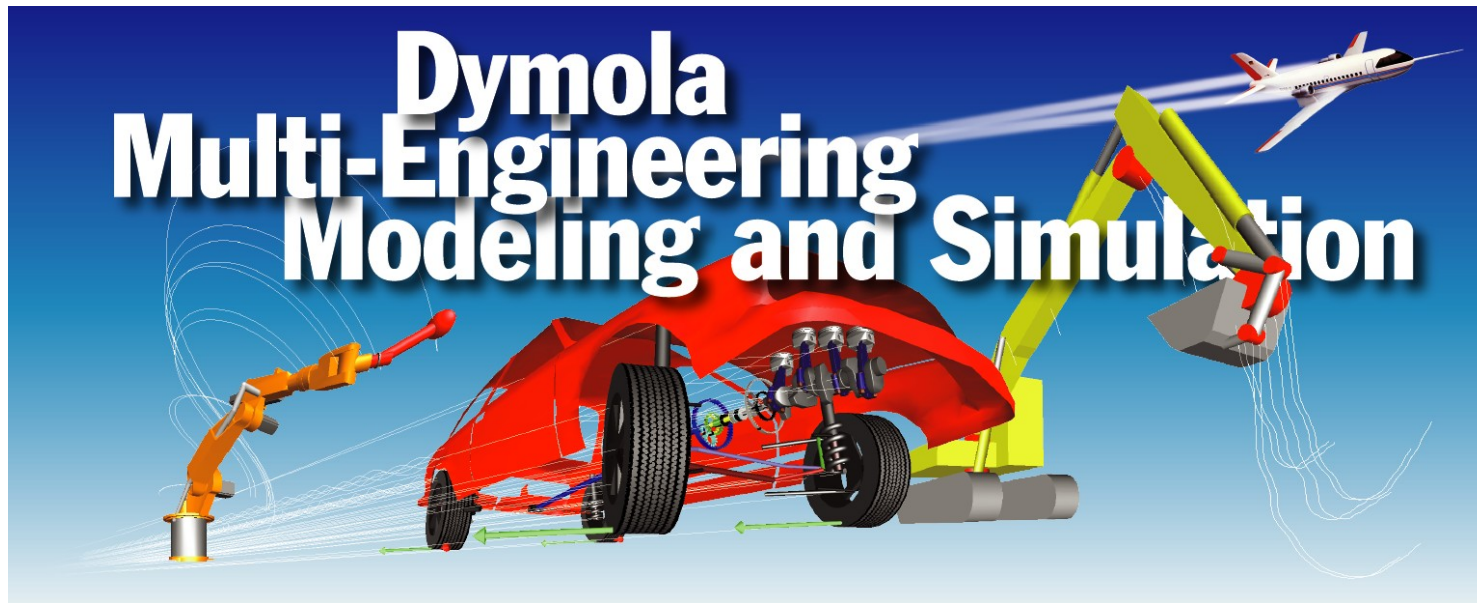


Überblick zu Dymola und Modelica



Inhalt

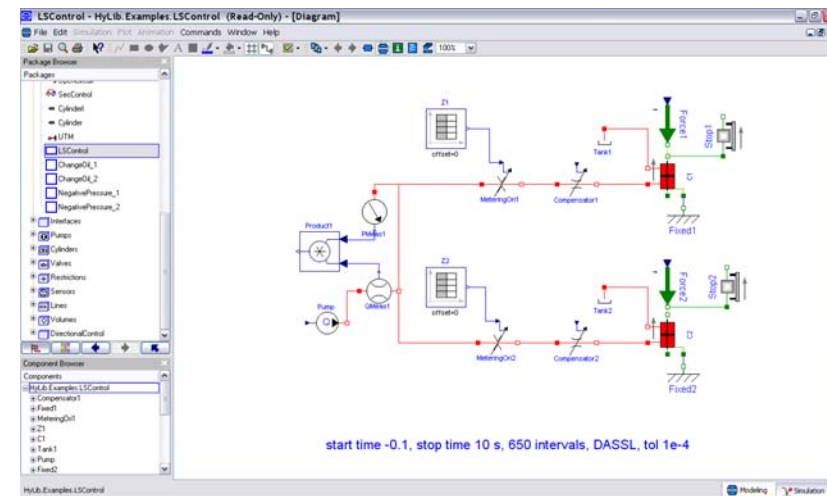
- Dymola
- Modelica
- Modellieren mit Dymola
- Modellbibliotheken
- Simulieren mit Dymola
- Dymola-Simulink-Interface
- Neuigkeiten in Dymola und Modelica
- Ergänzungen zu Dymola
- Kommerzielle Bibliotheken

BAUSCH-GALL GmbH

- Hans Gall und Dr. Ingrid Bausch-Gall seit 1981 selbständig
- Seit 1987 GmbH
- Vertrieb von Simulationssoftware
- Beratung im Simulationsumfeld
- Projektarbeit
- Schulungen
 - Dymola, Spice, Simulink, Matlab

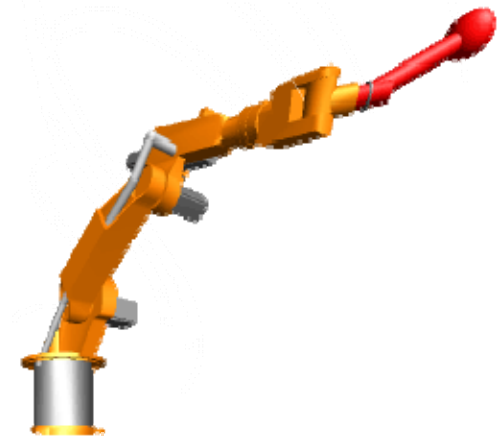
Dymola

- Dymola ist eine Software zur Modellierung und Simulation aus verschiedenen Ingenieurdisziplinen
- entwickelt von Dynasim AB in Schweden
- Vertrieb im deutschsprachigen Raum: Bausch-Gall GmbH
- Dymola versteht den offenen Sprachstandard Modelica



Die Firma Dynasim

- Entwickler von Dymola
- gegründet 1992 von Dr. Hilding Elmqvist
- Seit Juni 2006: Eigentümer Dassault Systèmes
- Zusammenarbeit mit Wissenschaftlern
- Einige Kunden
 - weltweit, z.B.: Toyota, Volvo, Ford, GM, GDF
 - Deutschland, z.B.: ABB, Airbus, Audi, BASF, Behr, BMW, Robert Bosch GmbH, Daimler, Hella, LfK, Siemens, ZF, Volkswagen, Universitäten und Forschungsinstitute
 - Österreich, z.B.: AVL, Elin, MAGNA STEYR, Arsenal Research, TU Wien



Ziel von Modelica und Dymola

Modellierung und Simulation physikalischer Systeme die aus Komponenten unterschiedlicher Fachgebiete bestehen. Z.B. ein detailliertes Fahrzeugmodell:

- Fahrzeugdynamik – 3-D Mechanik
- Antriebsstrang – 1-D Mechanik
- Hydraulik
- Verbrennung
- Klimaanlage
- Elektrische/elektronische Systeme, Steuergeräte, Busse

Modelica - Sprachstandard

- neue Methode
 - objektorientiertes Modellieren
 - gleichungsorientiertes Modellieren
- austauschbare, standardisierte formale Sprache
 - zum Modellaustausch
 - Wiederverwendung von Modellierungswissen
- Modelica: Sprachdefinition, aktuell Version 3.0
www.modelica.org
- Dymola: Software zur Modellierung und Simulation mit Modelica
www.dynasim.com

Modelica Design Group

- Zusammenschluß von Simulationsexperten seit 1997
- Ehrenamtlich, mehrere Treffen jährlich
- Erstellung und Veröffentlichung der Sprachdefinition
- Pflege der Homepage: www.modelica.org
- Homepage enthält
 - viele Veröffentlichungen
 - freie Libraries
- Letzte Tagung: September 2006 in Wien
- Nächste Tagung: 3.-4. März 2008 in Bielefeld

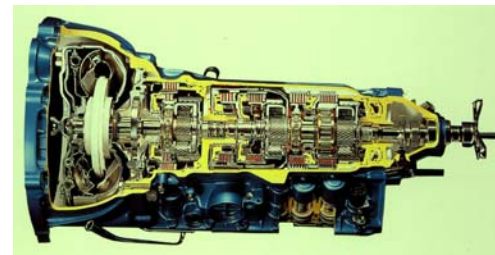
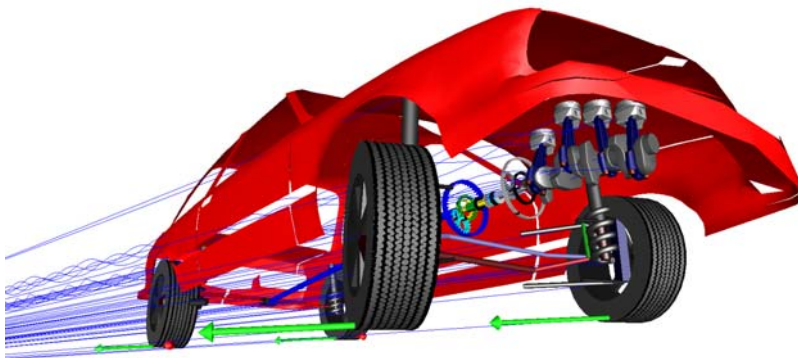
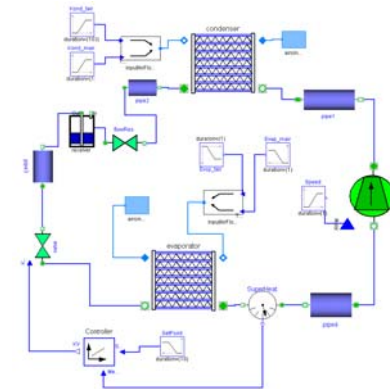
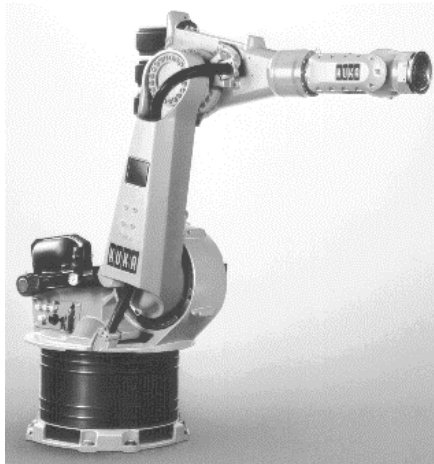
Ziele der Modelica Association

- offene Sprachdefinition
- objektorientierte Sprachen vereinheitlichen
- Wiederverwendbarkeit der Modelle fördern
- deklarative, d.h. mathematische Gleichungen statt prozedurale Zuweisungen
- Sprache soll effiziente Simulation unterstützen
- zur Zeit Version 3.0 (in Dymola noch 2.2.1)
- freie Bibliotheken auf der Modelica-Homepage

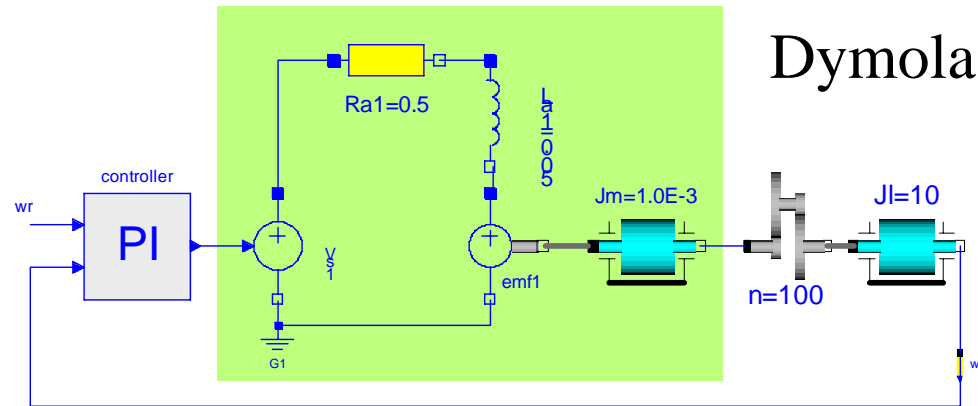
Dymola-Entwicklungsziele

- Software zur Modellierung komplexer Systeme
- versteht Modelica fast vollständig
- Modelle verschiedener Ingenieurdisziplinen
- Verfügbarkeit der Modelle als Quellcode
- homogenes Simulationsmodell
- effiziente Simulation
- Verkürzung der Entwicklungszeit
- Schnittstellen zu Standardsoftware

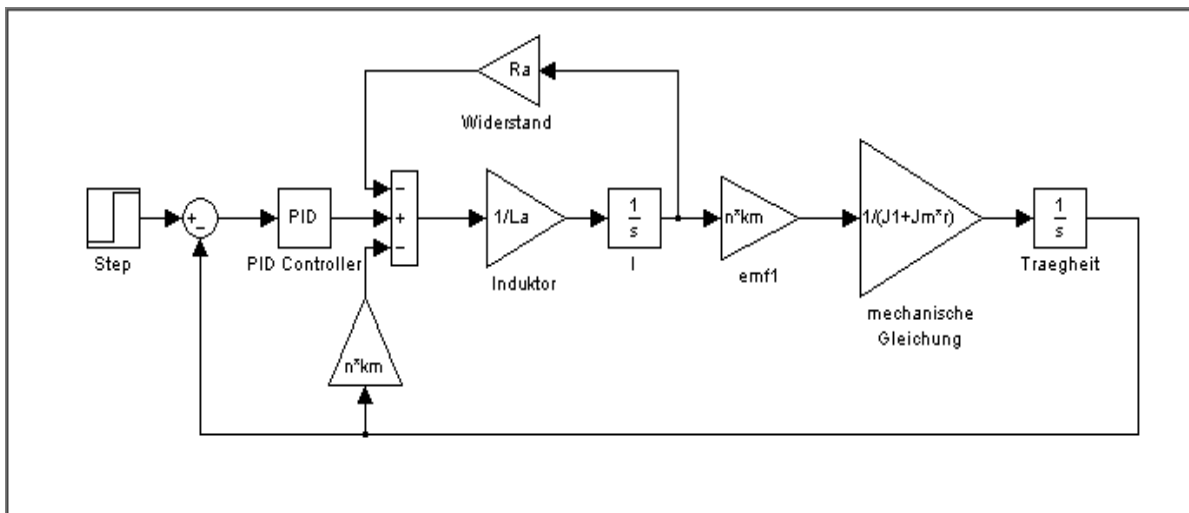
Anwendungsbeispiele



Energieflußorientierte Modellierung



Dymola



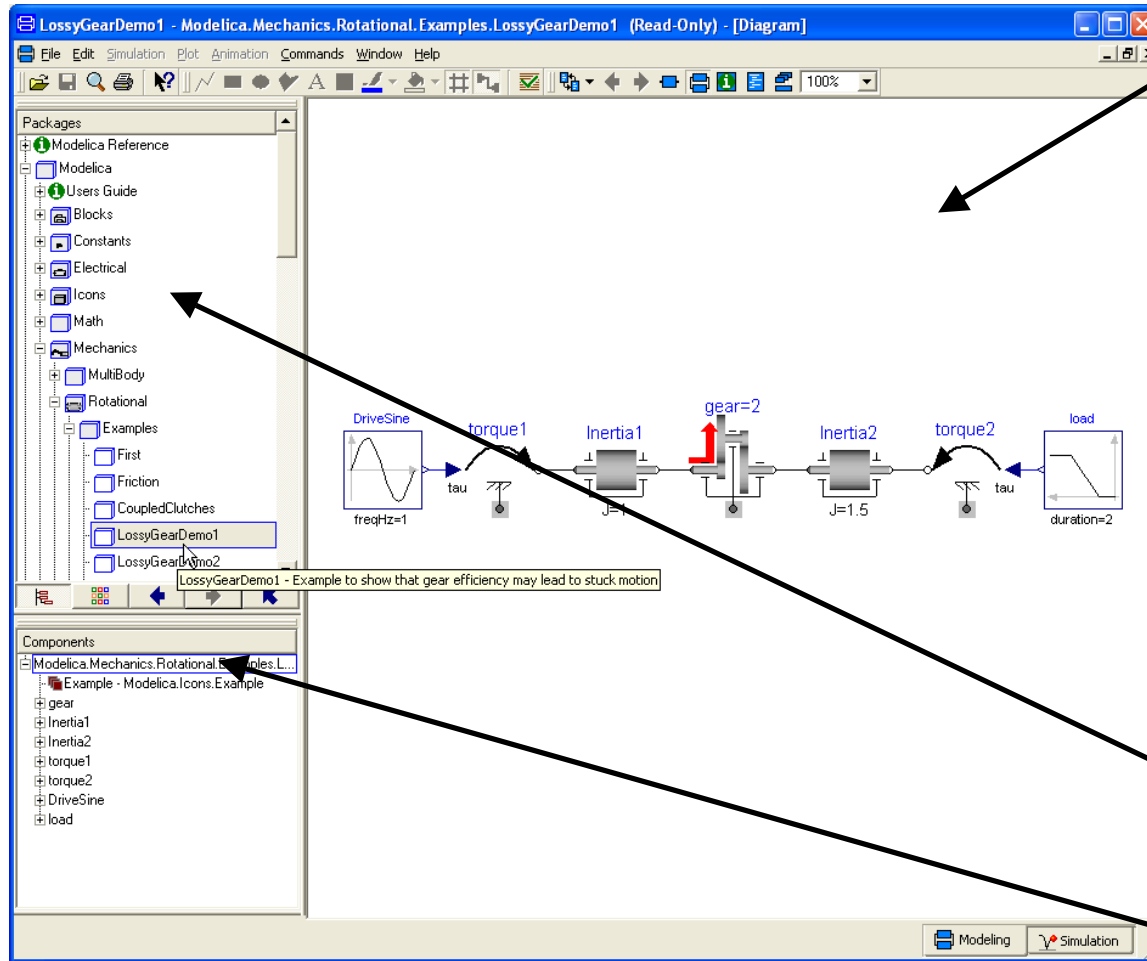
signalflußorientierte Modellierung

Modellierung mit Dymola

Modellierung in Modelica bedeutet:

- Gleichungen als Modellbasis
- übliche Datenstrukturen und Programmierkonzepte
- Felder und Matrizen, Operatoren
- objektorientierte Struktur der Modelle
Klassen, Konnektoren, Blöcke
- Einheiten (units), ab Dymola 7 auch Einheitenprüfung
- Funktionen und Algorithmen
- symbolische Manipulation der Gleichungen
- Modellbibliotheken

Dymola-Modellierungsfenster



Modellierungsfenster

Icon Layer: Definition der Graphik

Diagram Layer: Zusammenstellen des Modells

Documentation Layer

HTML-Dokumentation einfügen

Equation Layer:

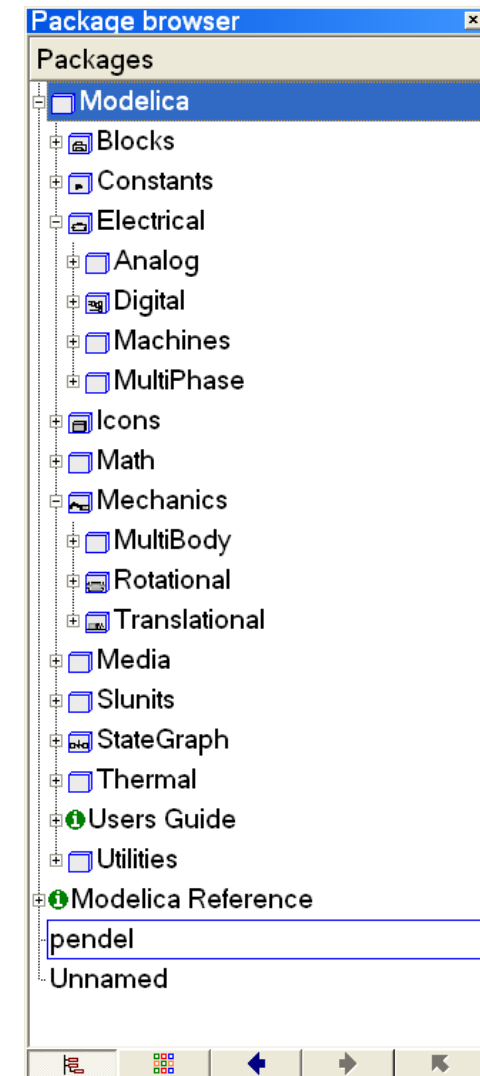
Gleichungen als Modelica-Kode

Model Library Browser

Model Browser

Modelica-Bibliotheken

- hierarchisch
- Komponenten lassen sich sinnvoll anordnen
- grosse Komponentenvielfalt
- offene Modelica-Bibliotheken enthalten z.B.:
 - einfache Regelungstechnik (Blöcke)
 - 1D-Mechanik (rotatorisch, translatorisch)
 - Mehrkörpersysteme
 - analoge und digitale Elektrik
 - elektrische Maschinen
 - einfache Pneumatik und Hydraulik
 - Zustandsautomaten (State Graph)

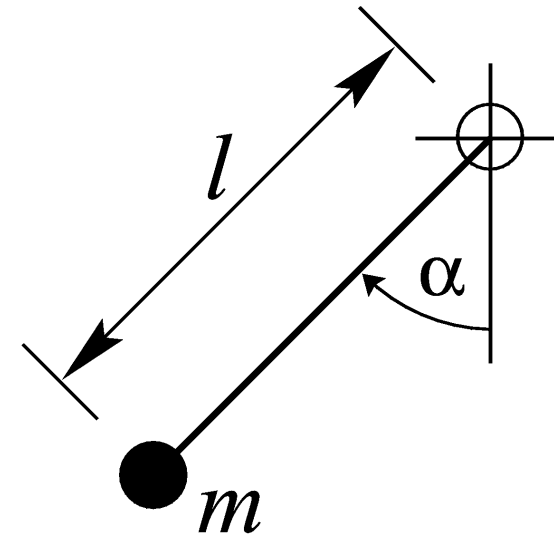


Modelica-Bibliotheken

- kommerzielle Bibliotheken für
 - Antriebsstränge (PowerTrain, DLR)
 - SmartElectricDrives (arsenal research)
- von der Firma Modelon (www.modelon.se)
 - Pneumatik
 - Hydraulik
 - Klimaanlage (AirConditioning)
 - Fahrdynamik (VehicleDynamics)
- weitere Bibliotheken sind in der Entwicklung
- viele freie Bibliotheken
www.modelica.org

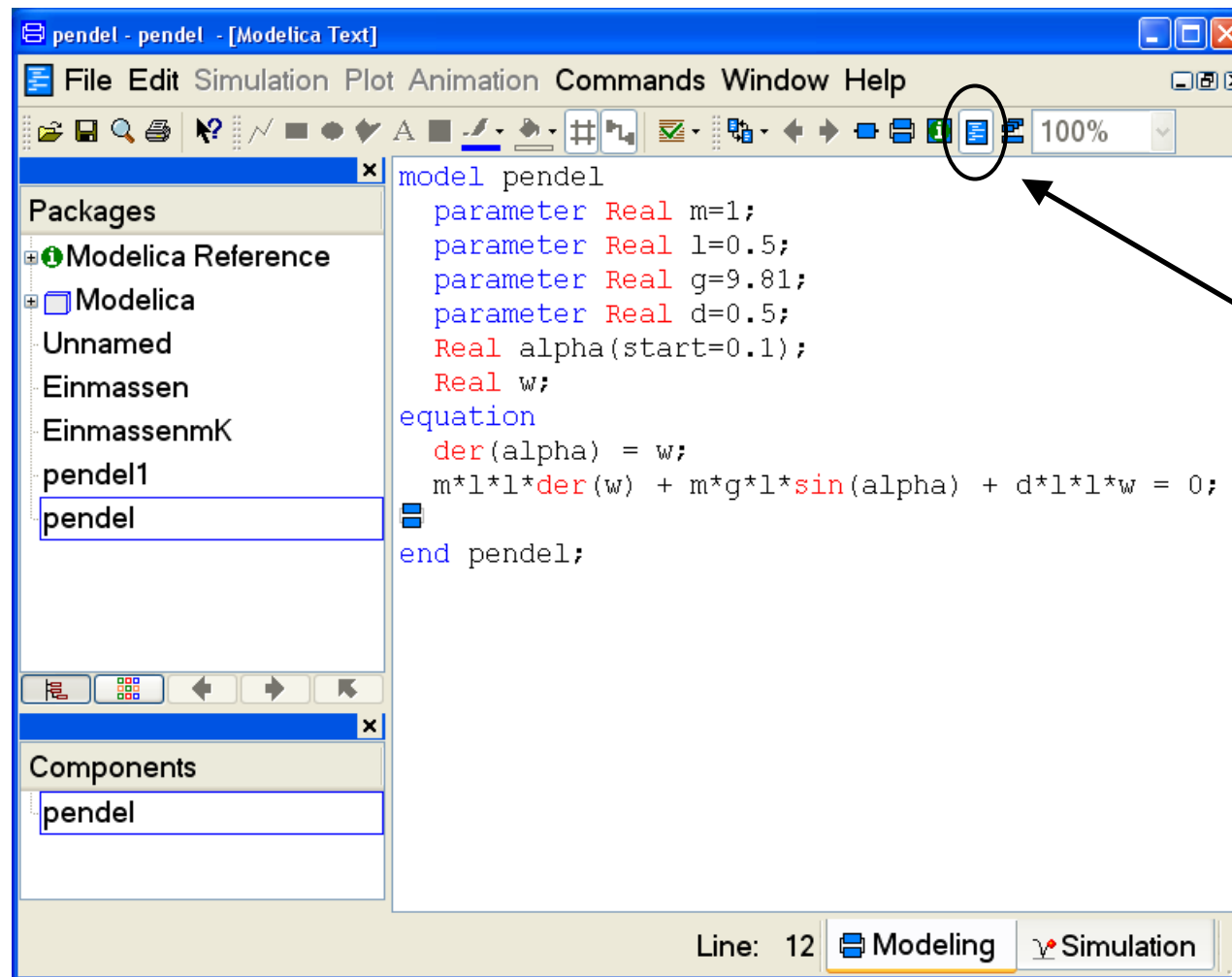
Grundlage der Modellbildung in Dymola

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



$$m l^2 \ddot{\alpha} + m g l \sin(\alpha) + d l^2 \dot{\alpha} = 0$$

Dymola-Modell

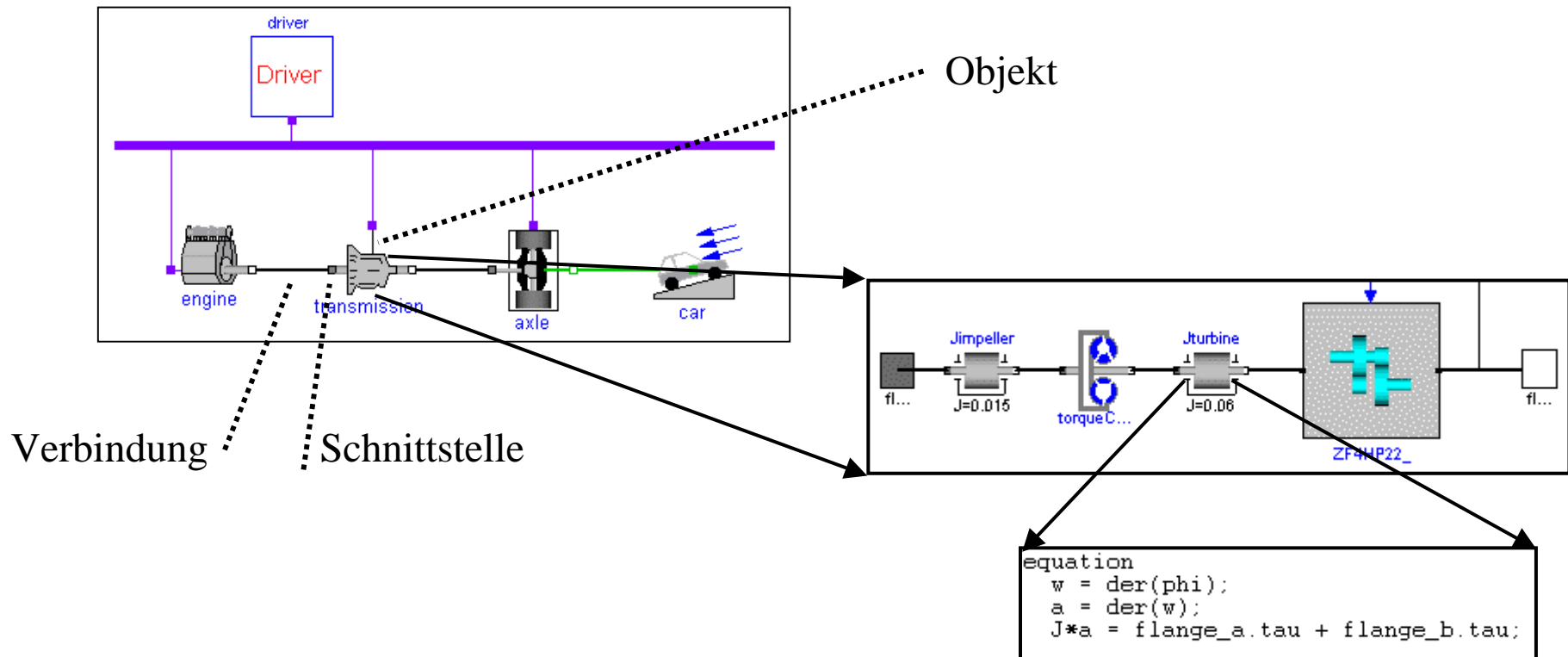


Textlayer

Modellierung -Zusammenfassung

- als sog. Objektdiagramme
- physikalische Konnektoren
- sinnvolle Graphik
- HTML-Dokumentation
- Beispiele:
 - allgemeines Objektdiagramm
 - Objekt: Rotational Spring aus der Modelica Library

Objektdiagramme



Objektdiagramme

- jedes Icon repräsentiert ein physikalisches oder logisches Objekt
- Verbindungen stellen tatsächliche physikalische Verbindungen dar
- jedes Objekt wird hierarchisch erstellt oder durch Gleichungen beschrieben
- mit **symbolischen** Algorithmen, wird die Modelica Beschreibung in die Zustandsform transformiert

$$\frac{dx}{dt} = f(t, x, u)$$

$$y = g(t, x, u)$$

Modelica - Rotational - Inertia

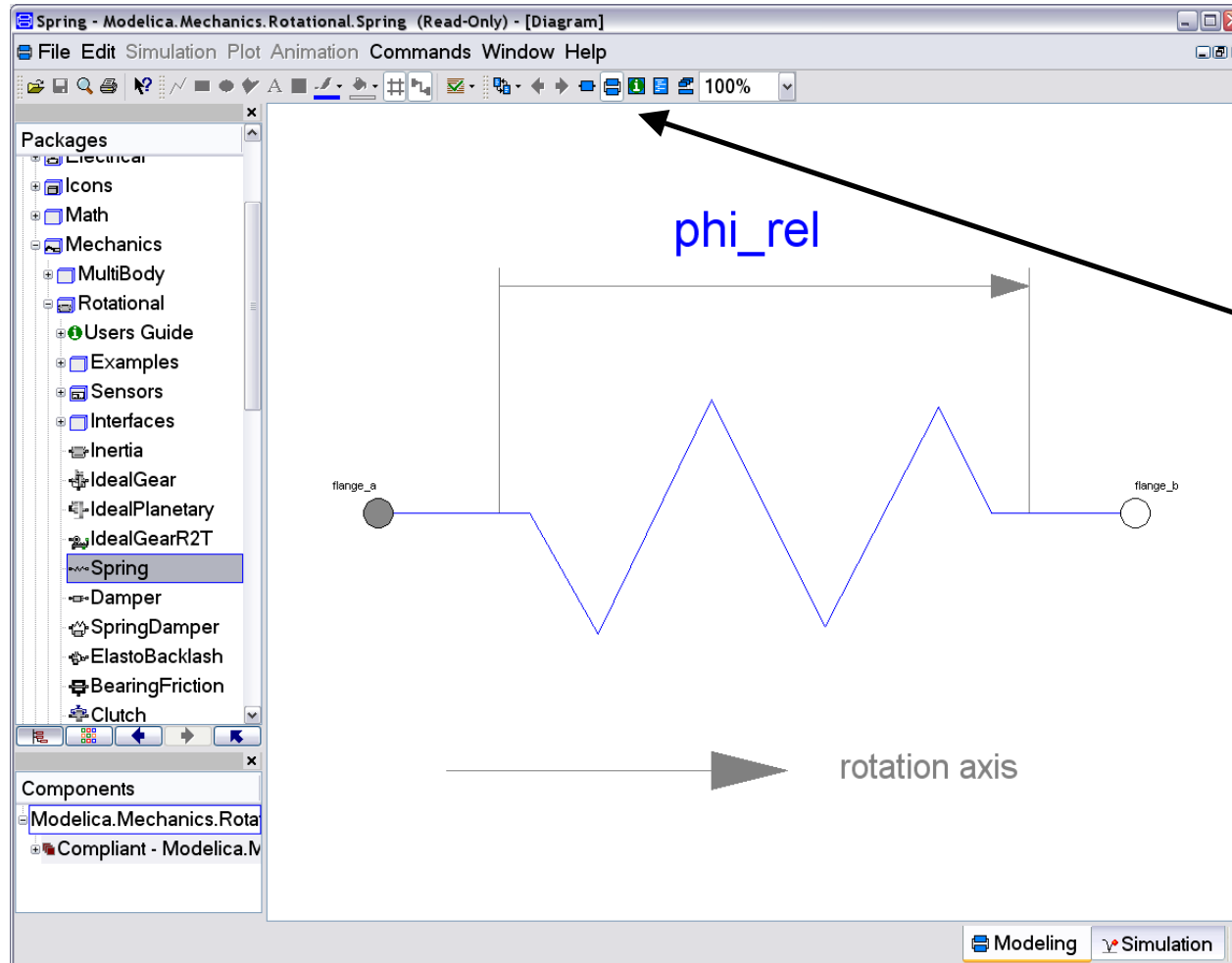
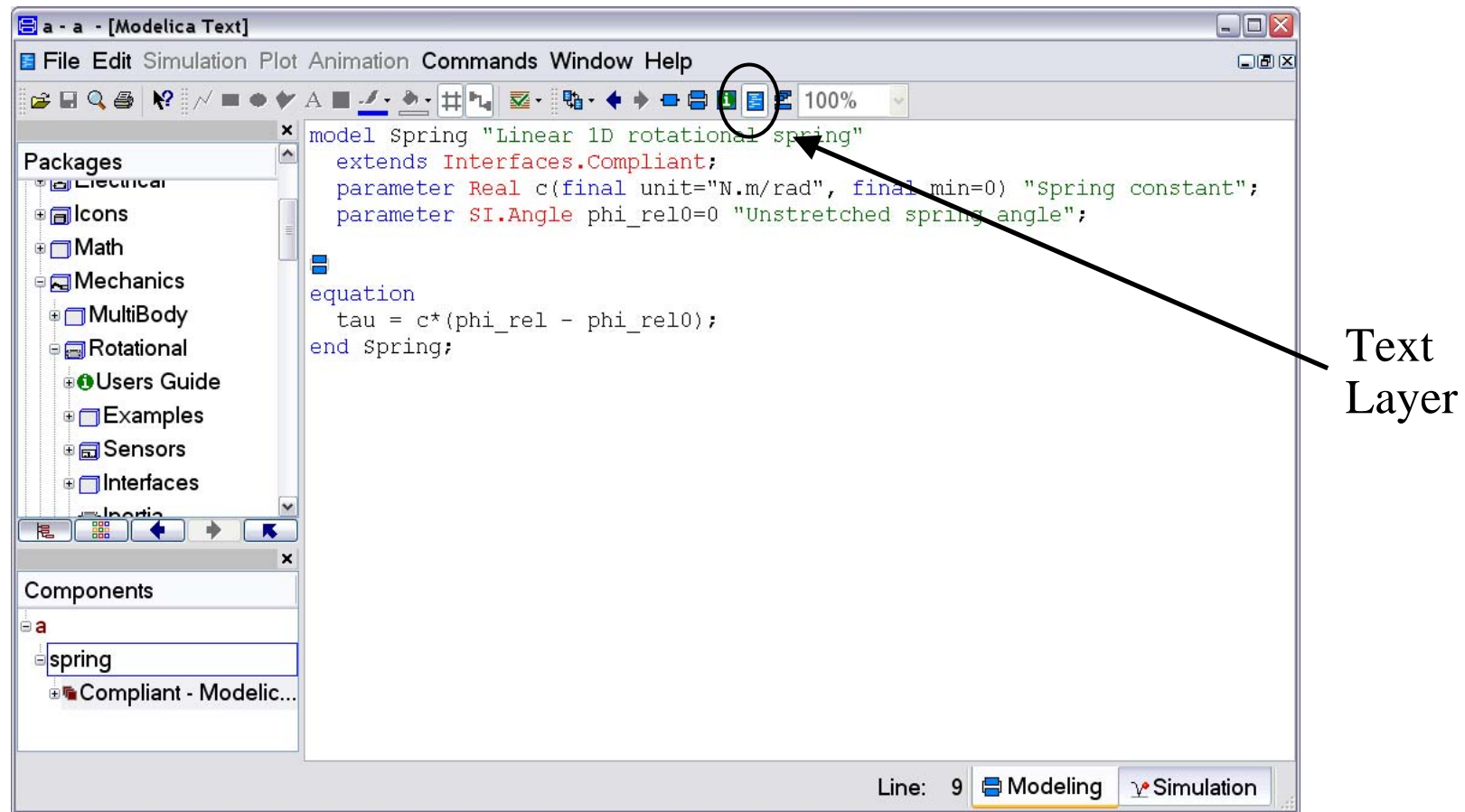


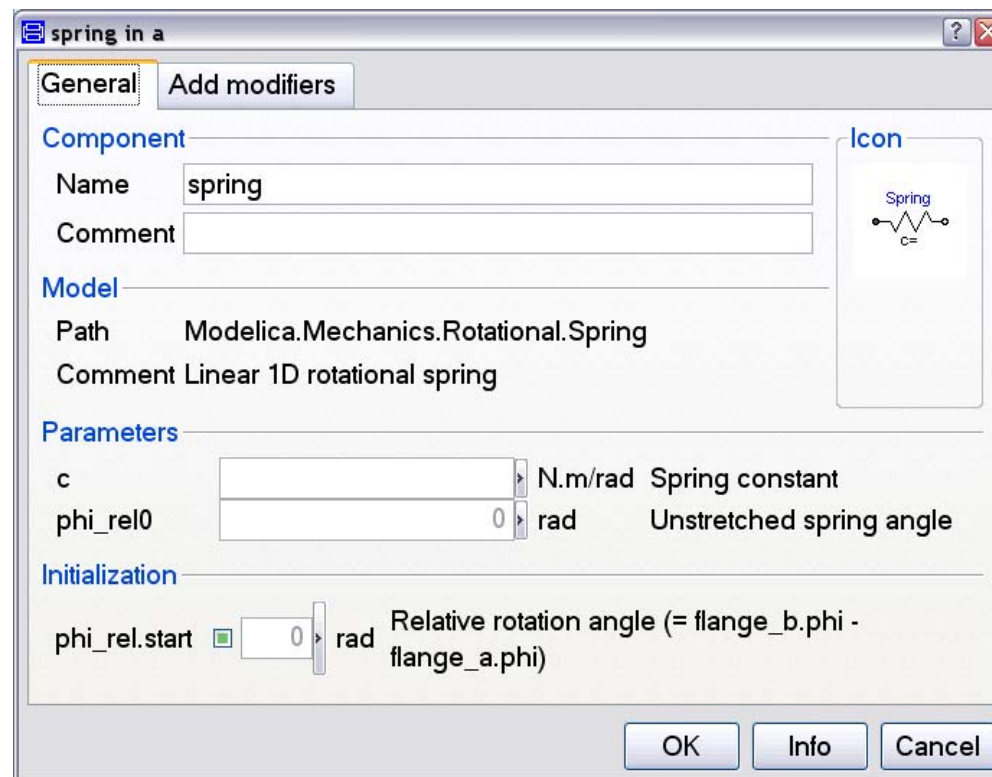
Diagram-Layer

Modelica – Rotational - Inertia



Modelica - Rotational - Inertia

Parameterfenster wird automatisch erstellt



Modelica – Rotational - Inertia

Linear 1D rotational spring

Information

A **linear 1D rotational spring**. The component can be connected either between two inertias/gears to describe the shaft elasticity, or between a inertia/gear and the housing (component Fixed), to describe a coupling of the element with the housing via a spring.

Parameters

Type	Name	Default	Description
Real	c		Spring constant [N.m/rad]
Angle	phi_rel0	0	Unstretched spring angle [rad]

Initialization

Angle	phi_rel		Relative rotation angle (= flange_b.phi - flange_a.phi) [rad]
-------	---------	--	---

Connectors

Type	Name	Description
Flange a	flange_a	(left) driving flange (flange axis directed INTO cut plane)
Flange b	flange_b	(right) driven flange (flange axis directed OUT OF cut plane)

Name: Spring
 Path: Modelica.Mechanics.Rotational.Spring
 Filename: C:/Programme/Dymola60b/Modelica/Library/Modelica 2.2.1/Mechanics/Rotational.mo
 Version: 2.2.1

Beschreibung
als HTML

Modelica – Dokumentation

Modelica Standard Library

Information

Package **Modelica** is a **standardized** and **free** package that is developed together with the Modelica language from the Modelica Association, see <http://www.Modelica.org>. It is also called **Modelica Standard Library**. It provides model components in many domains that are based on standardized interface definitions. Some typical examples are shown in the next figure:

For an introduction, have especially a look at:

- [Users Guide](#) discusses some aspects of the Modelica Standard Library, such as interface definitions and used conventions.
- [Release Notes](#) summarizes the changes of new versions of this package.
- Packages **Examples** in the various subpackages, demonstrate how to use the components of the corresponding sublibrary.

Copyright © 1998-2006, Modelica Association.

*This Modelica package is **free** software; it can be redistributed and/or modified under the terms of the **Modelica license**, see the license conditions and the accompanying **disclaimer** [here](#).*

Package Content

Name	Description
UsersGuide	Users Guide
Blocks	Library for basic input/output control blocks (continuous, discrete, logical, table blocks)
Constants	Mathematical constants and constants of nature (e.g., pi, eps, R, sigma)
Electrical	Library for electrical models (analog, digital, machines, multi-phase)
Icons	Icon definitions
Math	Mathematical functions (e.g., sin, cos) and operations on matrices (e.g., norm, solve, eig, exp)
Mechanics	Library to model 1-dim. and 3-dim. mechanical systems (multi-body, rotational, translational)
Media	Property models of media
Slunits	Type and unit definitions based on SI units according to ISO 31-1992
StateGraph	Library to model discrete event and reactive systems by hierarchical state machines
Thermal	Library to model thermal systems (heat transfer, simple thermo-fluid pipe flow)

Analogien für Konnektoren

Fachgebiet	Potential	Fluss
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 Flange_a
```

```
    SI.Angle phi "Absolute rotation angle of flange";
```

```
    flow SI.Torque tau "Cut torque in the flange";
```

```
end Flange_a;
```

```
connector Flange_b
```

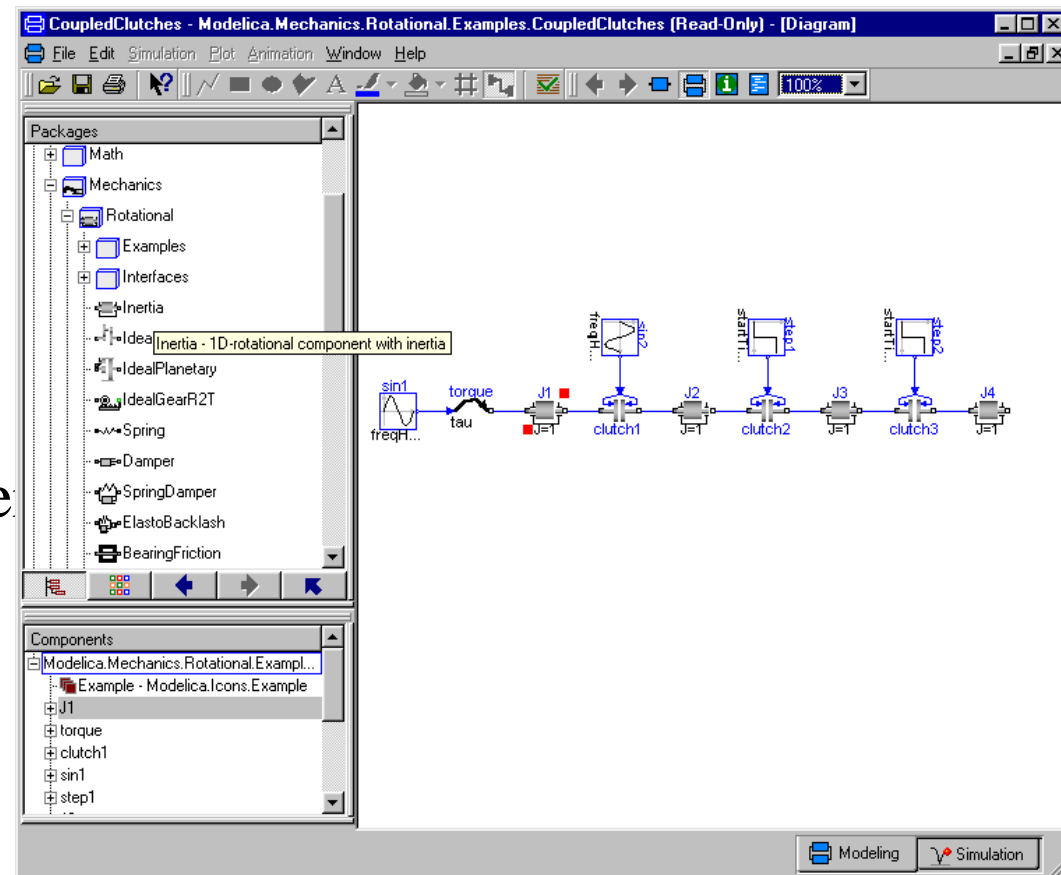
```
    SI.Angle phi "Absolute rotation angle of flange";
```

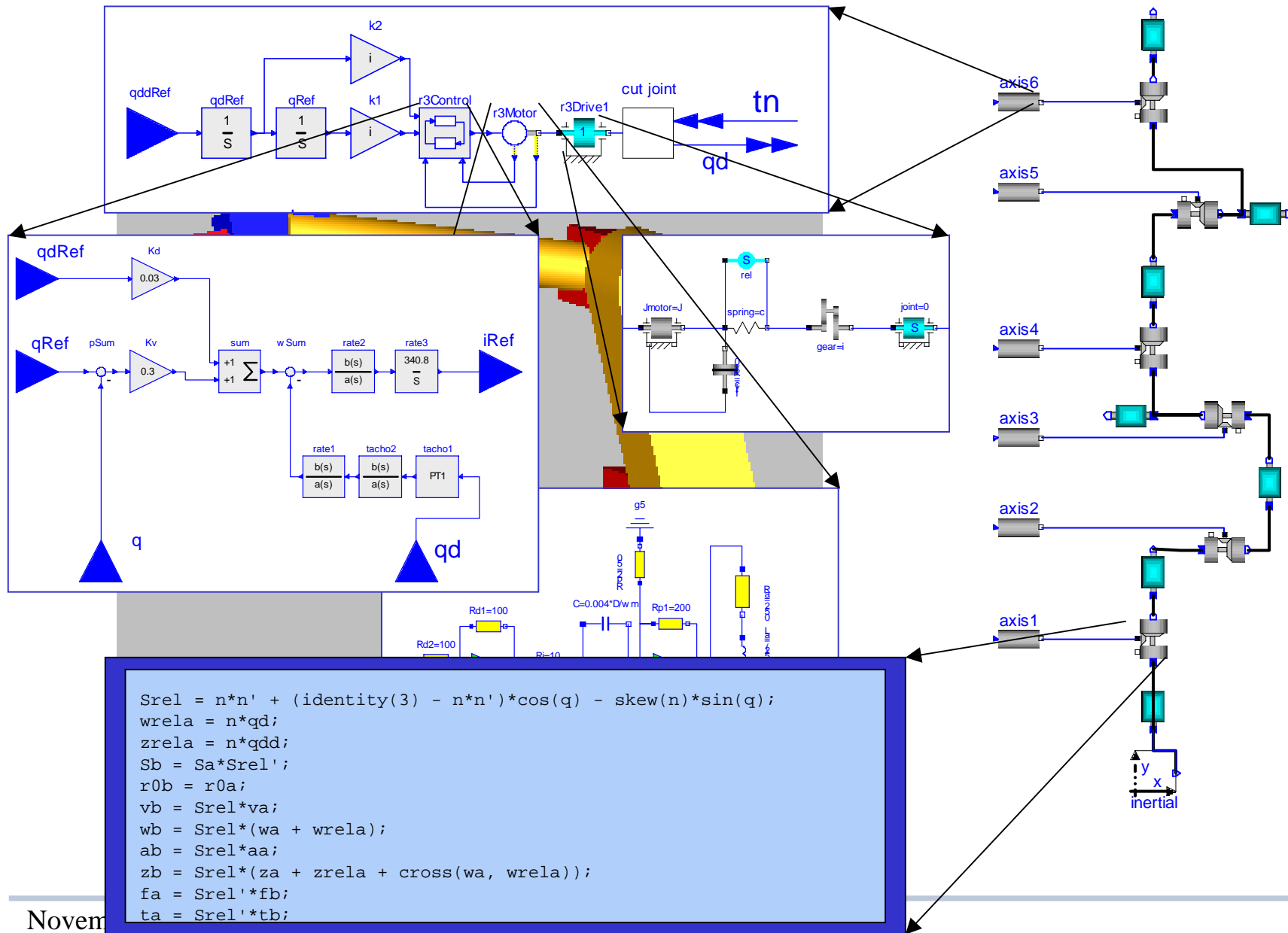
```
    flow SI.Torque tau "Cut torque in the flange";
```

```
end Flange_b;
```

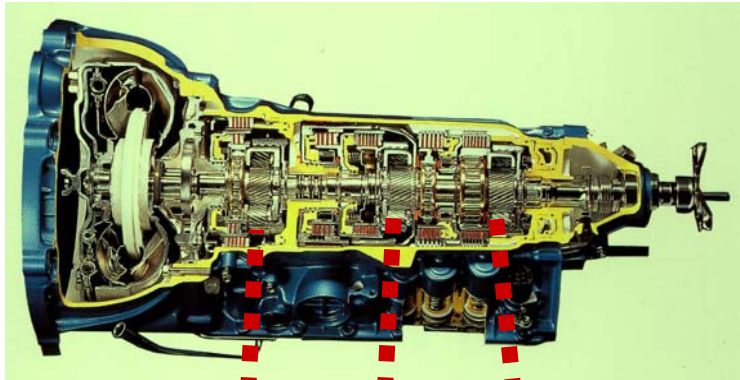
Modellierung aus Objekten

- Objekte sind in Bibliotheken
- Modell aus Grundobjekten zusammengesetzt
- Diagramm erscheint im Modellierungsfenster
- Objekte parametrieren
- Hierarchie für große Modelle



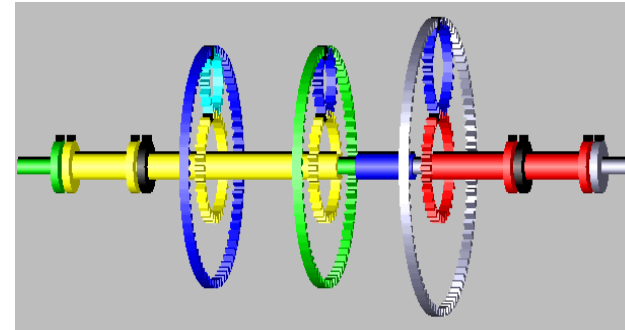
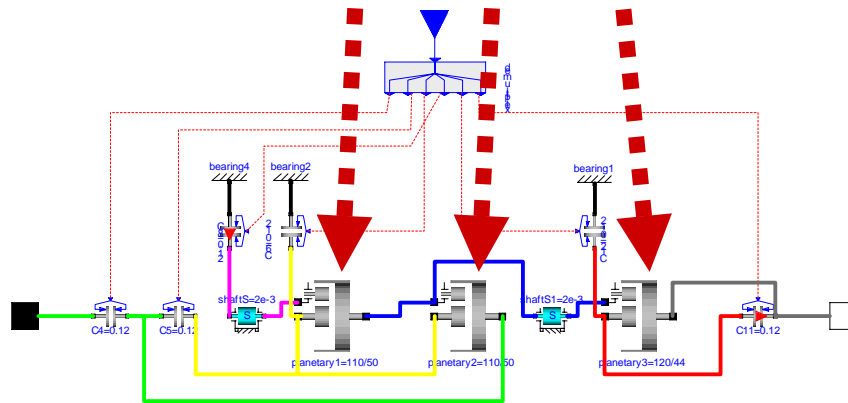


Getriebe-Modell



Courtesy Toyota Techno-Service

Powertrain Library



Modellierung von Ereignissen (event)

- z. B. Schalter, Reibung, Schlupf
- Strategie
 - Integration bis zum Ereignis
 - un stetige 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
 $y = \text{if noEvent}(u > 0) \text{ then } 1 \text{ else } -1$

Dymola: Symbolische Manipulation

- Sortierung
- Entfernung trivialer Gleichungen
- Symbolisches Lösen algebraischer Schleifen, sofern möglich
- Reduktion der Gleichungen
- Symbolische Lösung kleiner Systeme
- Index-Reduktion, falls nötig
- LU/QR-Zerlegung der Jacobi-Matrizen für lineare Systeme
- Symbolische Jacobi-Matrizen für nichtlineare Systeme

```

Translation of ITDCRT\_OpenLoopTK:
DAE having 21052 scalar unknowns and 21052 scalar equations.
Warning: The following connector variables are not used in the model and
are removed from the simulation problem:

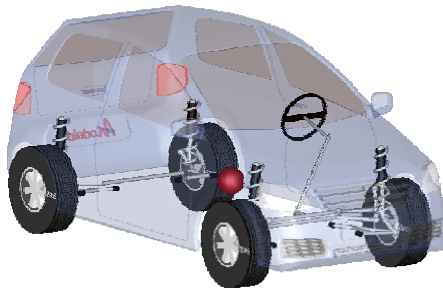
STATISTICS

Original Model
Number of components: 1312
Variables: 12140
Constants: 8 (8 scalars)
Parameters: 4586 (10587 scalars)
Unknowns: 7546 (21110 scalars)
Differentiated variables: 318 scalars
Equations: 5278
Nontrivial : 3826

Translated Model
Constants: 7840 scalars
Free parameters: 2525 scalars
Parameter depending: 6420 scalars
Inputs: 0
Outputs: 0
Continuous time states: 38 scalars
Time-varying variables: 4335 scalars
Alias variables: 10633 scalars
Assumed default initial conditions: 1
LogDefaultInitialConditions=true; gives more information
Number of mixed real/discrete systems of equations: 4
Sizes of linear systems of equations: {4, 4, 2, 2, 2, 4, 4, 2, 616, 2}
Sizes after manipulation of the linear systems: {0, 0, 2, 2, 2, 0, 0, 2, 20, 2}
Sizes of nonlinear systems of equations: {}
Sizes after manipulation of the nonlinear systems: {}
Number of numerical Jacobians: 0

Initialization problem
Sizes of linear systems of equations: {4, 7, 4, 4, 4}
Sizes after manipulation of the linear systems: {0, 0, 0, 0, 0}
Sizes of nonlinear systems of equations: {23, 1, 1, 1}
Sizes after manipulation of the nonlinear systems: {1, 1, 1, 1}
Number of numerical Jacobians: 0
    
```

Beispiel: Chassis-Modell mit 40 Freiheitsgraden



Elasto-kinematische
Radaufhängung

Quelle: Modelon, Vehicle Dynamics Library

Originalmodell: 31,689 scalar equations

80 kontinuierliche Zustandsgrößen

7649 zeitabhängige skalare Größen

Größe der linearen Gleichungssysteme

{4, 3, 3, 3, 3, 4, 3, 3, 3, 3, 2, 2, 2, 2, 566, 4, 3, 3, 3, 3, 97, 2, 4, 3, 3, 3, 3, 97, 2, 2, 2, 3, 3, 3, 3, 2}



Nach der symbolischen Manipulation

{0, 3, 3, 3, 3, 0, 3, 3, 3, 3, 2, 2, 2, 2, 29, 0, 3, 3, 3, 3, 7, 2, 0, 3, 3, 3, 3, 7, 2, 2, 2, 3, 3, 3, 3, 2}

Deutliche Reduzierung der Rechenzeit!

Dymola - Zusammenfassung

- versteht und simuliert Modelle in der objektorientierten Sprache Modelica
- Modellierungs- und Simulationsfenster
- Symbolische Manipulation
- übersetzt in C-Code
- Modelldokumentation in HTML, in gleicher Datei
- automatische Extraktion der Modelldokumentation

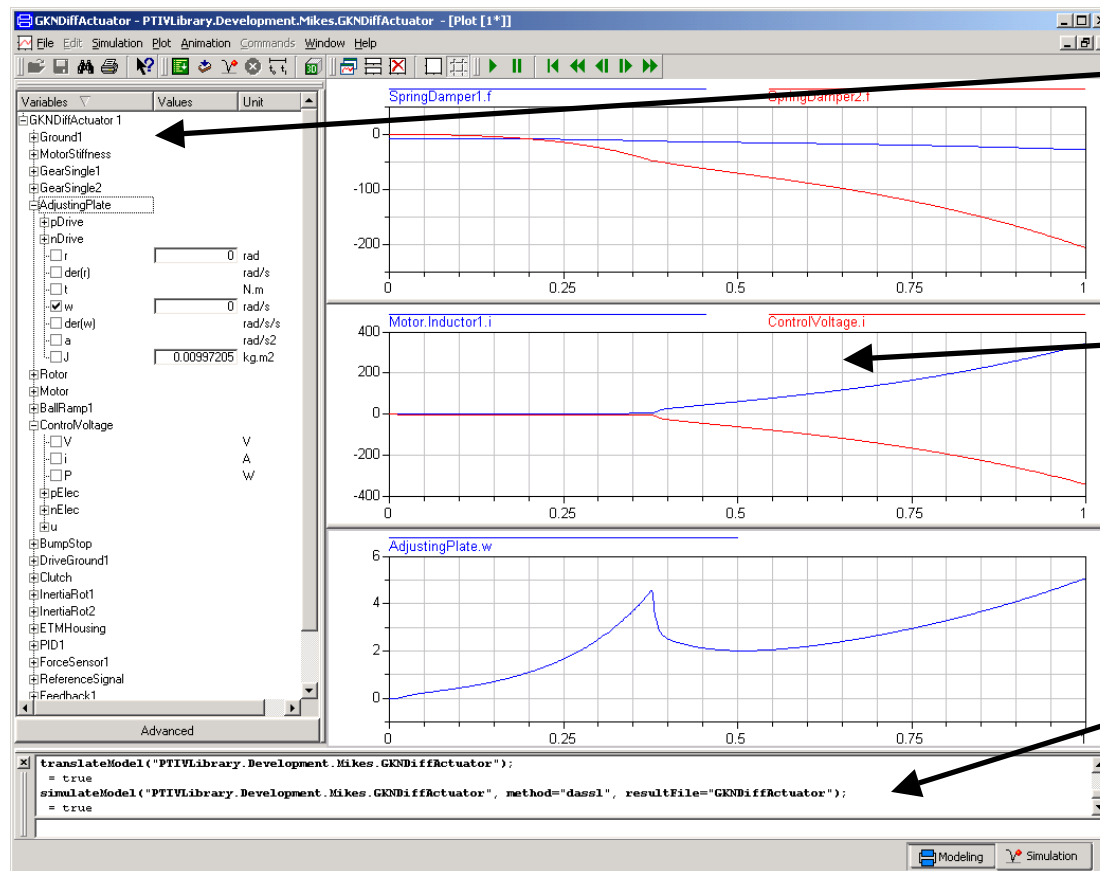
Simulation

- 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

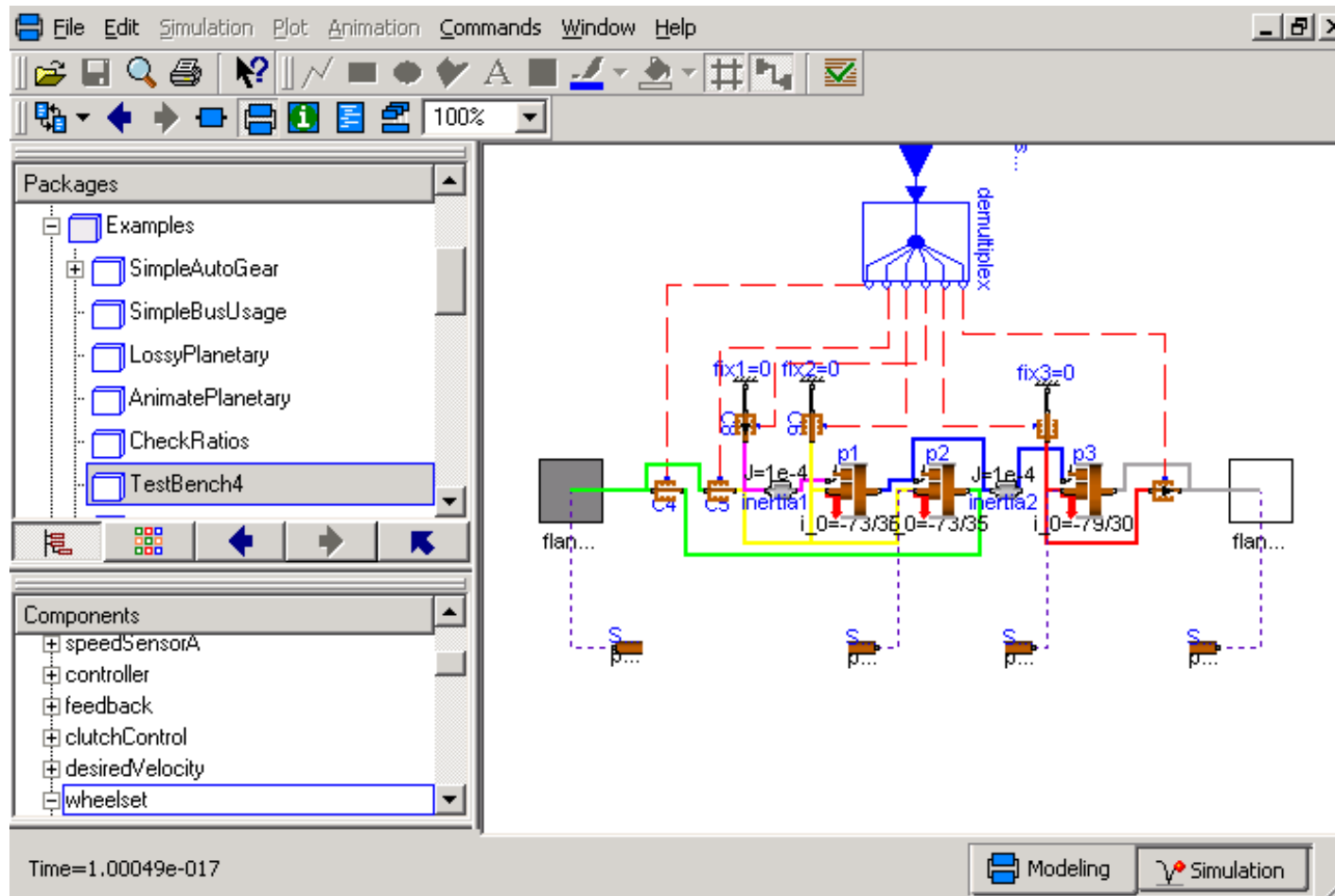
Plots

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

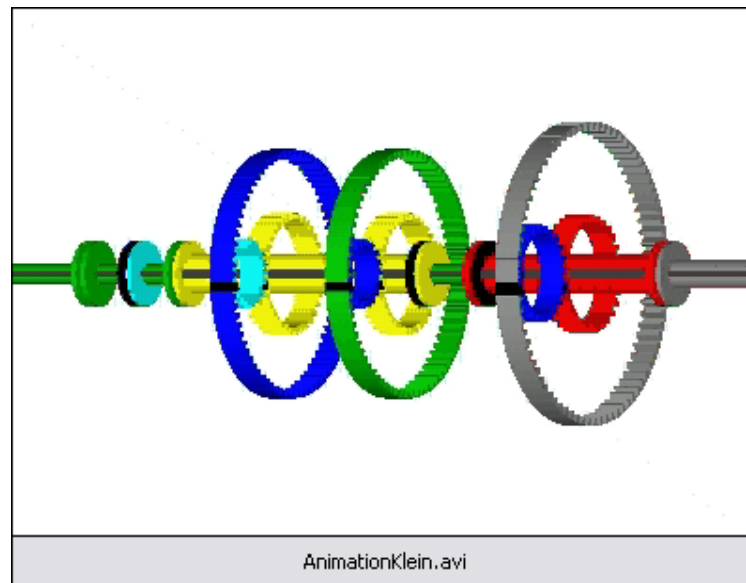
Command window

Befehlsprache und Aufruf von Skripten möglich

Animationsbeispiel: Getriebe



Animation



Dymola Experiment Skriptfile

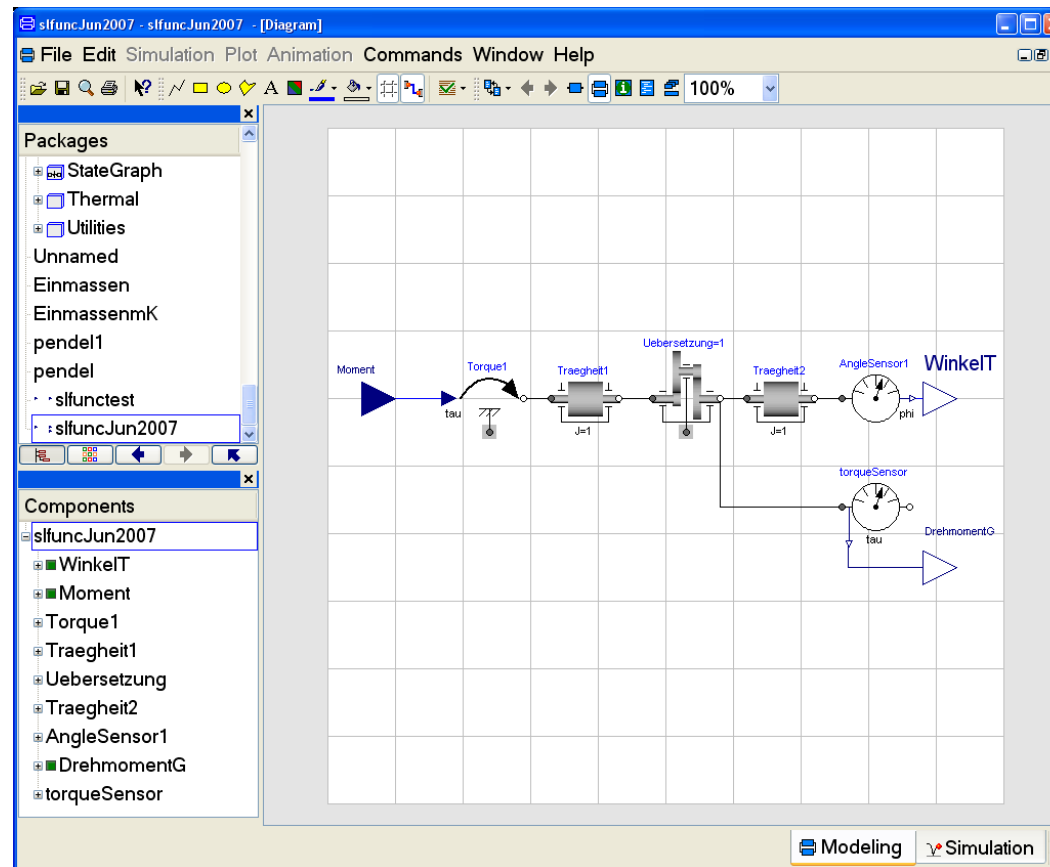
- Ablaufsteuerung
- Parametervariationen
- Plotmöglichkeiten
- Modelica Syntax
- benutzerdefinierte Funktionen

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;
```

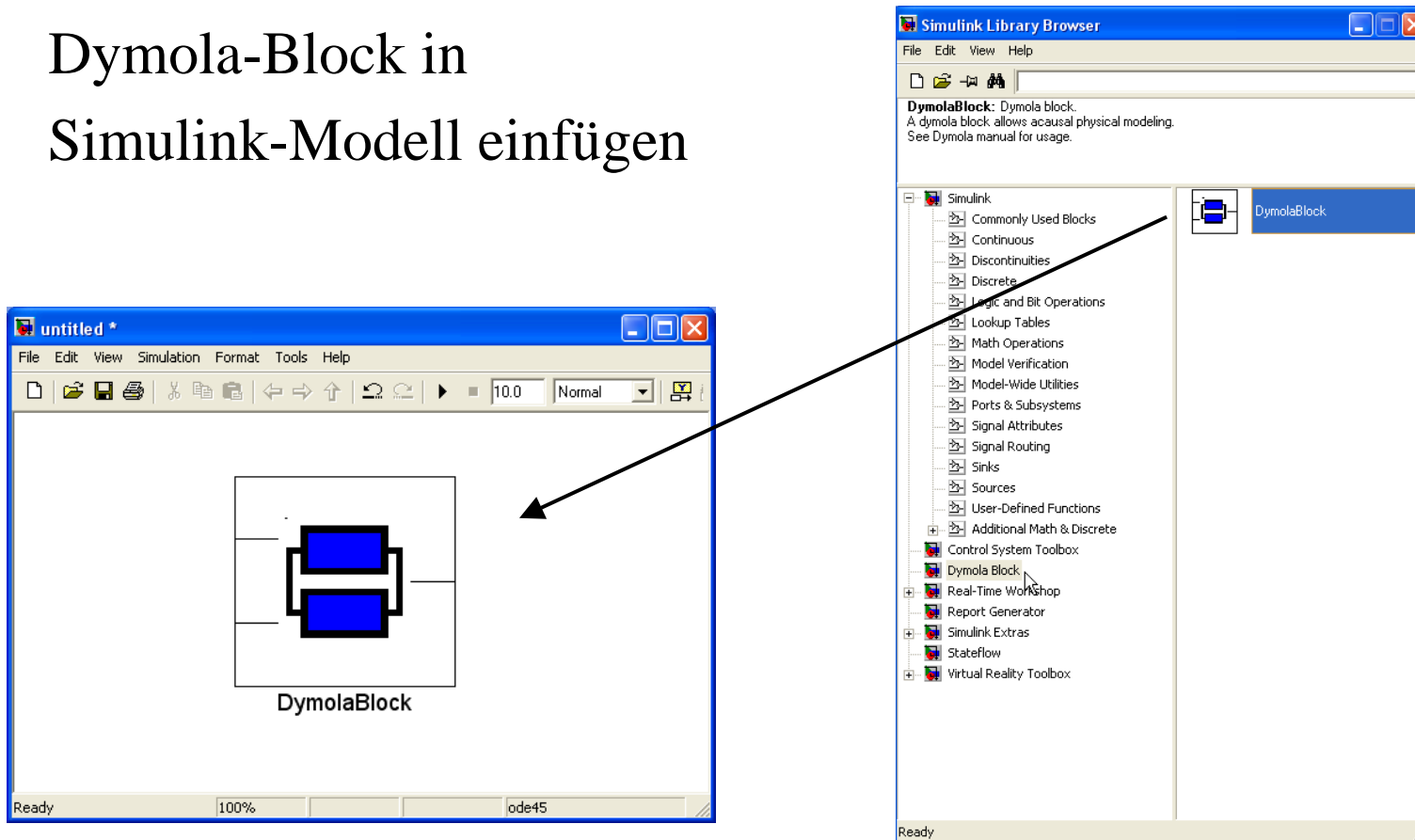
Dymola-Simulink-Interface

- Ein-/Ausgänge im Dymola-Modell festlegen



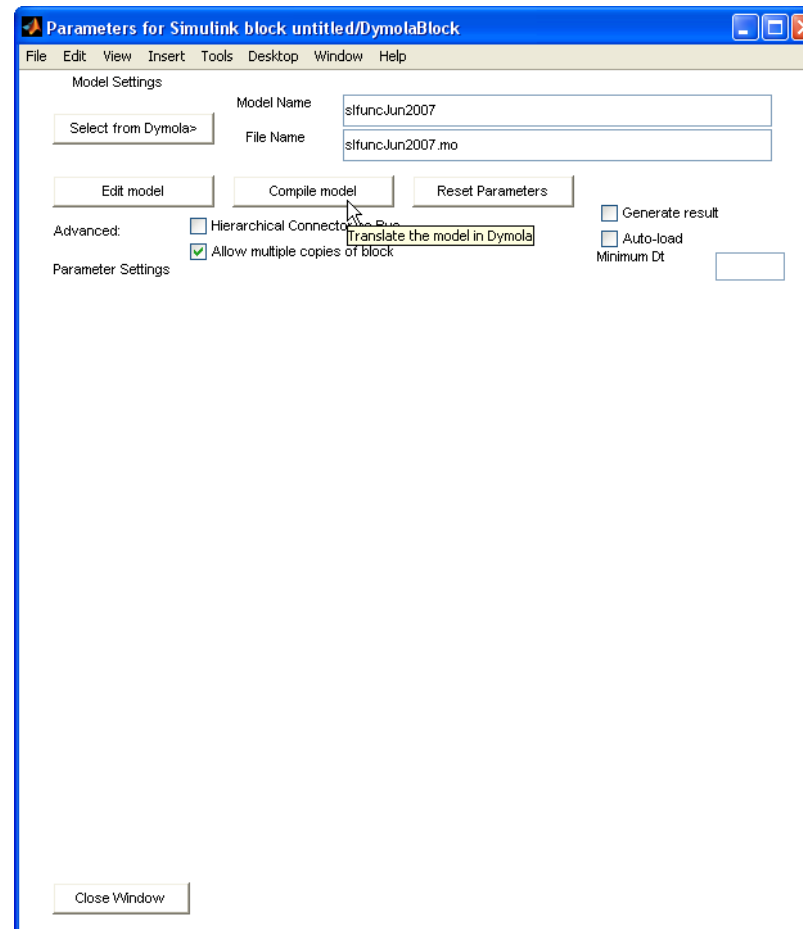
Dymola-Simulink-Interface

- Dymola-Block in Simulink-Modell einfügen



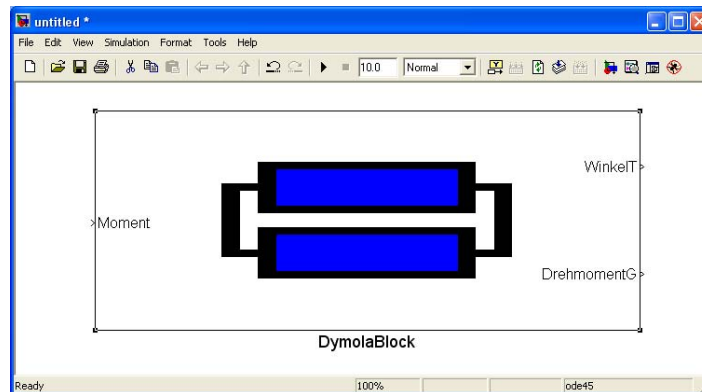
Dymola-Simulink-Interface

- Dymola-Block öffnen
- Modell in Dymola übersetzen
- Erzeugt S-Function als DLL



Dymola-Simulink-Interface

- ergibt Block mit Ein- und Ausgängen
- Modellparameter lassen sich noch ändern



The screenshot shows the 'Parameters for Simulink block' dialog box for 'DymolaBlock'. The dialog has a menu bar (File, Edit, View, Insert, Tools, Desktop, Window, Help) and a title bar. It is divided into several sections:

- Model Settings:** Includes a 'Select from Dymola>' button, 'Model Name' (slfuncJun2007), and 'File Name' (slfuncJun2007.mo). Buttons for 'Edit model', 'Compile model', and 'Reset Parameters' are present.
- Advanced:** Includes checkboxes for 'Hierarchical Connector as Bus' (unchecked) and 'Allow multiple copies of block' (checked). There are also checkboxes for 'Generate result' and 'Auto-load'.
- Parameter Settings:** A list of parameters with input fields:

Traegheit1.J	1
Traegheit1.ph_i_start	0
Traegheit1.w_start	0
Traegheit1.a_start	0
Uebersetzung.ratio	1
Traegheit2.J	1
Traegheit2.ph_i_start	0
Traegheit2.w_start	0
Traegheit2.a_start	0
- Start values:** Includes input fields for 'Traegheit2.ph_i' (0) and 'Traegheit2.w' (0).

A 'Close Window' button is located at the bottom left.

Dymola 6.2 (7.0?) Neuigkeiten

- Verbesserung des graphischen Editors
- Anzeige von Einheiten
- Einheitenprüfung möglich
- Modelica 3 wird weitgehend unterstützt
- Verbesserung der Simulink-Schnittstelle

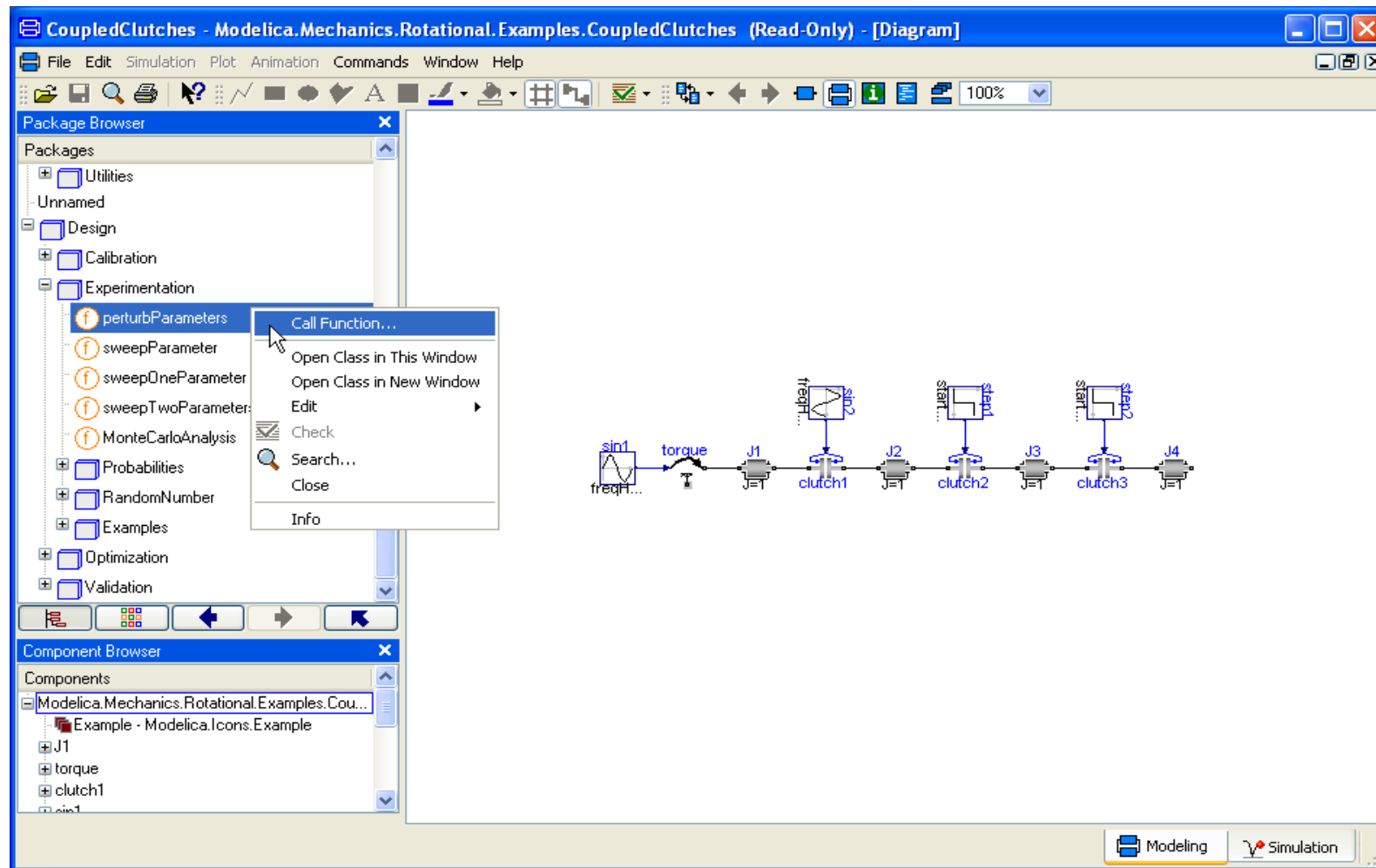
Einige Neuigkeiten in Modelica 3.0

- Bereinigung der Sprachdefinition
- Modelle müssen „locally balanced“ sein, d.h. die Zahl der Gleichungen und Unbekannten muss gleich sein
- Klarere Sprachdefinition
- „built-in“-Operatoren erzeugen keine Events mehr
- Elementweise Operatoren wie in MATLAB
- Weitere „annotations“
- Neue Attribute
- Sh. www.modelica.org

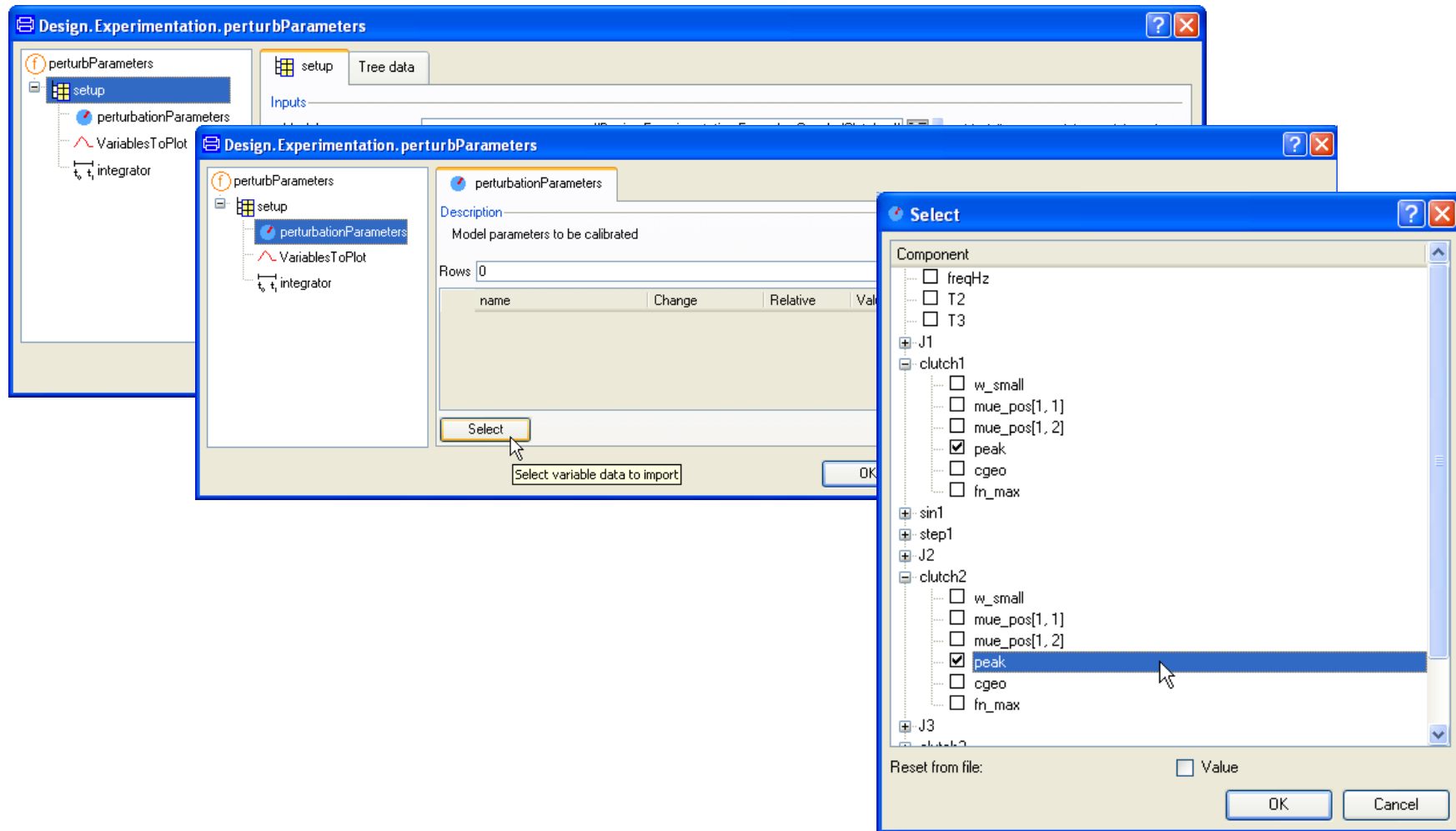
Ergänzungen zu Dymola

- Automatisierte Parameterstudien
- Kalibrierung von Parametern
- Optimierung
- 3D-Visualisierung
- Verschlüsselung der Modelle
- Schnittstelle zu Versionsverwaltungsprogrammen

Parameterstudien

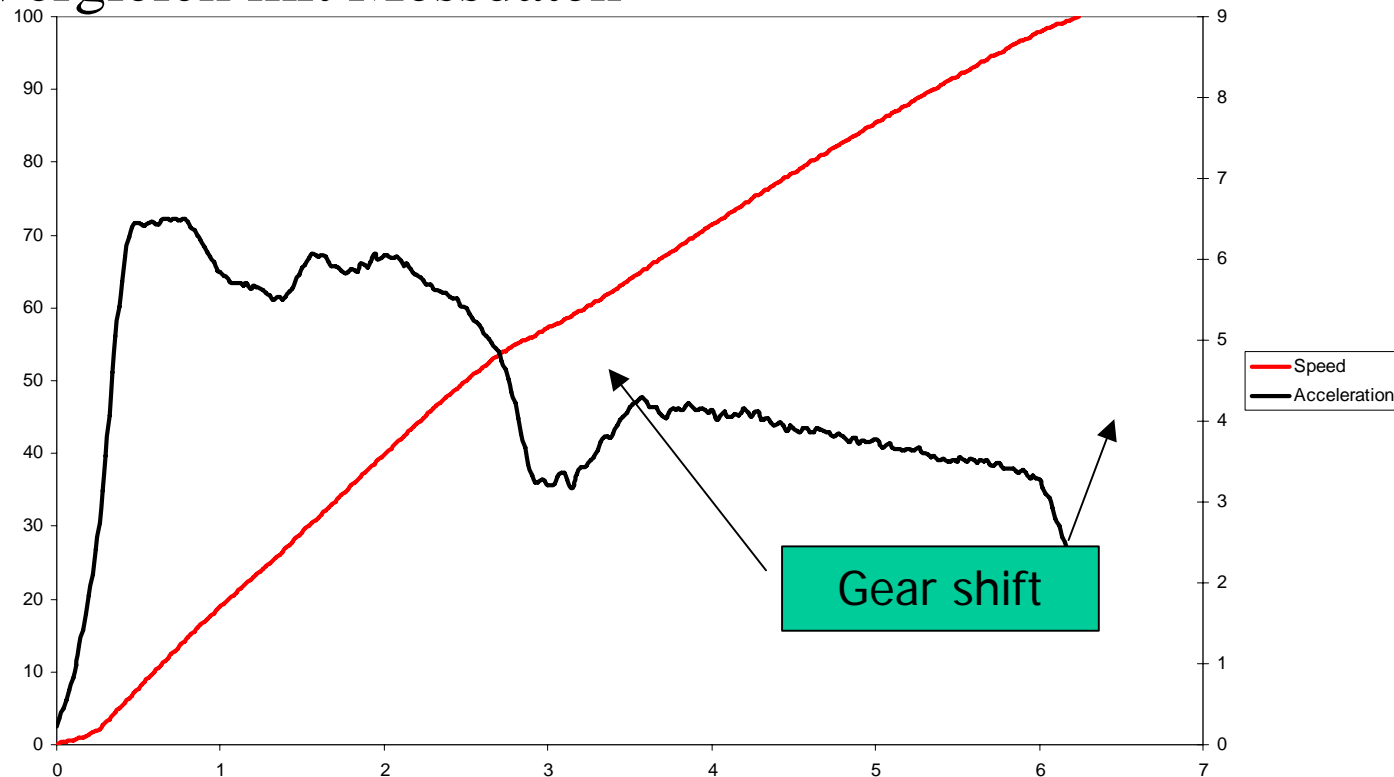


Parameterstudien



Kalibrierung

Bestimmung unbekannter oder unsicherer Modellparameter durch Vergleich mit Messdaten



Quelle: Auto Mobil, Issue 2, 2005

Kalibrierung

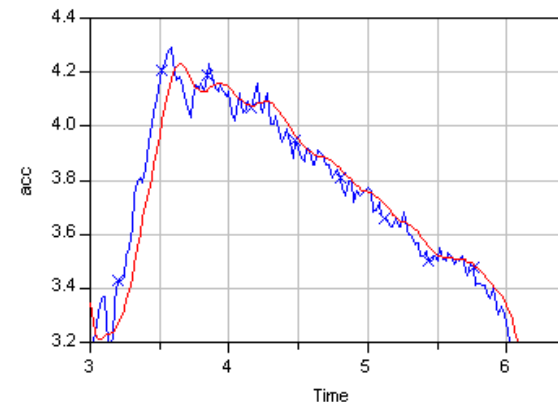
Messdaten

- CSV-Dateien
- MATLAB-MAT-Files V4

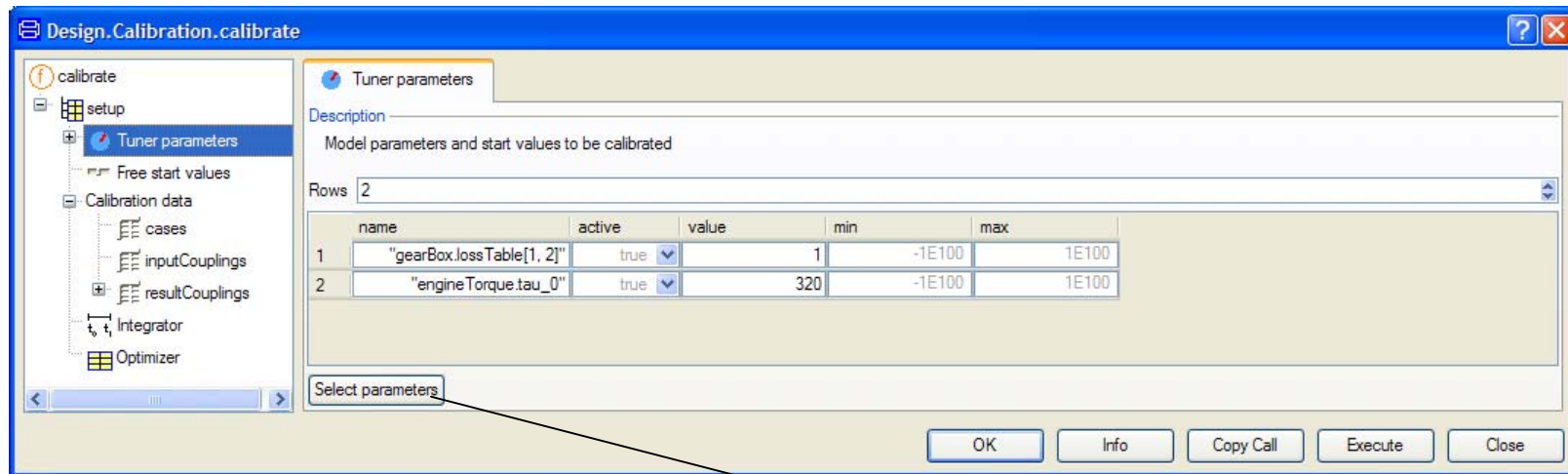
	A	B	C	D
1	time	speed	dist	acc
2	0	0	0	0.22
3	0.02	0.2	0	0.33
191	3.78	68.1	37.84	4.14
192	3.8	68.4	38.22	4.16
193	3.82	68.7	38.6	4.13

Daten vorbereiten, z.B.

- Daten filtern
- Daten begrenzen
- Trend entfernen

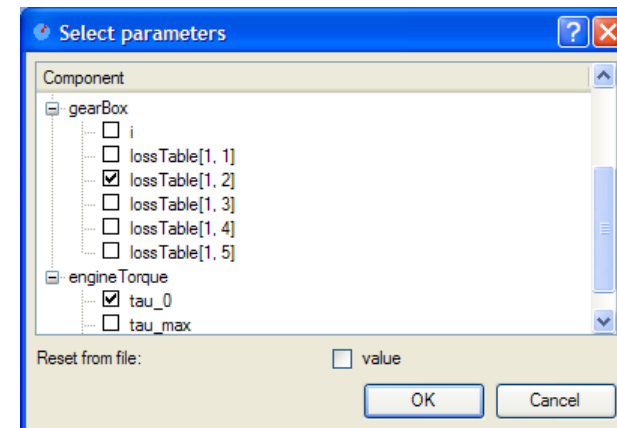


Kalibrierung



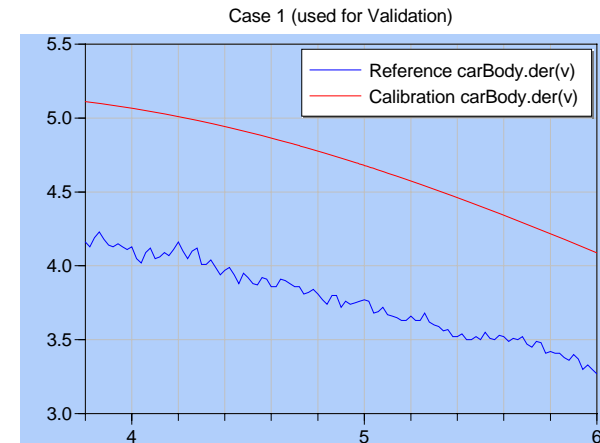
Einfach anzuwenden durch Dialog

- Parameter und Startwerte
- Dateien mit Messdaten
- Zusammenhänge zwischen Daten und Parametern

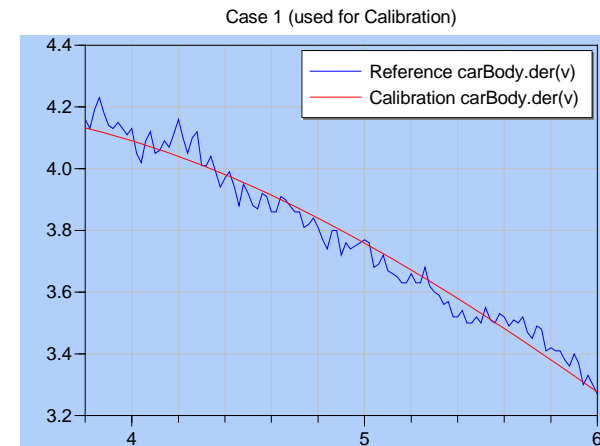


Kalibrierung

Startwerte

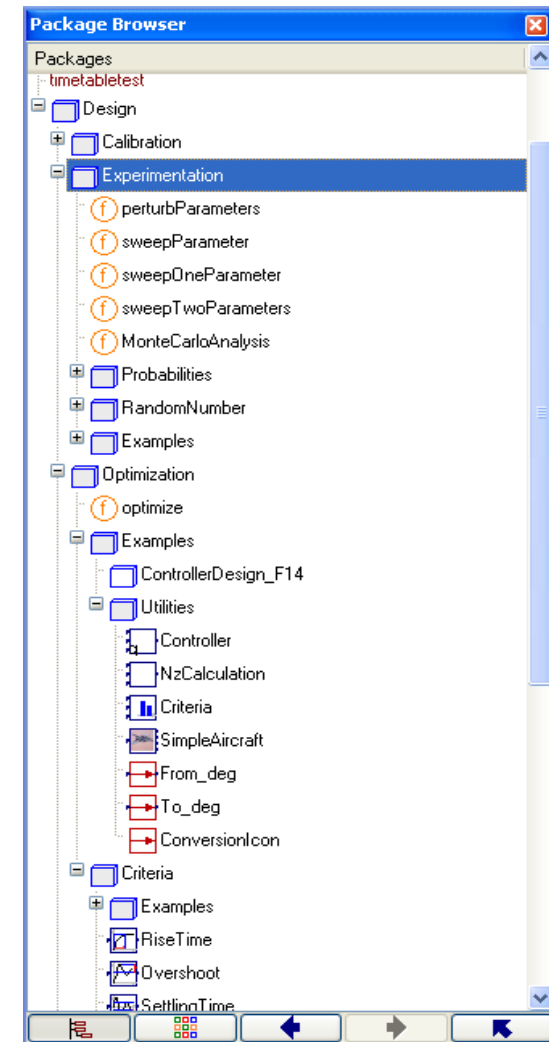


Nach der Kalibrierung

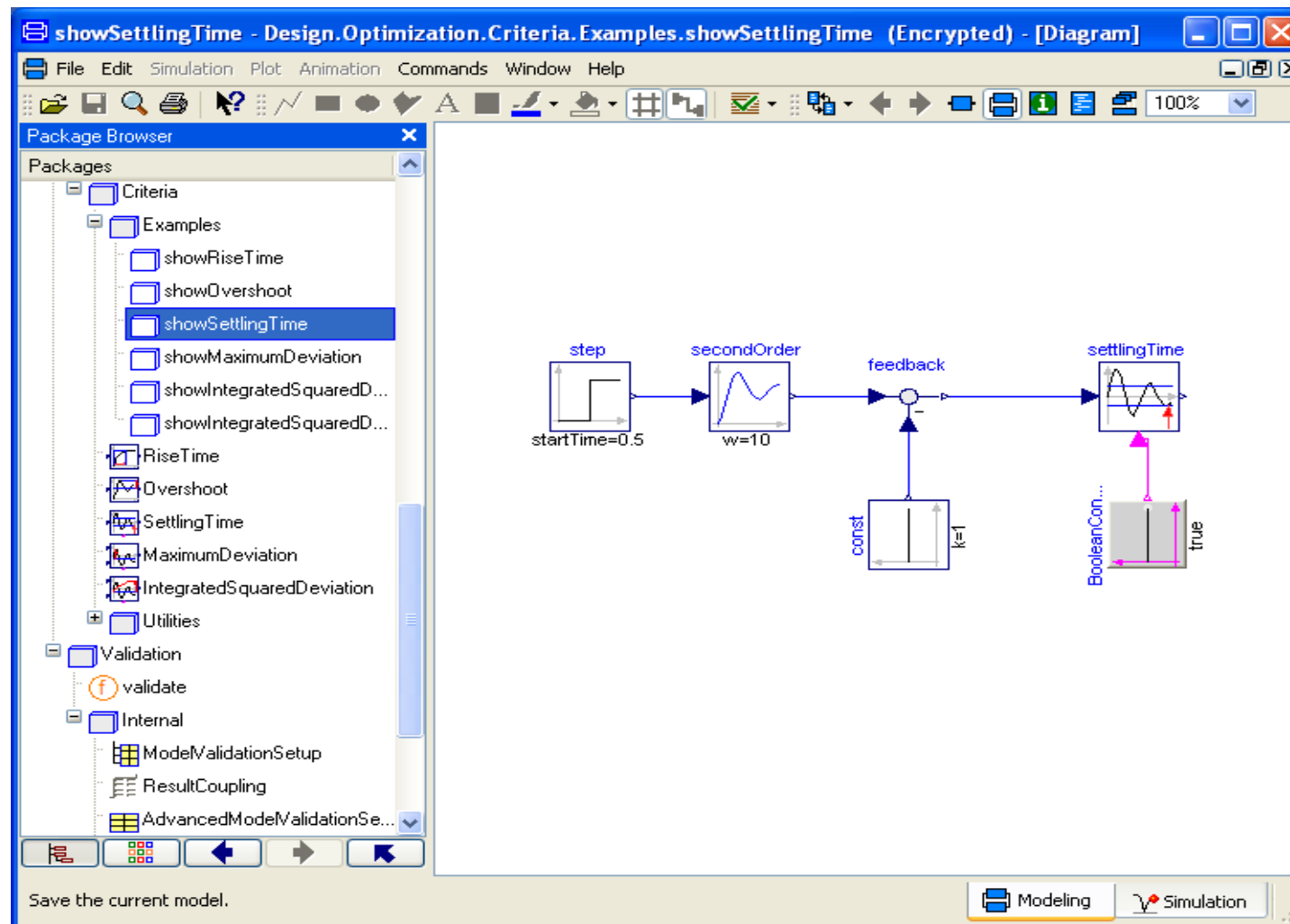


Optimierung

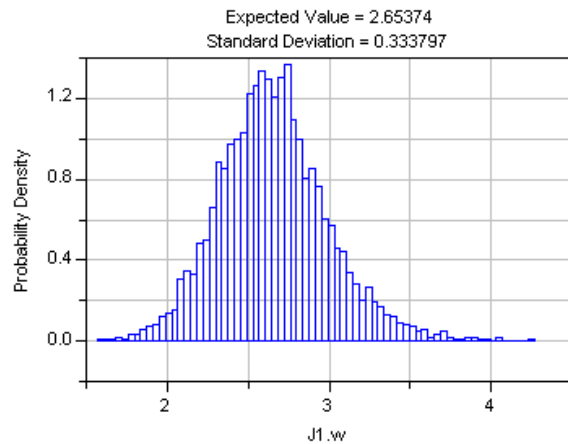
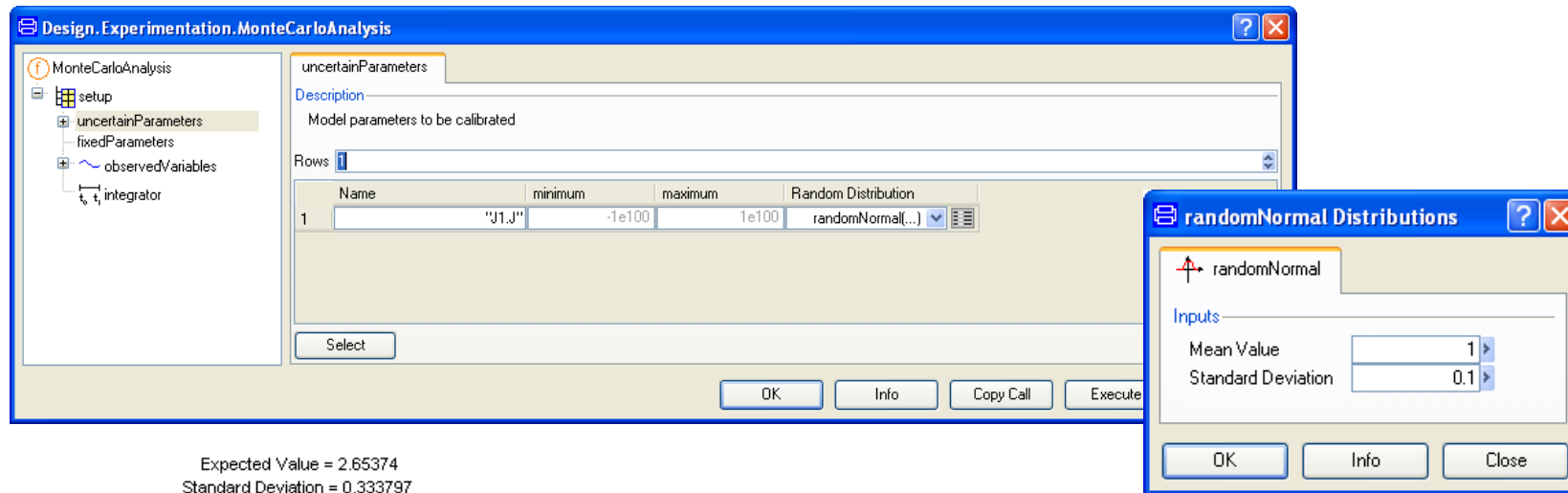
- Bestimmung von Parametern
- Definition einer Zielfunktionen
- Mehrere Zielfunktionen können kombiniert werden
- Mehrere Optimierungskriterien sind möglich



Optimierung



Monte-Carlo-Analyse



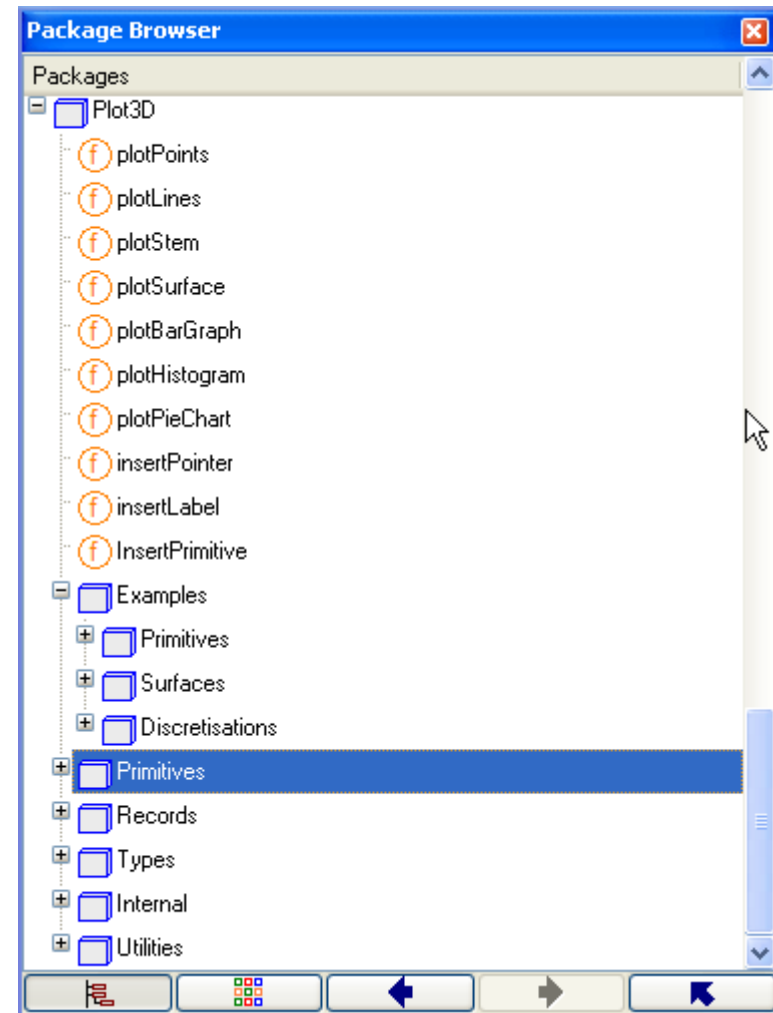
J1.J des Kupplungsmodells
genüge einer Normalverteilung

Verteilung der
beobachteten Variablen

Verteilung J1.w zum Zeitpunkt 1.2 s

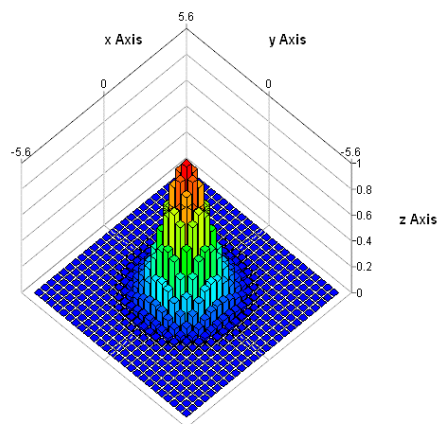
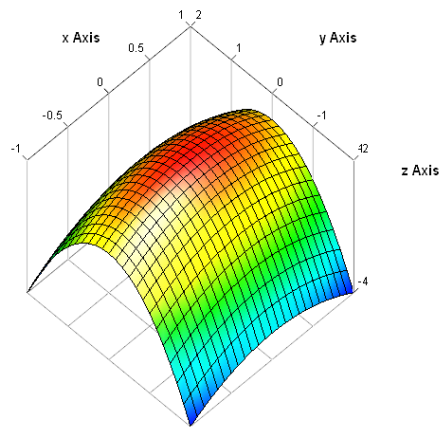
3D-Graphik

- vorbereitete Befehle
- interaktiv oder aus Skript-Sprache
- angepasst für Simulation

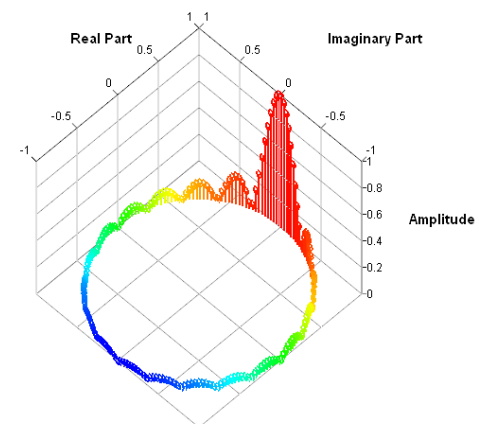
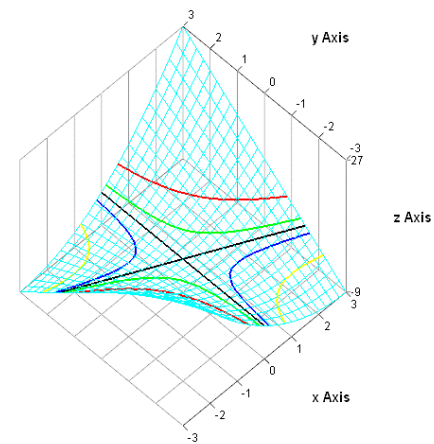


3D-Graphik - Beispiele

Flächen



3D-Kurven



Weitere Eigenschaften

- Verschlüsselung der Modelle
 - Encryption speichert Modelle binär ab
 - Scrambling entfernt vertrauliche Struktur und Daten
- Modellverwaltung (Model Management)
 - Einbindung in übliche Versions-Verwaltungsprogramme
CVS (current version systems) und
SVN (subversion systems)
- Komfortable Erstellung eigener GUIs
- Verbesserung der Scripting-Sprache (MOS-Files)

Weitere Eigenschaften

Modellabhängigkeit (Model Dependency)

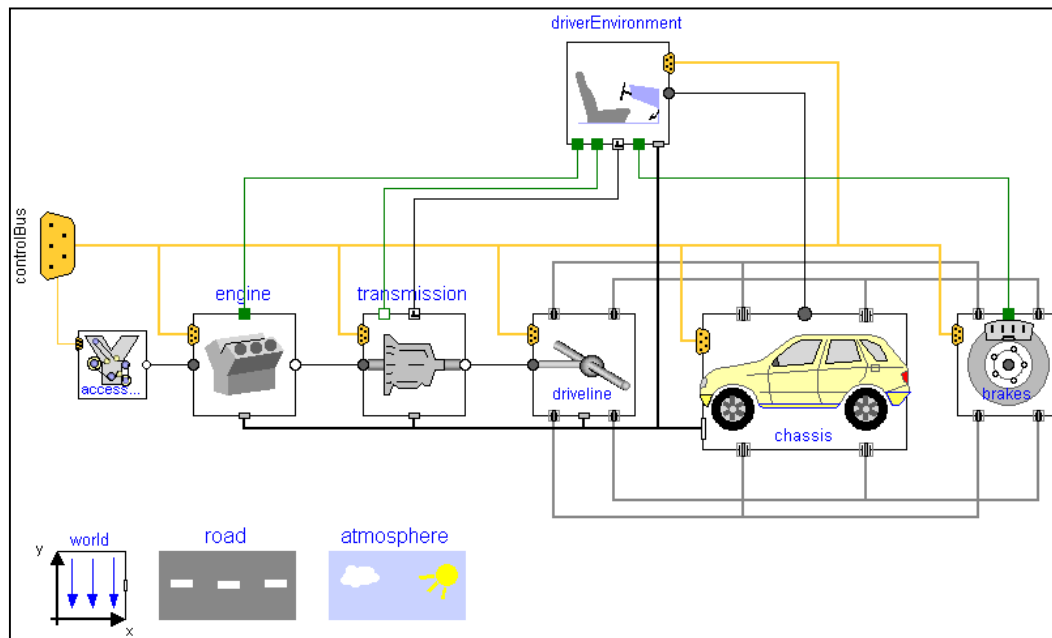
- Zeigt die Abhängigkeit eines Package von den Klassen
- Links zur HTML-Dokumentation

These classes have been referenced in this package.

Class	Referenced From
Plot3D	sweepTwoParameters
Design.Internal.Records.MatCsvFileName	dataPreprocessing
Design.Internal.Records.MatCsvFileNameOut	dataPreprocessing
Design.Internal.Records.ModelCalibrationSetup	calibrate , checkCalibrationSensitivity , perturbParameters , sweepParameter , sweepTwoParameters
Design.Internal.Records.PerturbationParameter	perturbParameters
Design.Internal.Records.PreprocessingSignal	dataPreprocessing
Modelica.Utilities.Streams	dataPreprocessing
Modelica.Utilities.Streams.print	checkCalibrationSensitivity , sweepTwoParameters
Modelica.Utilities.Strings	checkCalibrationSensitivity , perturbParameters , sweepTwoParameters
Modelica_LinearSystems	dataPreprocessing

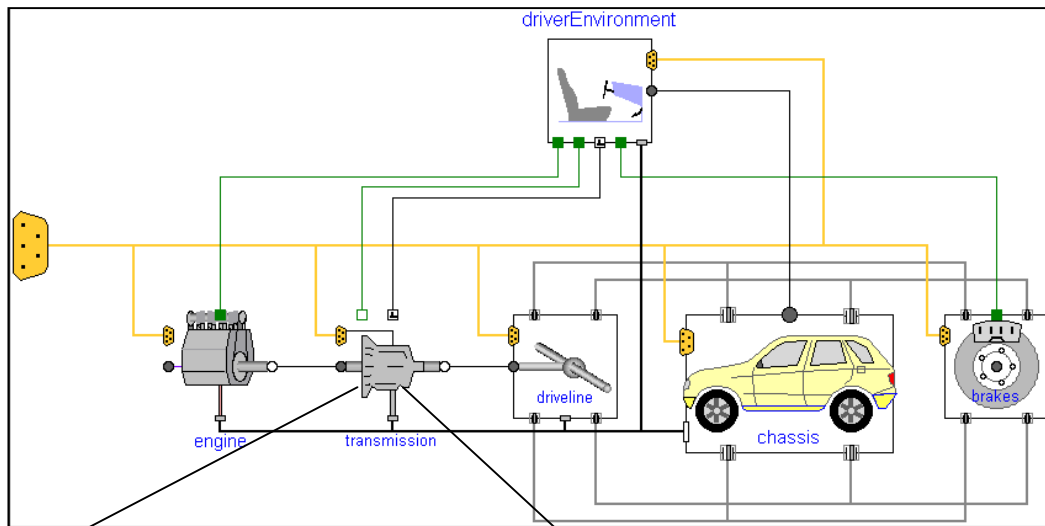
Neue Bibliotheken

- Powertrain-Library Version 2.0, angekündigt
- Flexible Bodies Library, angekündigt
- Smart Electric Drives Library
- Vehicle Dynamics Library

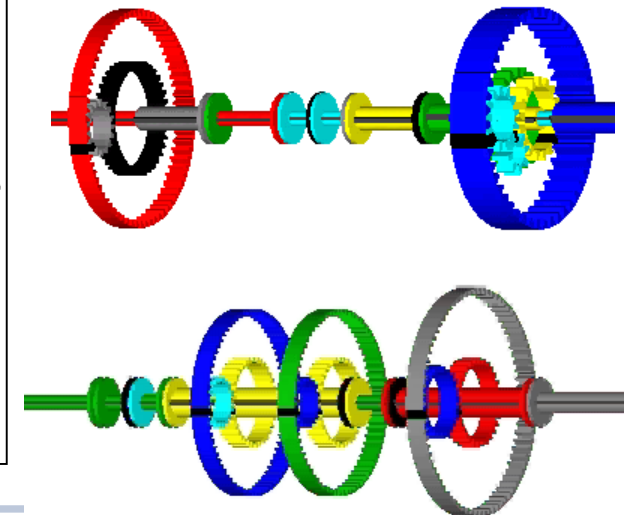
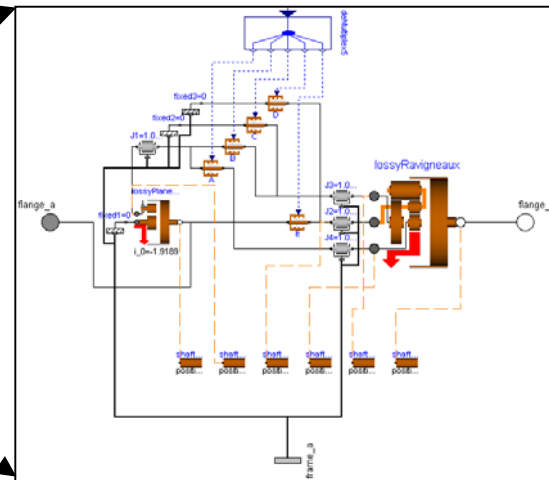
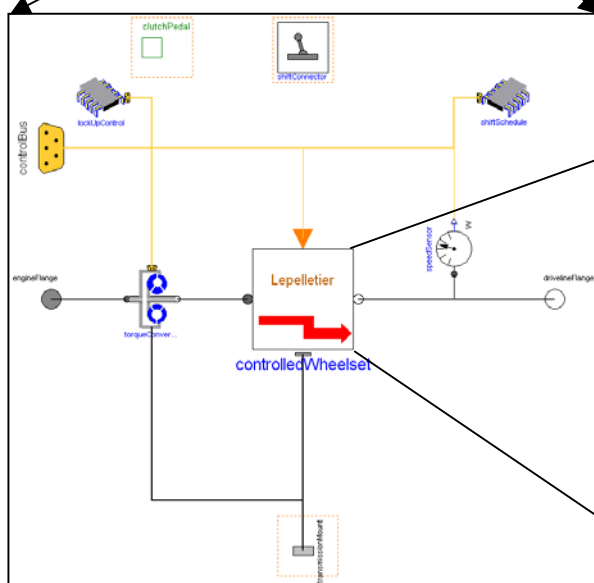


Powertrain-Library 2.0



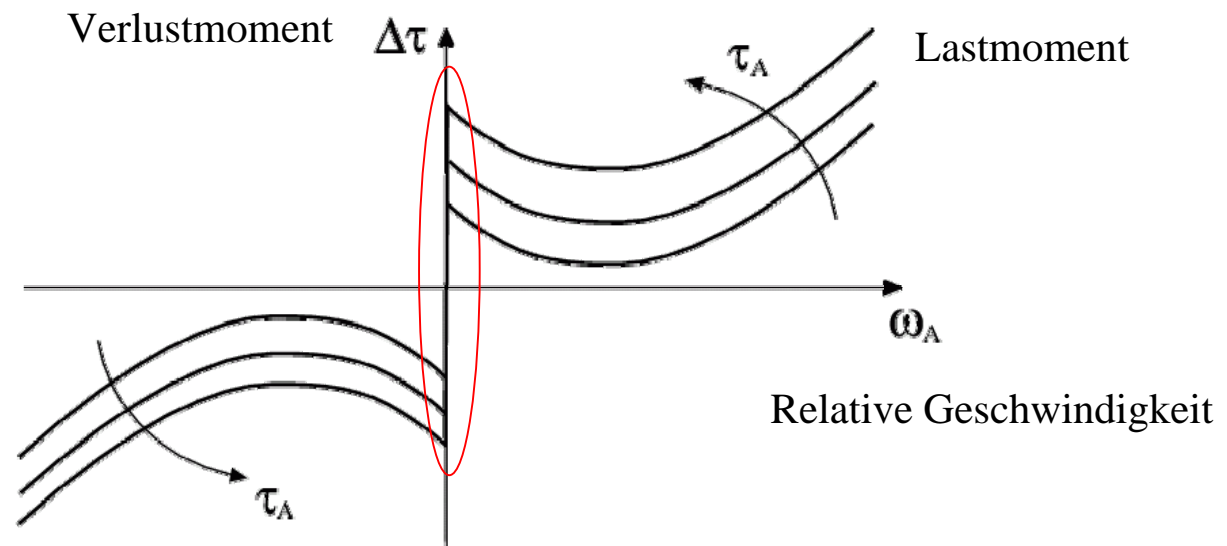
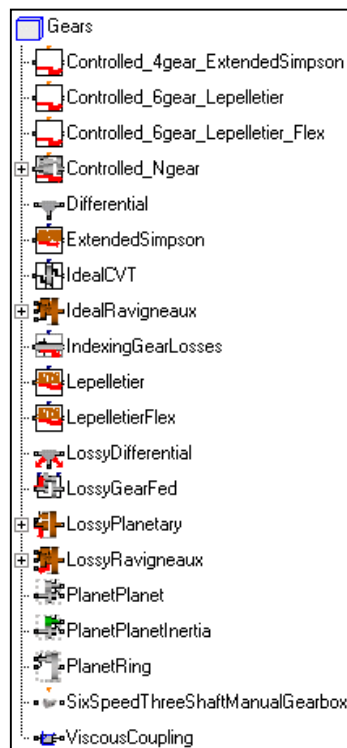


- Wesentliche neue Version
- 70 Komponenten
- Längsdynamik
- Standard- und Planetengetriebe mit Verlusten, Kupplungen mit Reibung
- Flexible Antriebsstrangmodelle
- Arbeitet mit der 3D Vehicle Dynamics Library



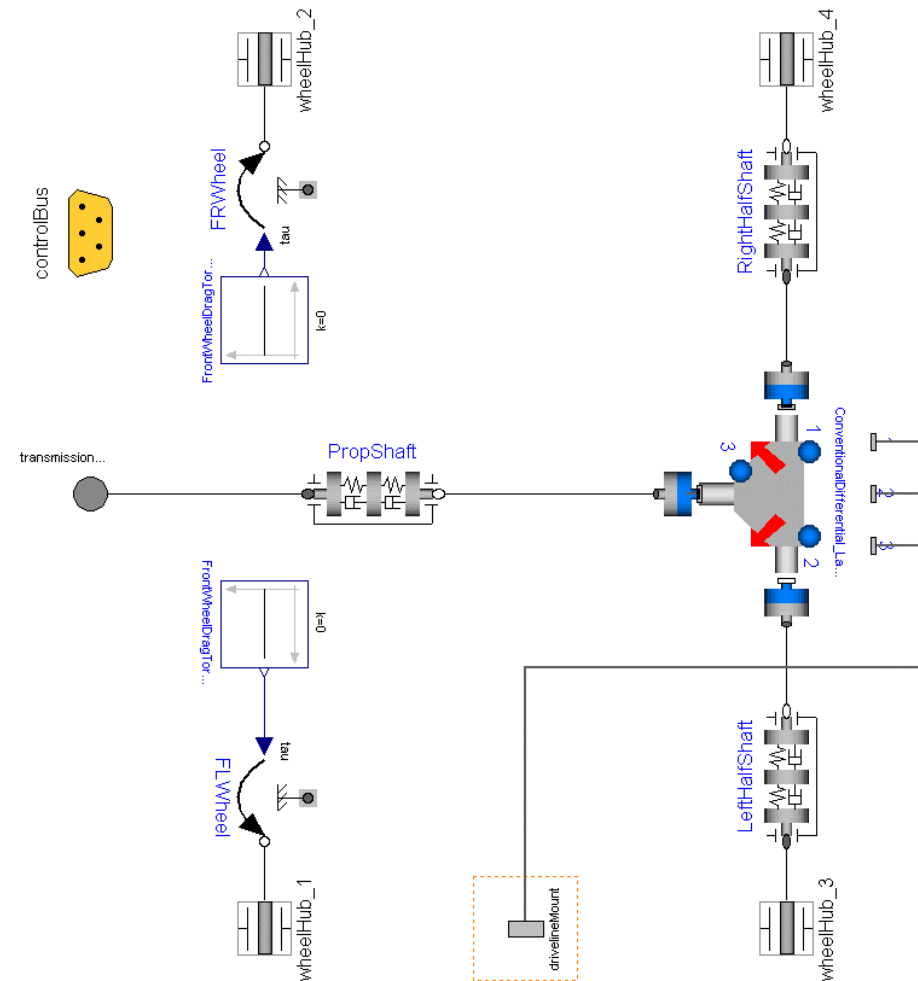
Modellierung der Verlusteffizienz

- Modellierung der Getriebeverluste ist in vielen Modellen enthalten.



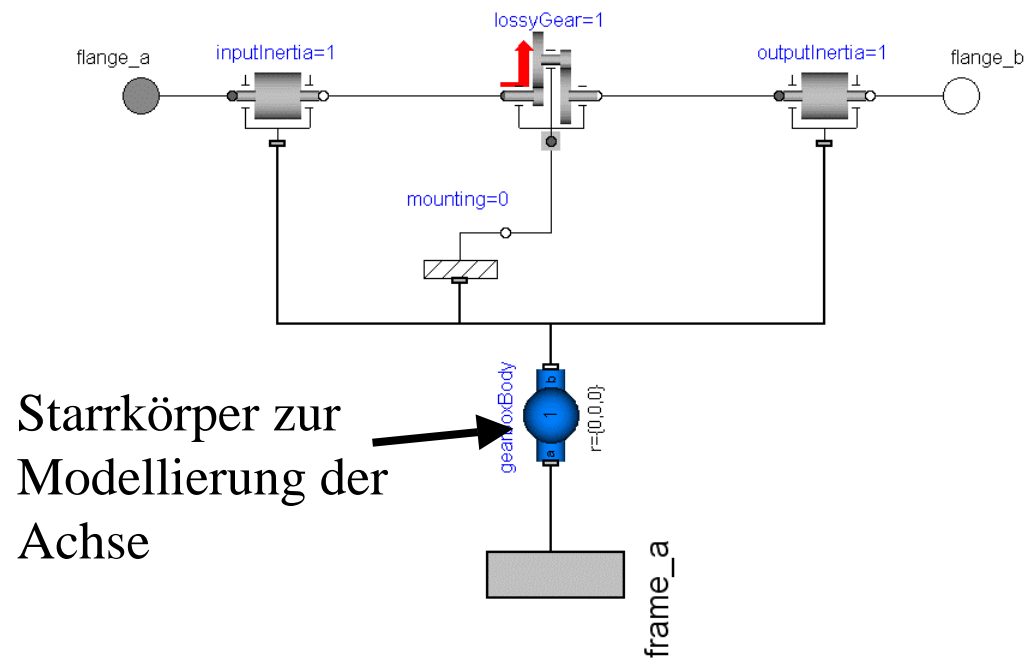
Modellierung der Elastizität

- Vibrationen
- Wichtig für Fahrverhalten und Schaltqualität
- Steifigkeit, Dämpfung, Spiel



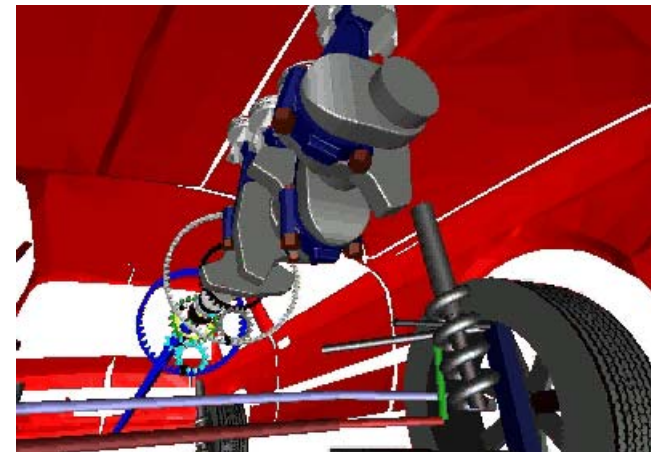
Optionale 3D-Effekte

Nur wenige zusätzliche Daten

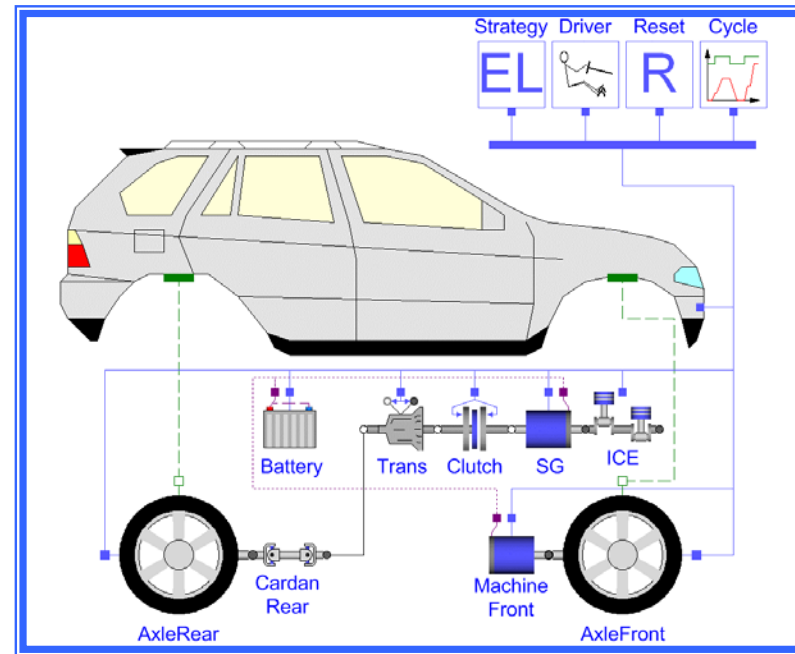


Starrkörper zur Modellierung der Achse

Schnittstelle zum Fahrzeugmodell



Antriebsstrang mit 3D-Fahrzeugmodell



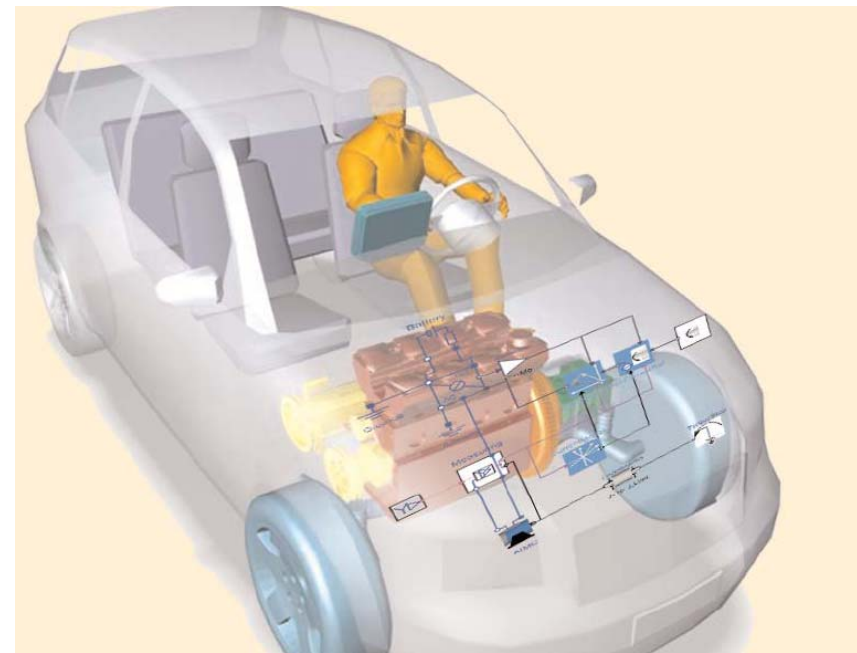
The Smart Electric Drives (SED) Library

arsenal research

An Enterprise of the Austrian Research Centers.

Anwendungen im Automobilbereich

- Simulation des vollständigen elektrischen Antriebssystems, einschließlich alternativer Antriebskonzepte
- Minimaler Verbrauch durch Zuschaltung elektrischer Antriebskomponenten
- Elektrische Leistung im Fahrzeug
- Entwicklung und Optimierung der Regelstrategien

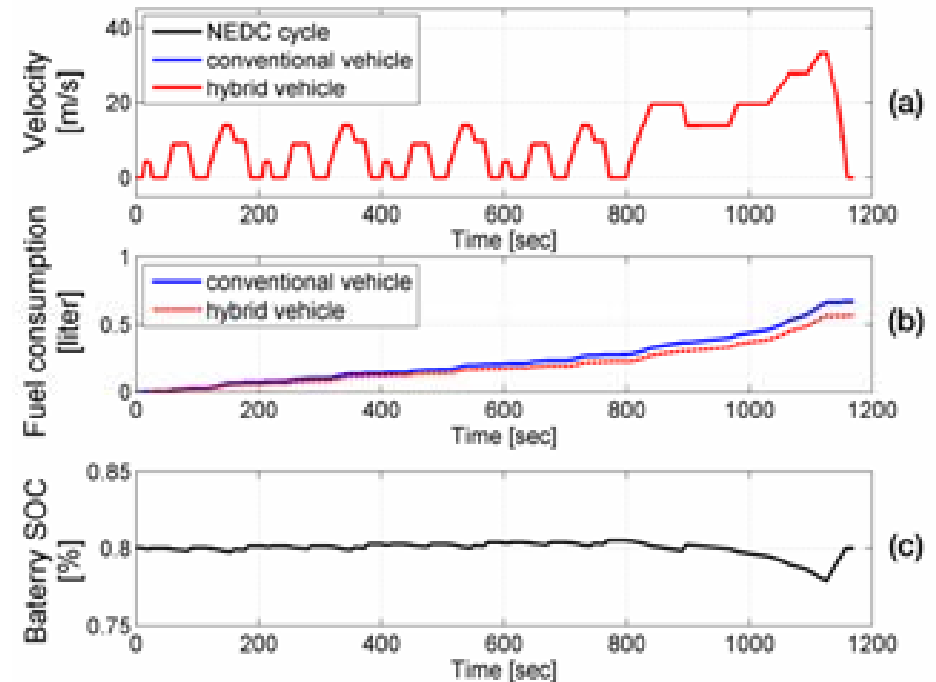
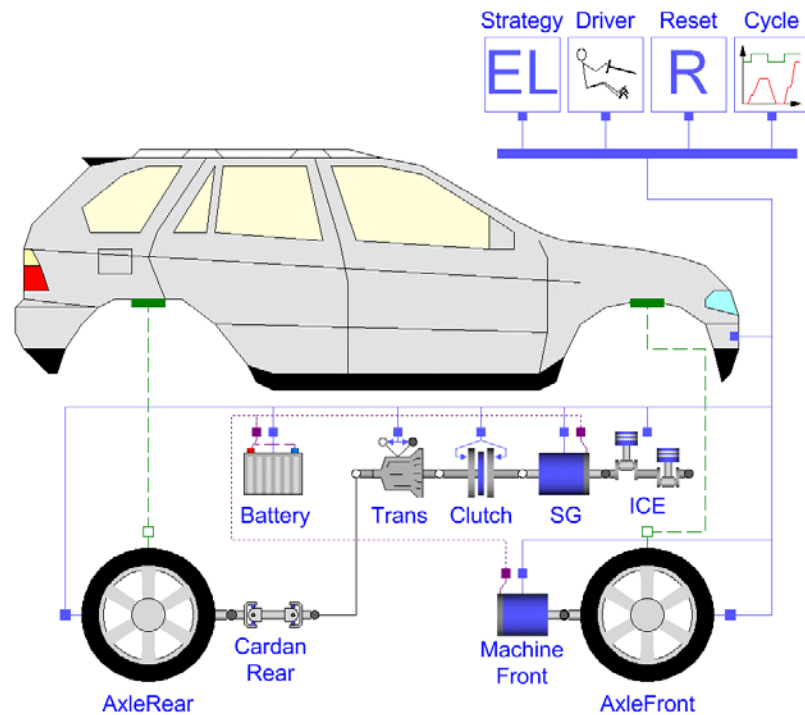


Smart Electric Drives (SED) Library

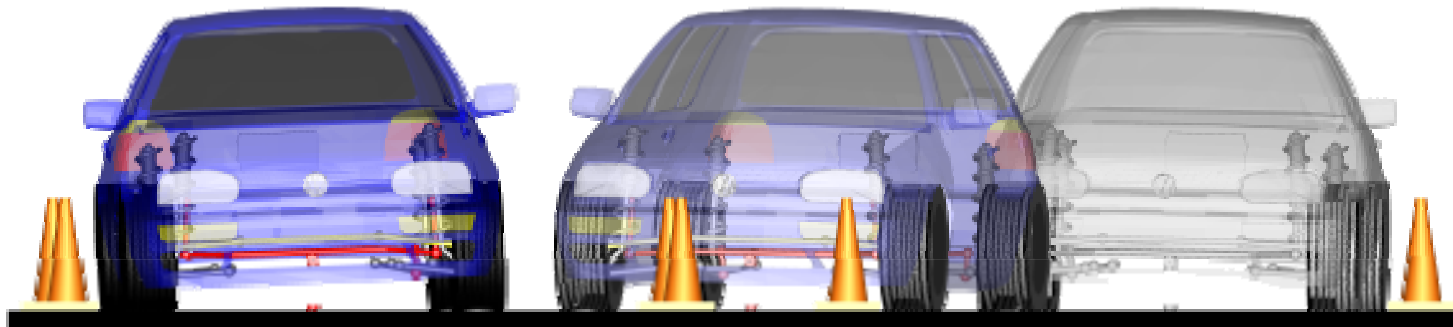
- Komponenten, z.B.
 - Asynchronmaschinen, Permanentmagnet-Synchronmaschinen, weitere Gleichstrommaschinen
 - Regelungen: Feldorientierte Regelungen, bürstenloser Gleichstrommotor
 - Stromrichter, verschiedene Abstraktionstiefen
 - Konverter, Energiequellen (Batterien, Brennstoffzellen)
- Anwendungen, z.B.
 - Hybrid Elektrische Fahrzeuge (HEV) zusammen mit der Powertrain Library
 - Starter-Generatorkonzepte
 - Elektrisch angetriebene Wasserpumpe
 - Elektrisch angetriebene Ölpumpe
 - Elektrisch angetriebene Klimaanlage
 - Quasi stationäre Anwendungen

Beispiel: Gesamtsimulation eines Hybrid-Elektrischen-Fahrzeugs

mit Allradantrieb



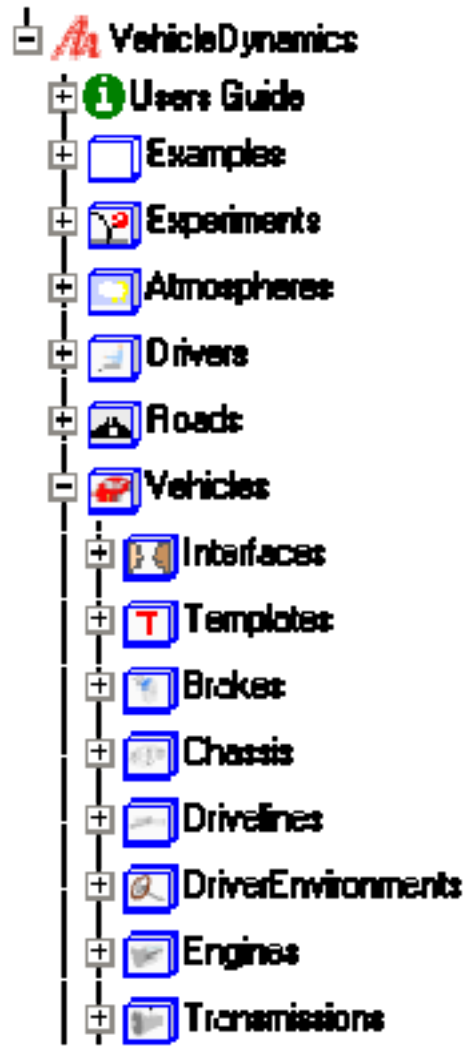
Vehicle Dynamics Library



Modelon

www.modelon.se

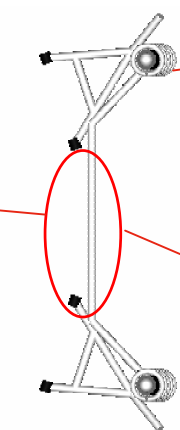
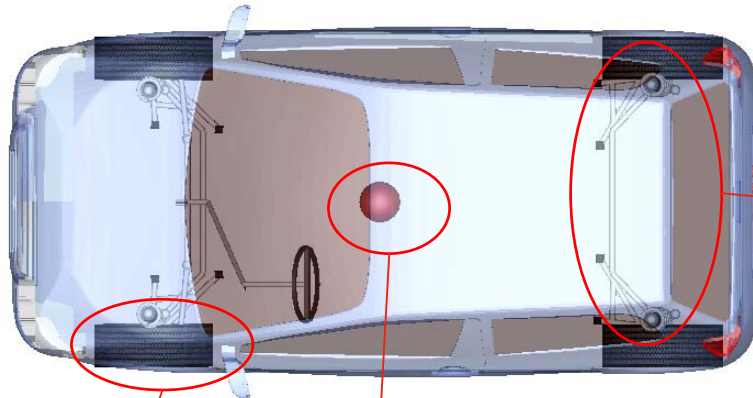
Vehicle Dynamics Library: Inhalt



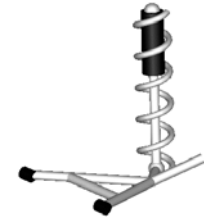
- große Anzahl von Chassis- und Fahrwerksmodellen
- Schnittstelle zu gängigen Reifenmodellen
- 3D-Straßenmodelle
- Fahrermodelle für geschlossenen und offenen Regelkreis
- elementarer Antriebsstrang
- Schnittstellen zur Powertrain und Transmission Library
- Motormodelle
- Bremssysteme
- Aerodynamik
- Beispiele und Beispielperimente

Vehicle Dynamics Library

Fahrwerk



Achse



Federbein



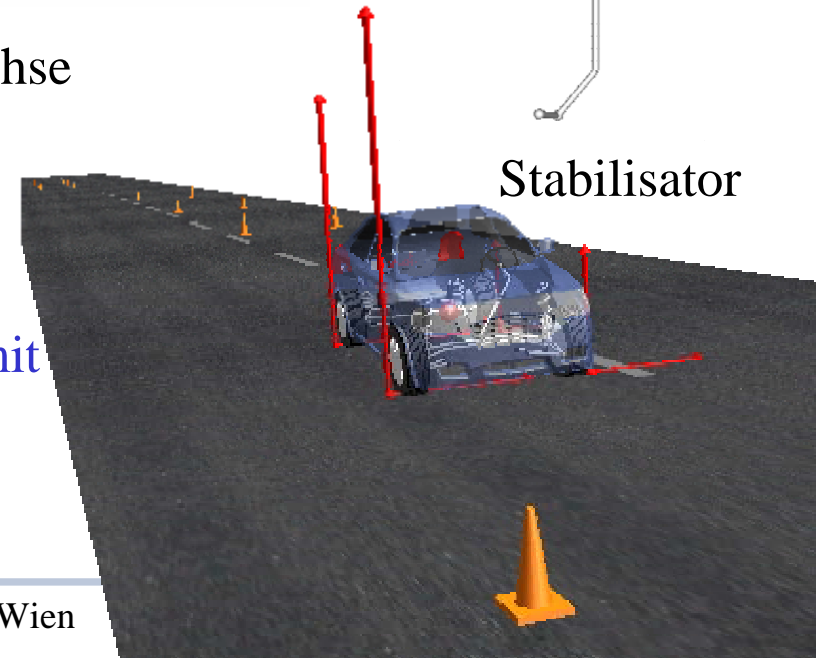
Stabilisator



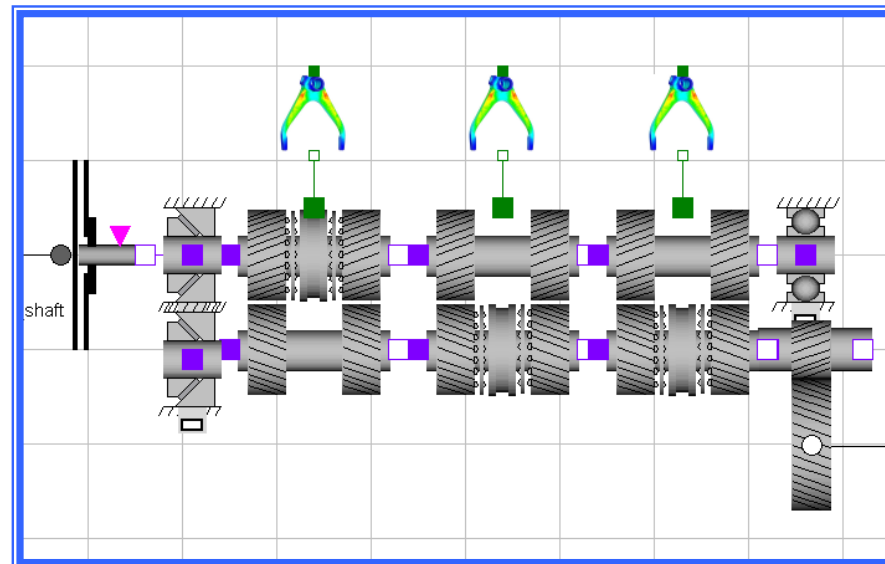
Rad mit Reifen

Aufbau

Experimentierumgebung mit Animation

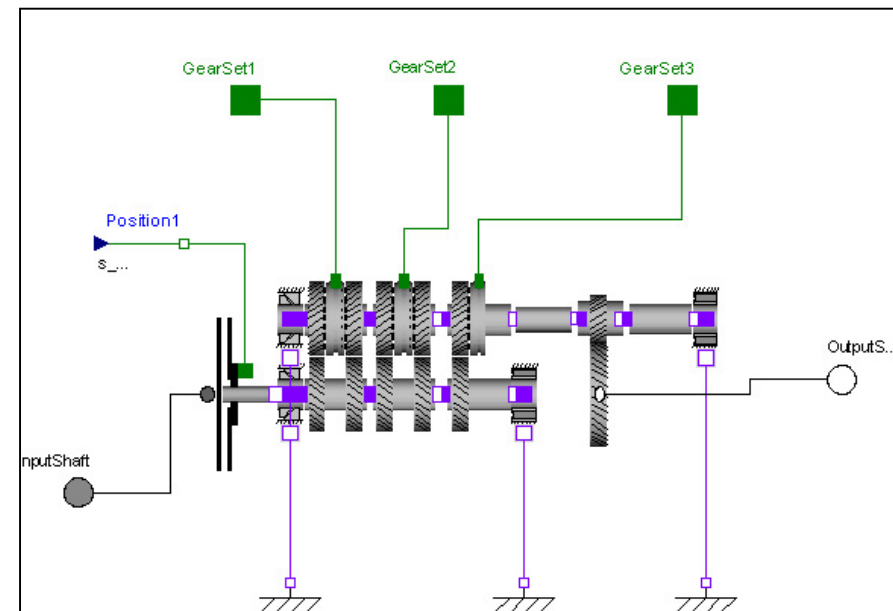


Transmission Library



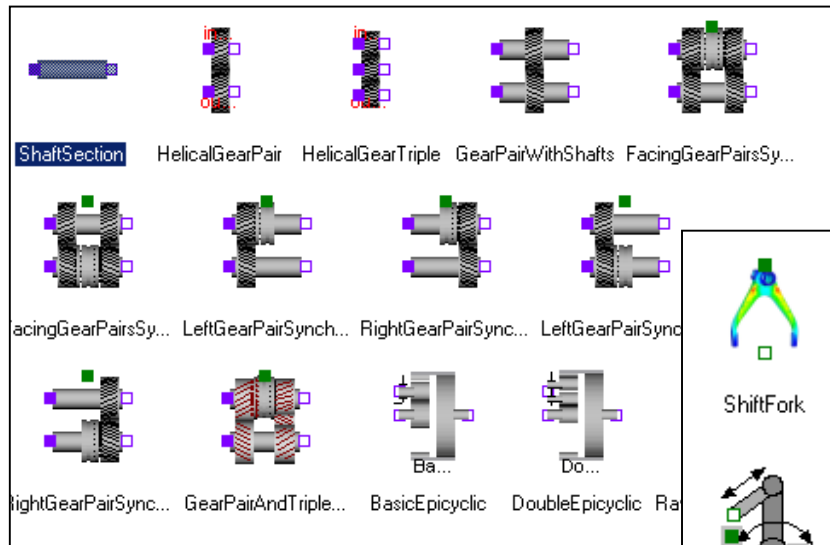
Transmission Library

- Detaillierte Modellierung von Antriebssträngen
- Ideal für Vorhersage der Schaltqualität und für das bessere Verstehen des Verhaltens des Antriebsstranges
- Axiale und rotatorische Freiheitsgrade

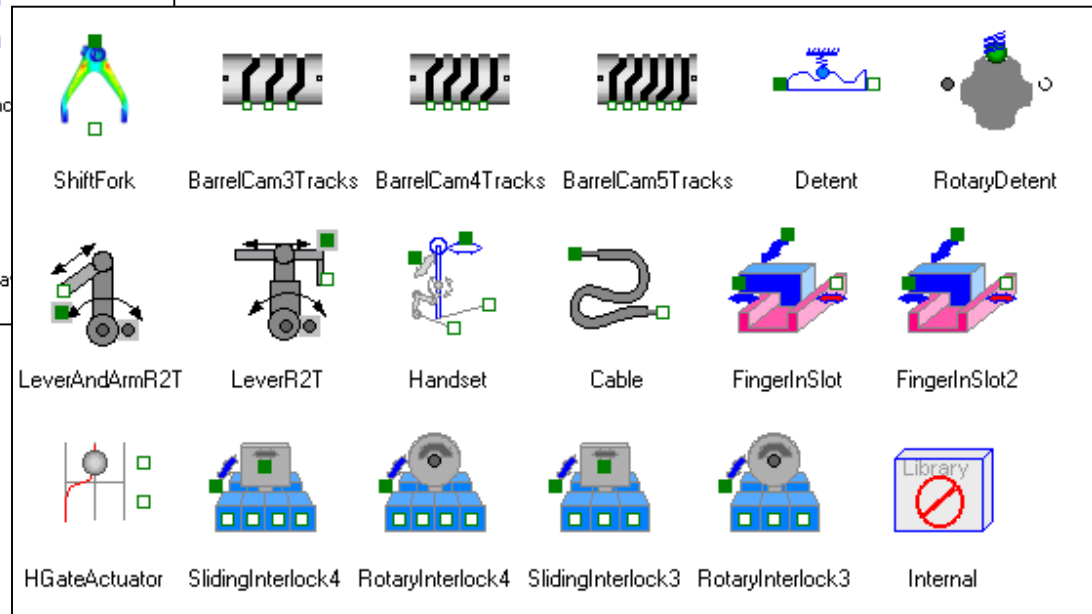


Beispiele der Komponenten

Achsen und Getriebe



Schaltmechanik



Literatur

- Principles of Object Oriented Modeling and Simulation with Modelica von Michael Tiller
2002, Kluwer Academic Publisher; ISBN 0-7923-7367-7
- Principles of Object-Oriented Modeling and Simulation with Modelica 2.1 von Peter Fritzson
2004, IEEE Press, John Wiley&Sons, Inc. ISBN 0_471-47163-1
- Objektorientierte Modellierung von Antriebssystemen von Martin Otter
Kapitel 20 in „Elektrische Antriebe – Regelung von Antriebssystemen“ von D. Schröder, 2. Auflage, 2001, Springer-Verlag
- Modelica 2.2 Specification, sh. www.modelica.org/documents
- Veröffentlichungen der Modelica Konferenzen der Jahre 200, 2002, 2003 und 2005 sh. www.modelica.org , dort unter den jeweiligen Konferenzen

Zusammenfassung

- Modelica soll (de-facto) Sprach-**Standard** werden
- Dymola (Software) 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.com
 - www.modelica.org
 - www.bausch-gall.de