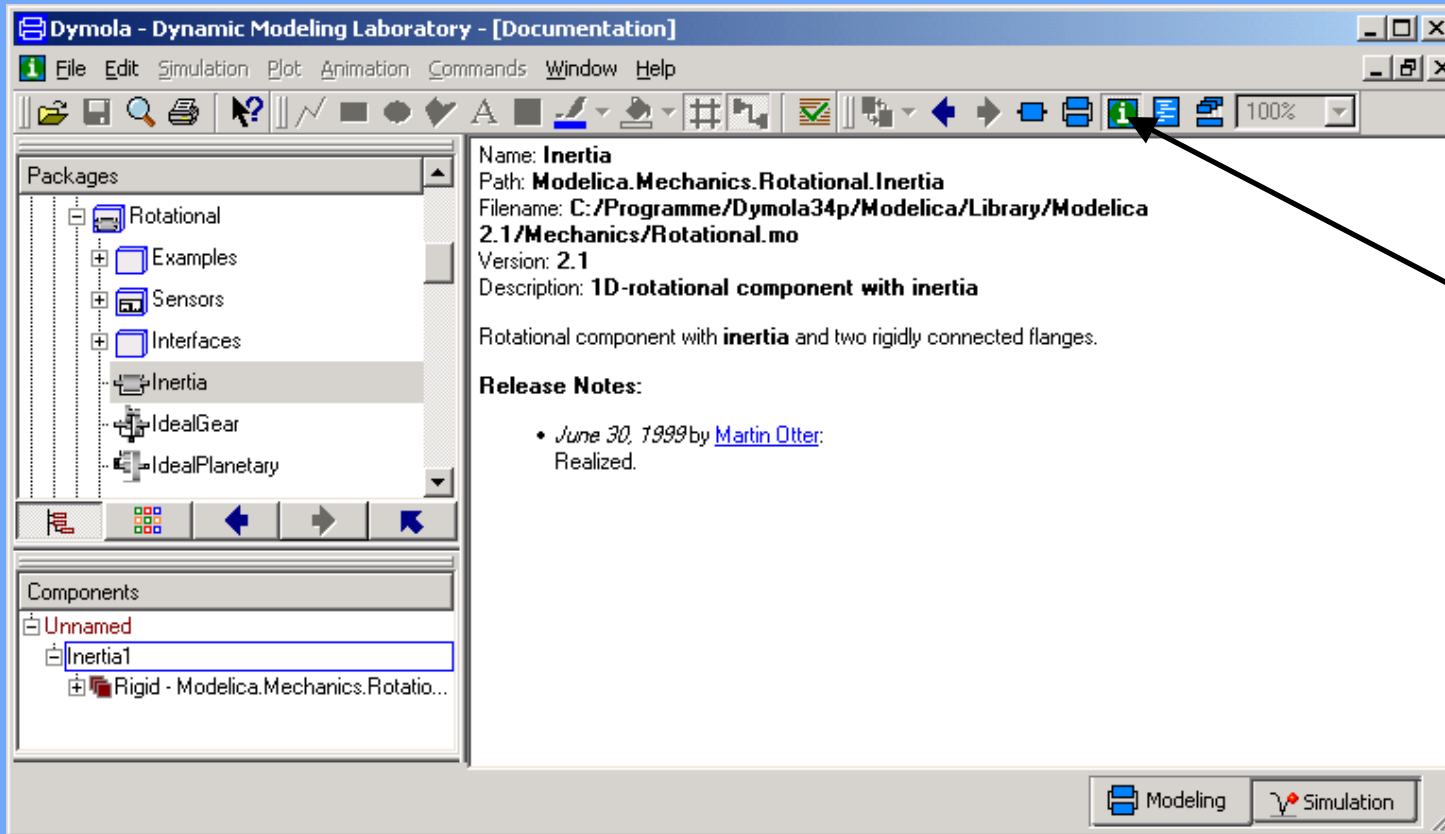
Text

Modelica - Rotational - Inertia

Parameterfenster wird automatisch erstellt



Modelica – Rotational - Inertia



Beschreibung
als HTML

Modelica – Rotational - Inertia

Modelica.Mechanics.Rotational.Inertia

1D-rotational component with inertia

Information

Rotational component with **inertia** and two rigidly connected flanges.

Release Notes:

- *June 30, 1999* by [Martin Otter](#):
Realized.

Parameters

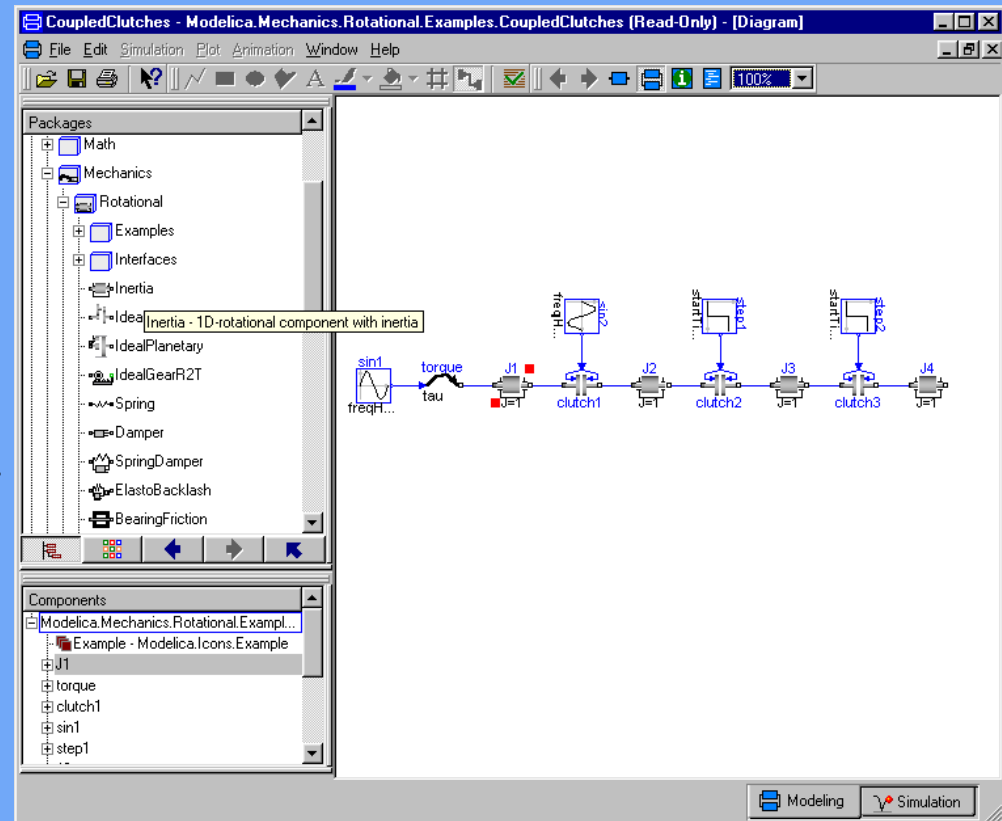
Name	Default	Description
J	1	Moment of inertia [kg.m2]

Modelica definition

automatisch
erstellte
Dokumentation

Modellierung aus Komponenten

- Komponenten in Bibliotheken
- Modell aus Grundkomponenten zusammengesetzt
- Diagramm erscheint im Modellierungsfenster
- Komponenten parametrieren
- Hierarchie für grosse Modelle

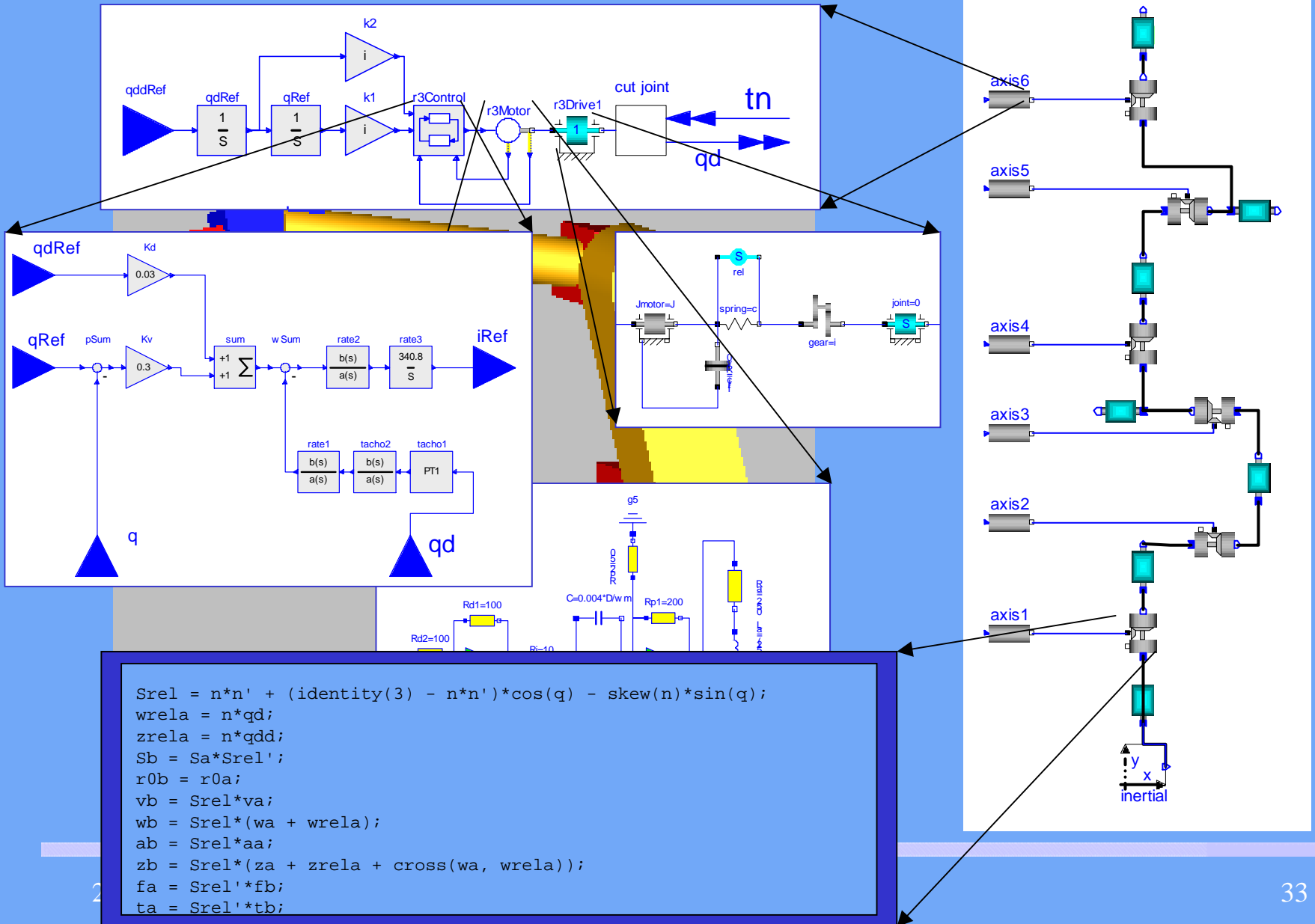


Modellierung von Ereignissen (event)

- z. B. Schalter, Reibung, Schlupf
- Strategie
 - Integration bis zum Ereignis
 - unstetige Änderung durchführen
 - Integration neu starten
 - Werte werden 2x gespeichert, vor und nach dem Ereignis

Modellierung von Ereignissen

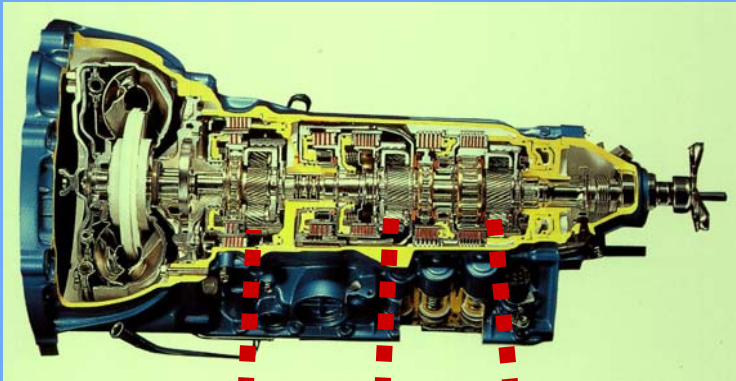
- Durch logische Ausdrücke, z.B.
 $y = \text{if } u > 0 \text{ then } 1 \text{ else } -1$
- Glätten (stetig)
 $y = \text{smooth}(1, \text{if } u > 0 \text{ then } 1 \text{ else } -1)$
- Event-Suche abschalten
 $y = \text{if noEvent}(u > 0) \text{ then } 1 \text{ else } -1$



```

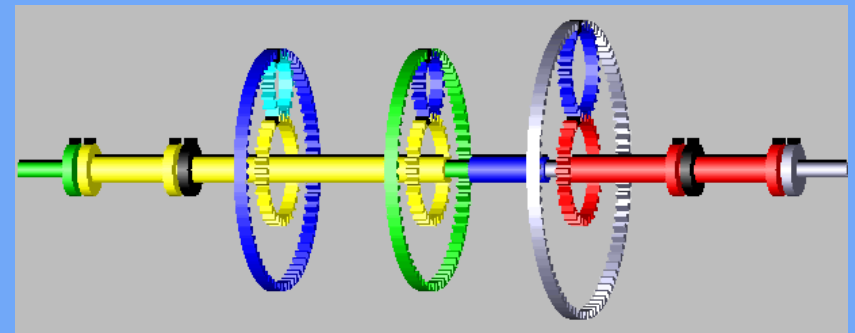
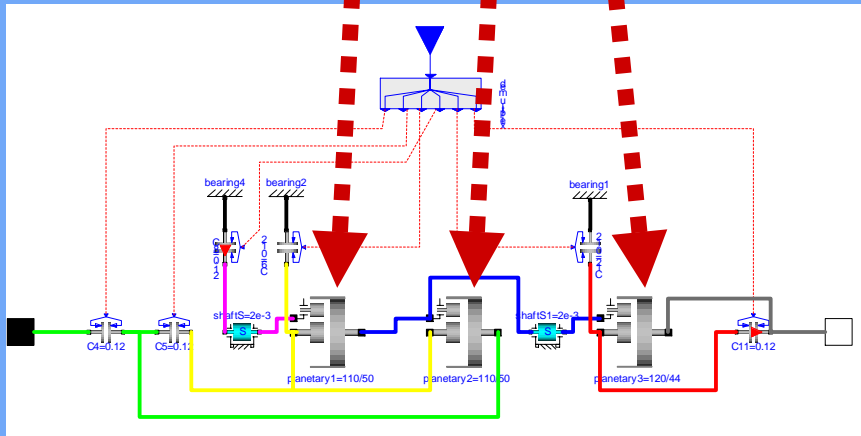
Srel = n*n' + (identity(3) - n*n')*cos(q) - skew(n)*sin(q);
wrela = n*qd;
zrela = n*qdd;
Sb = Sa*Srel';
r0b = r0a;
vb = Srel*va;
wb = Srel*(wa + wrela);
ab = Srel*aa;
zb = Srel*(za + zrela + cross(wa, wrela));
fa = Srel'*fb;
ta = Srel'*tb;
    
```

Getriebe-Modell



Courtesy Toyota Tecno-Service

Powertrain Library



Dymola

- versteht und simuliert Modelica
- Modellierungs- und Simulationsfenster
- übersetzt in C-Code
- Modelldokumentation in HTML, in gleicher Datei
- automatische Extraktion der Modelldokumentation

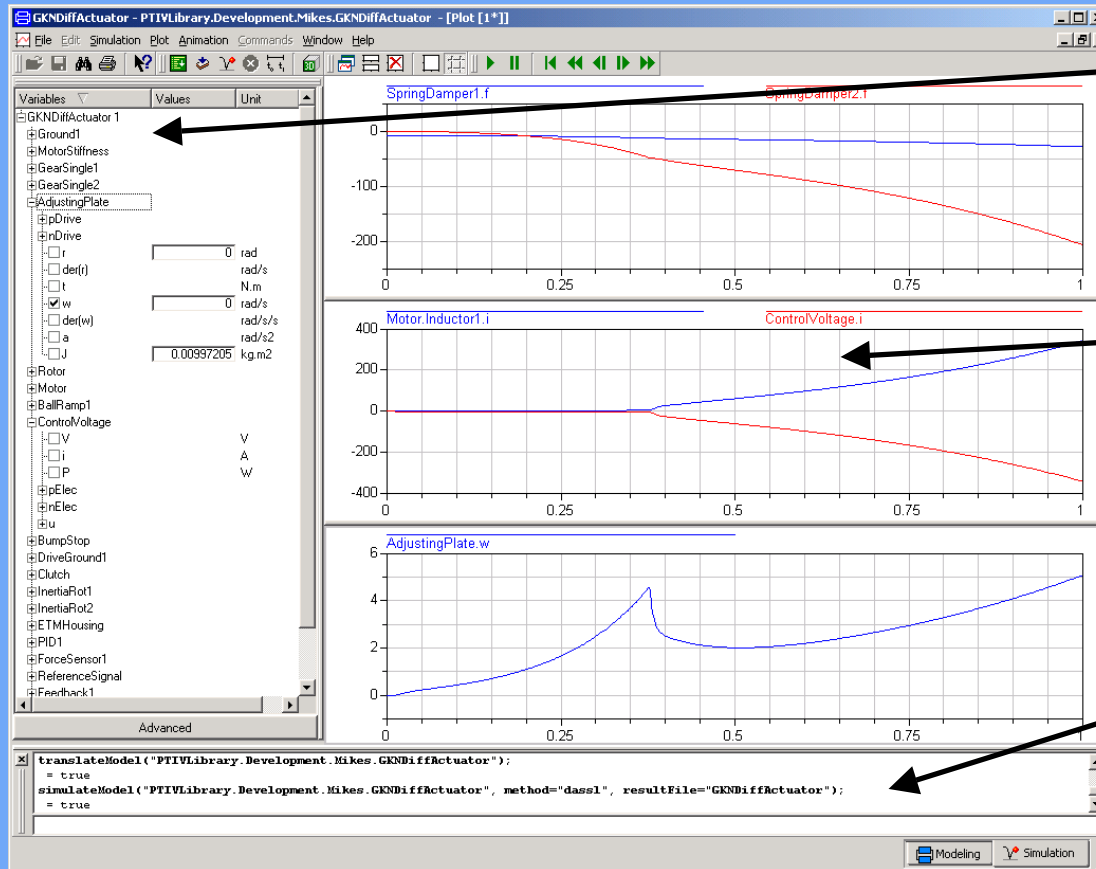
Simulation

- Simulationsfenster
- automatische Initialisierung
- interaktive Studien
- viele effiziente und aktuelle Integrationsverfahren
- Plotmöglichkeit
- Animation

Simulation

- Skripts für vorbereitete Experimente
- Linearisierung der Modelle
- Real-time Hardware-in-the-loop Simulation
- Schnittstelle zu MATLAB zur Datenauswertung
- Modell-Schnittstelle zu Simulink

Simulationsfenster in Dymola



Signal Browser

Alle Parameter und Ergebnisse werden hier angezeigt

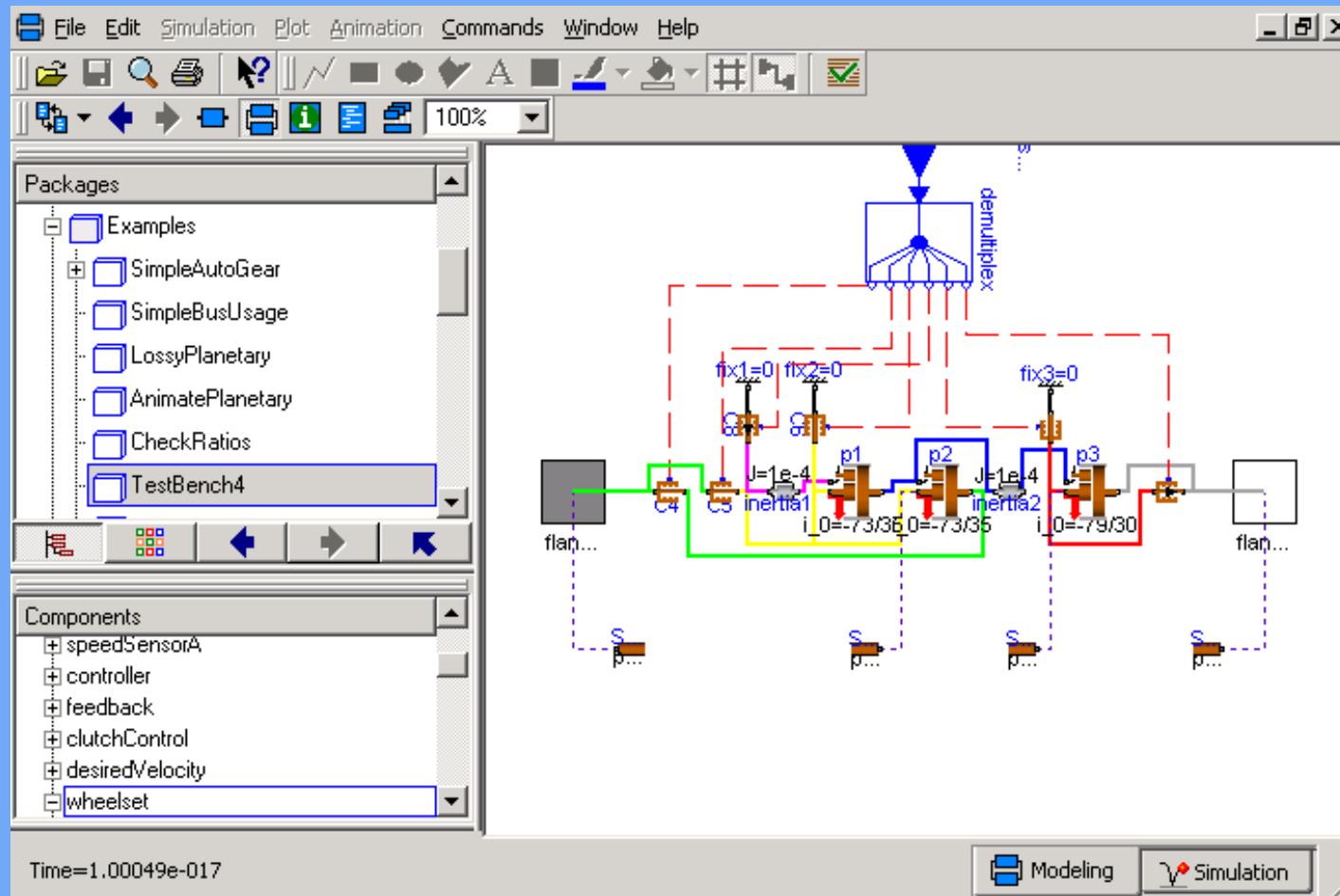
Plots

- mehrere Signale in einem Diagramm
- mehrere Plots in einem Fenster

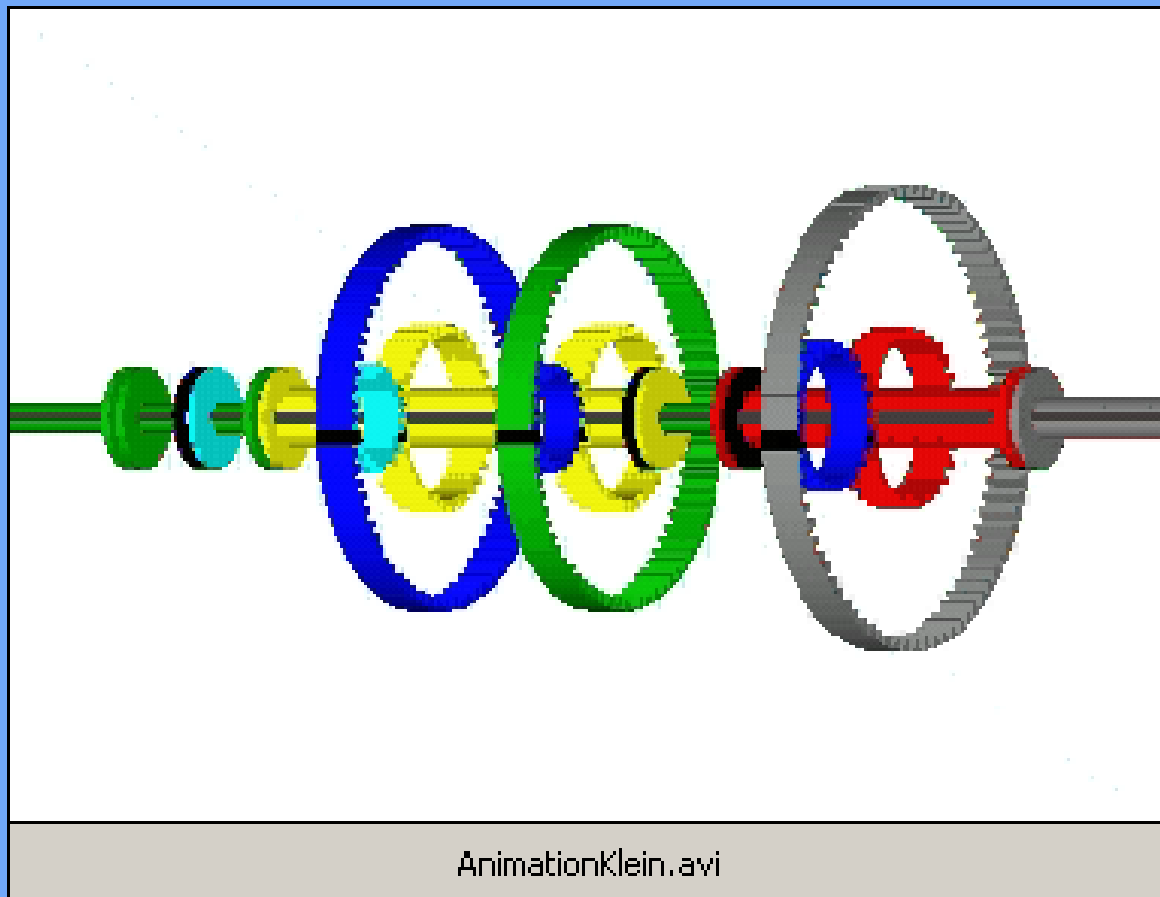
Command window

Befehlssprache und Aufruf von Skripten möglich

Animationsbeispiel: Getriebe



Animation



Dymola Experiment Skriptfile

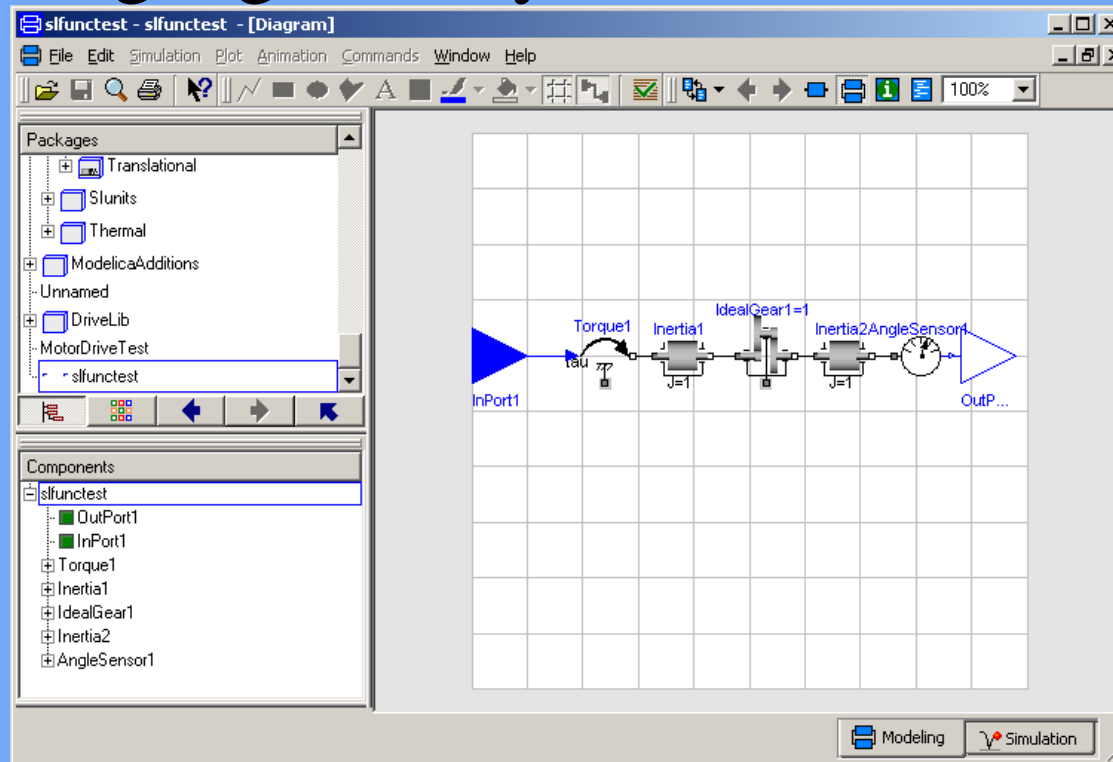
- Ablaufsteuerung
- Parametervariationen
- Plotmöglichkeiten
- Modelica Syntax
- benutzerdefinierte Funktionen

Skript – Beispiel: Parameterstudie

```
openModel("controllerTest.mo");
omega = 1;    // Declare omega.
k = 1;       // Declare gain.
for D in {0.1, 0.2, 0.4, 0.7} loop
    // Parameter sweep over damping coefficient.
    tr.a = {1, 2*D*omega, omega**2};
    tr.b = {k*omega**2};
    simulateModel("controllerTest", 0, 10);
    plot({"u", "y"});
end for;
```

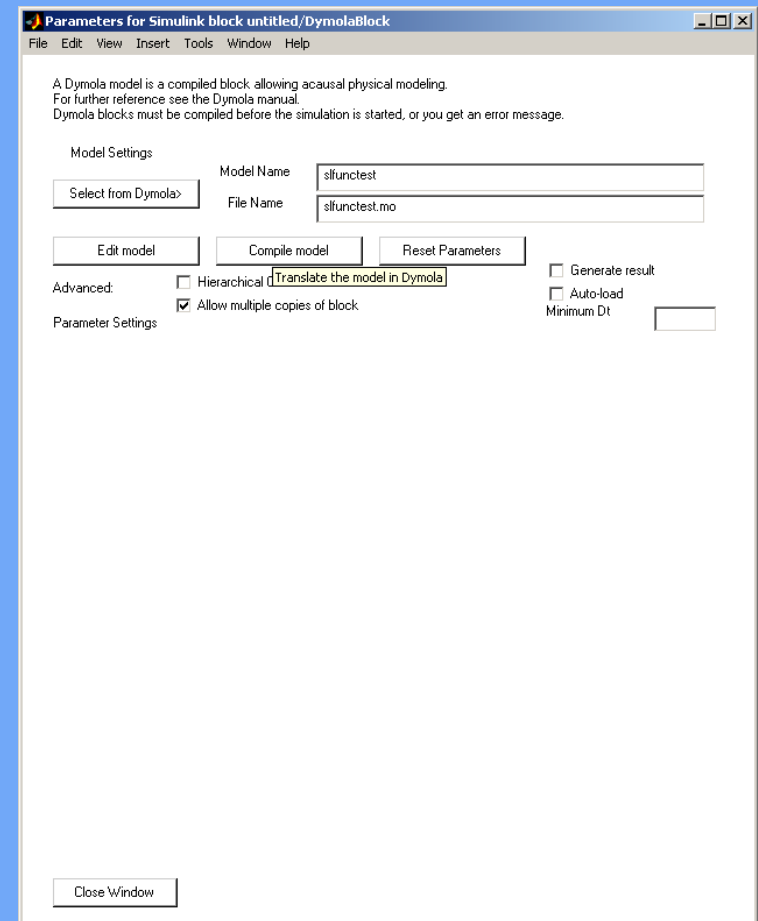
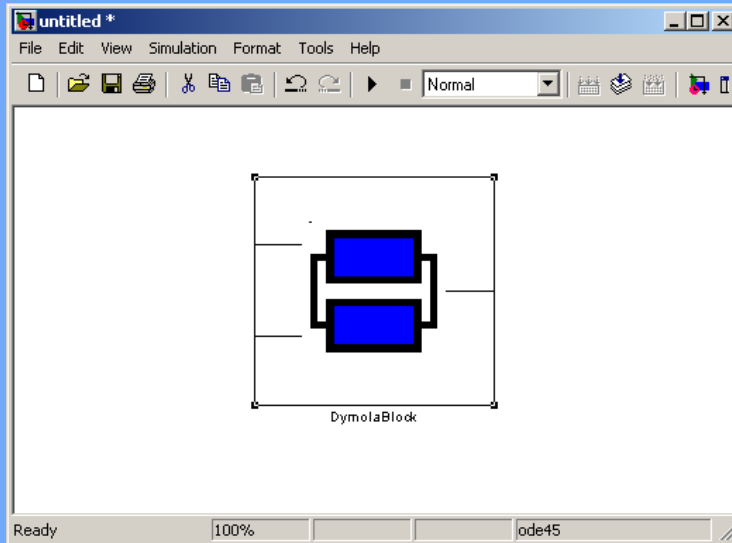
Dymola-Simulink-Interface

Ein-/Ausgänge im Dymola-Modell festlegen



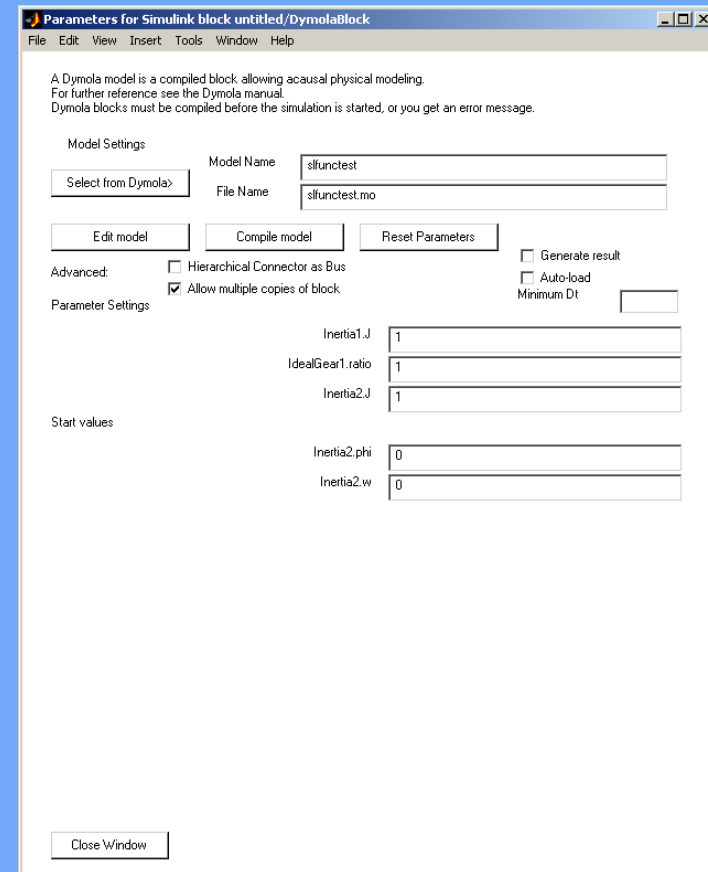
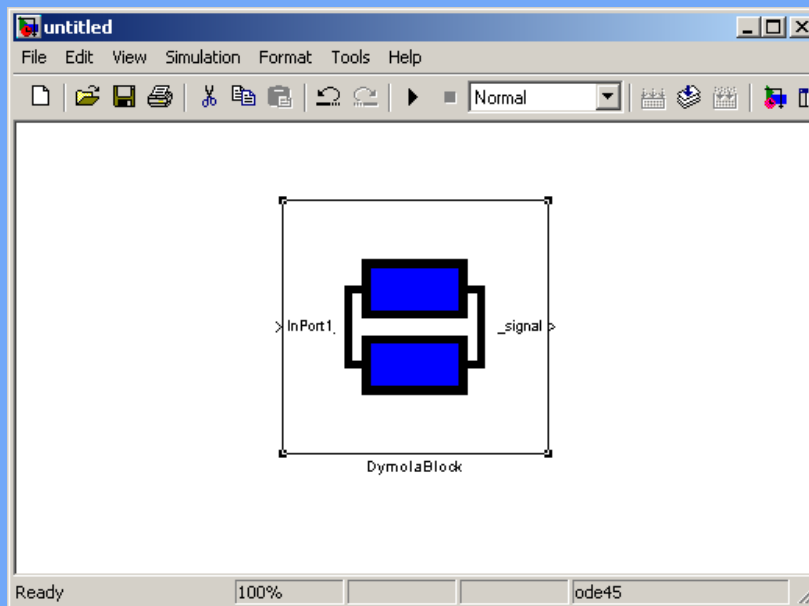
Dymola-Simulink-Interface

Dymola-Block in
Simulink-Modell einfügen
Modell übersetzen



Dymola-Simulink-Interface

ergibt Block mit Ein- und Ausgängen



Einige Neuerungen in Modelica 2.1

- Felder und deren Indizierung
- Konnektoren verbessert
 - Verbindung hierarchischer Konnektoren
 - überbestimmte Konnektoren
- Break- und Return-Anweisungen
- Builtin-Funktionen: String, Integer, Semilinear
- Neue Annotationen
- und mehr, sh.

www.modelica.org/news_items/modelica_2_1

Weiterentwicklung: Version 5.3 nach 5.4

Beispiele (sh. Dymola Release Notes 5.3++++)

- bessere automatische Initialisierung
- Modelica 2.1 mit automatischer Modellkonversion
- bessere Tabelleneditoren

Zusammenfassung

- Modelica soll (de-facto) **Standard** werden
- Dymola unterstützt Modelica voll
- viele Experten arbeiten an Modelica mit
- freie Bibliotheken stehen auf der Modelica-Homepage zur Verfügung
- einige werden mit Dymola geliefert
- zusätzliche Bibliotheken können erworben werden
- Homepages:
www.dynasim.se
www.modelica.org