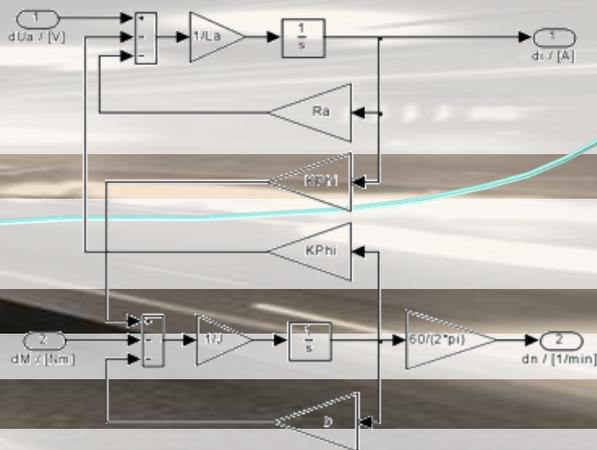
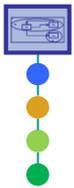


Institut für Regelungs- und Steuerungssysteme

Vorlesung: Modellbildung und Identifikation

Kapitel 2





Übersicht Vorlesungsinhalt (1)

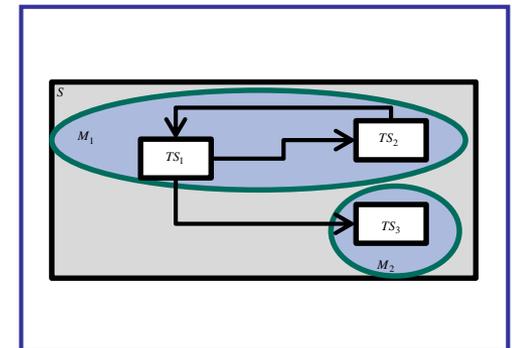
1. Einführung

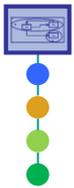
1. Motivation
2. Organisatorisches
3. Übersicht
4. Anwendung von Modellen
5. Klassifikation
6. Vorgehen bei der Modellbildung
7. Validierung und Verifikation



2. Strukturierung

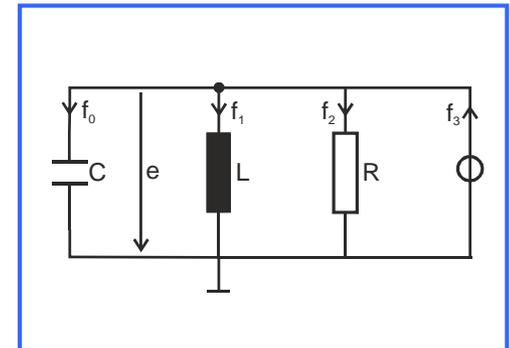
1. Abgrenzung
2. Strukturiertes System
3. Kausale Modellierung
4. Kopplungsanalyse
5. Strukturierung mit Matlab/Simulink
6. Objektorientierte Modellierung





3. Generalisierte Ersatzschaltbilder

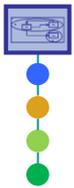
1. Motivation generalisierter Ersatzschaltbilder
2. Methode der generalisierten Variablen
3. Grundlegende Systemelemente
4. Methode der generalisierten Netzwerkanalyse



4. Variationsanalyse

1. Methode der Variationsanalyse
2. Aufstellen der Zustandsgleichungen

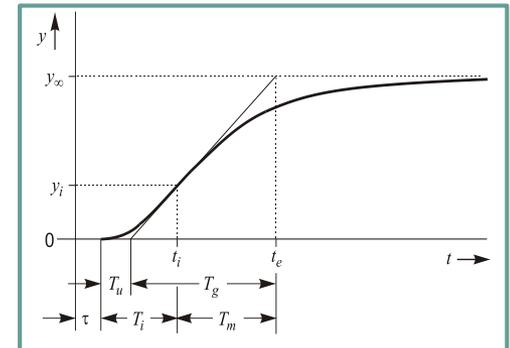
$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_{e_j}} \right) - \frac{\partial L}{\partial q_{e_j}} = F_j^{(Q)} - F_j^{(F)}$$
$$L = T^* - U$$



Übersicht Vorlesungsinhalt (3)

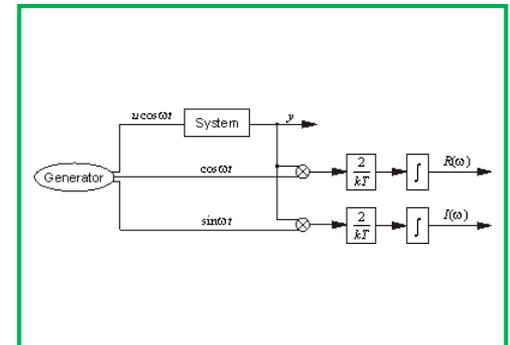
5. Identifikation mit parametrischen Modellen

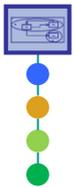
1. Übersicht
2. Kennwertermittlung
3. Least-Square-Verfahren für statische Prozesse
4. Least-Square-Verfahren für dynamische Prozesse
5. Generalized-Least-Square-Methode
6. Methode der Hilfsvariablen
7. Nichtlineare Methoden



6. Identifikation mit nichtparametrischen Modellen

1. Frequenzganganalyse
2. Korrelationsanalyse



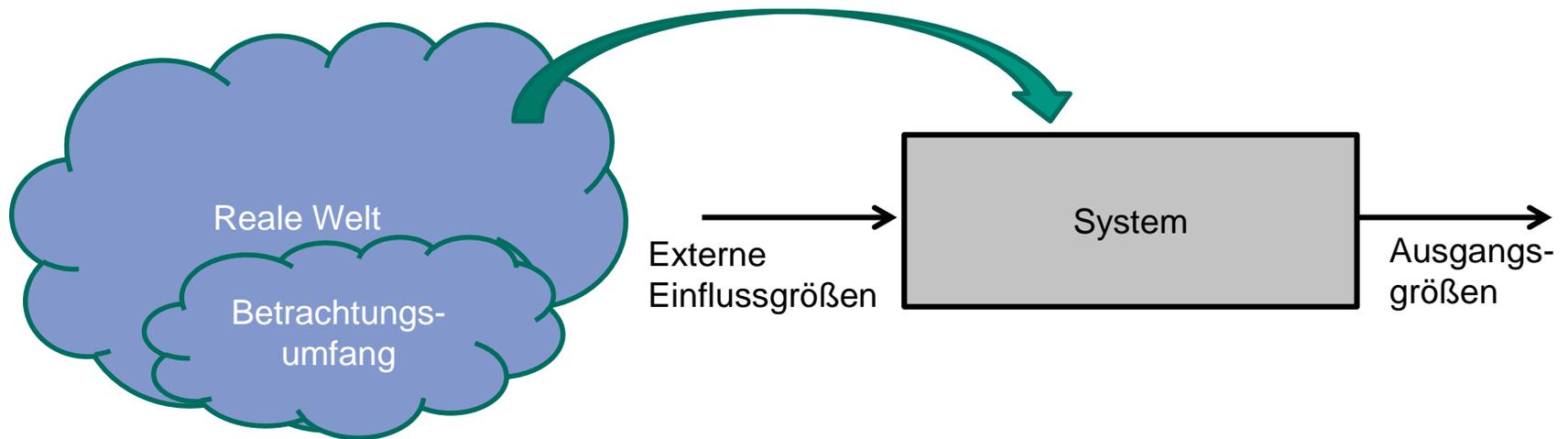


2.1 Abgrenzung

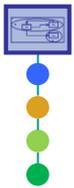
2.1.1 Grundidee

Die Abgrenzung erfolgt:

1. aufgrund der Definition der Aufgabenstellung bei einer gegebenen Anlage (Anforderungen, Lastenheft),
2. aufgrund der Definition eines Produktes (Anforderungen, Lastenheft)



Ergebnis: Definition von allen Eingangs- und Ausgangsgrößen
Festlegung des Wertebereichs
Definition der Zielstellung



2.1 Abgrenzung

2.1.2 Anforderungsdefinition

Anforderungsdefinition an ein Produkt erfolgt durch folgende Methoden

- Generalisierung
- Beispielszenarien
- Normative Szenarien
- Anforderungsbaum
- Maßgrößen

Ziel: Anforderungskatalog aus dem sich die Systemgrenzen ableiten lassen



Randbedingungen

- Häufig ist das Problem dem Kunden selbst unklar
- Bedürfnisse sind meist vage, ungenau und manchmal auch (technisch) falsch
- Anforderungen an ein Produkt sind damit oft unrealistisch

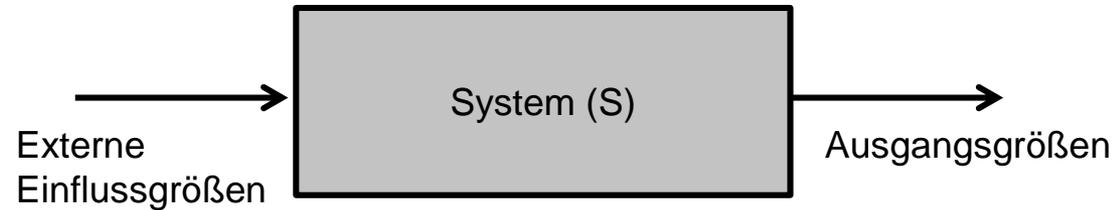
Wichtig: die Produktrandbedingungen und damit die Systemgrenzen können sich während des Entwurfs verschieben



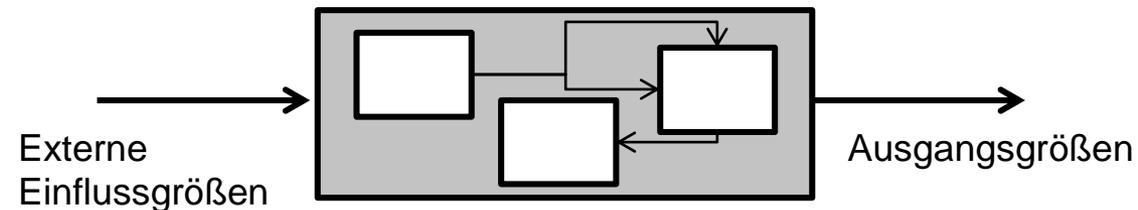
2.2 Strukturiertes System

2.2.1 Grundidee

Das abgegrenzte System besteht aus Teilsystemen/Teilelementen



Dekomposition  Aggregation 



- Welche Teilelemente betrachtet werden (bzw. werden müssen), hängt von der konkreten Aufgabenstellung ab.
- Ggf. ist nach erfolgtem modellbasierten Systementwurf eine Verfeinerung notwendig.



2.2 Strukturiertes System

2.2.2 Definition



Ein System ist durch folgende Axiome charakterisiert:

1. Strukturprinzip: Das System besteht aus einer Menge von Teilen, die untereinander und mit der Umgebung in wechselseitiger Beziehung stehen.
2. Dekompositionsprinzip: (Hierarchisierung) Das System besteht aus einer Menge von Teilen, die ihrerseits in Unterteile zerlegt werden können. Diese stehen in einer pyramidenförmigen Rangordnung.
3. Kausalprinzip: Das System besteht aus einer Menge von Teilen, deren Beziehung untereinander und deren Veränderung selbst eindeutig determiniert sind.
4. Temporalprinzip: Die Struktur und/oder der Zustand eines Systems unterliegen zeitlichen Änderungen.



2.2 Strukturiertes System

2.2.3 Methoden zur Strukturierung - Überblick

Folgende Methoden werden unterschieden:

- a) Signalbasierte Methoden → Matlab/Simulink
- b) Objektorientierte Methoden für Modellierung von physikalischen Systemen
- c) Funktionsbasierte Methoden → Strukturierte Analyse z.B mit case 4/0
- d) Objektorientierte Methoden für Softwaresysteme; UML (unified modeling language)

Anmerkungen:

- a) und c) werden vornehmlich für den Reglerentwurf verwendet
- b) hat den Fokus auf Simulation i.d.R. Reglerentwurf sehr schwierig
- d) hat den Fokus auf SW-Technik

Heutzutage sind alle Methoden durch entsprechende CAE-Tools gestützt.

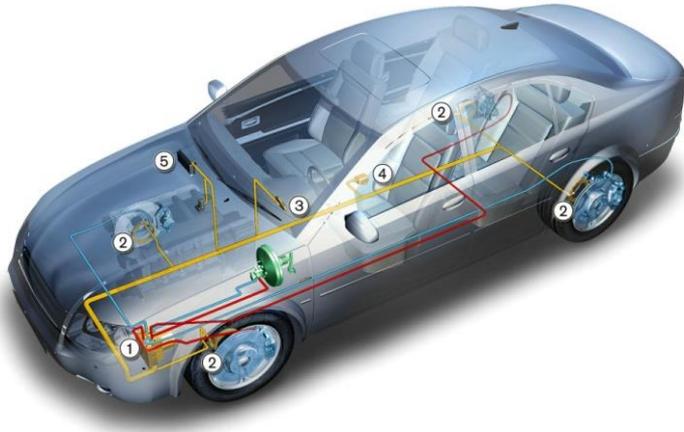


2.3

Kausale Modellierung

2.3.1

Beispiel: geregelter Gleichstrommotor eines ESP

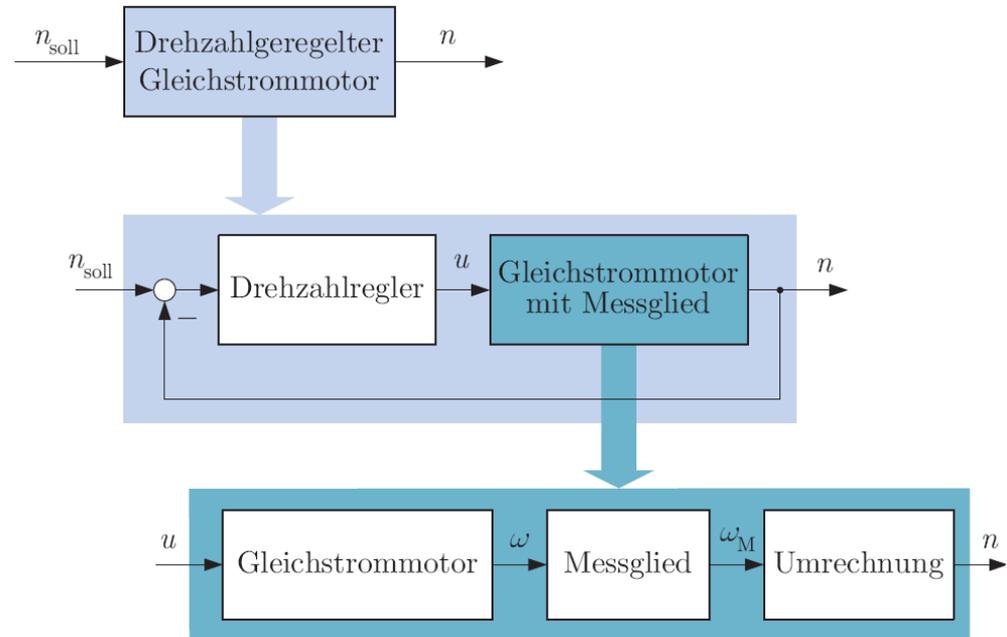


ESP-Komponenten:

1. ESP-Hydroaggregat mit integriertem Steuergerät
2. Raddrehzahlsensor
3. Lenkwinkelsensor
4. Drehratensensor mit integriertem Beschleunigungssensor
5. Steuergerät für Motormanagement zur Kommunikation



Quelle: Bosch



J. Lunze: Automatisierungstechnik, Oldenbourg-Wissenschaftsverlag 2007



2.3 Kausale Modellierung

2.3.2 Graphenbasierte Beschreibung

Definition: Gerichteter Graph

Ein gerichteter Graph ist ein Paar $G = (V, E)$ mit einer endlichen Menge von Knoten V und einer Menge von Kanten $E \subseteq V \times V$ (homogene Relation).

Definition: Blockschaltbild

Zur Modellierung von Systemen werden als Knoten Teilsysteme und als Kanten

1. Informations- (Signal-),
2. Stoff-, Masse-, Temperatur-
3. Energieflüsse

definiert.

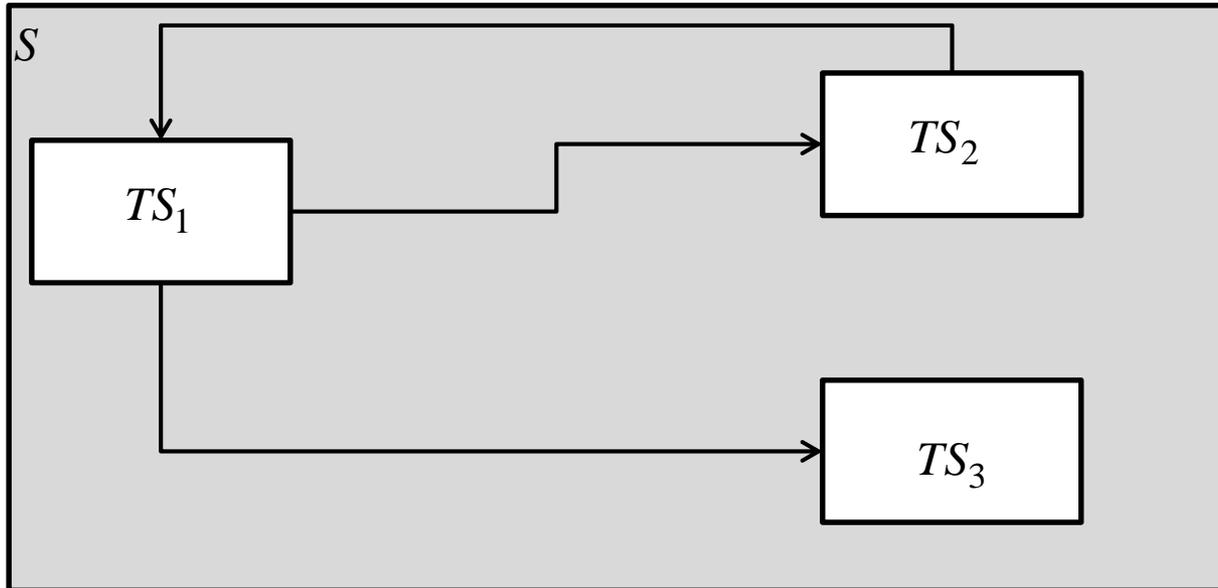
Vorteil:

Methoden der Graphentheorie können zur Analyse von Systemen herangezogen werden



2.3 Kausale Modellierung

2.3.2 Beispiel 1



$$G = (V, E)$$

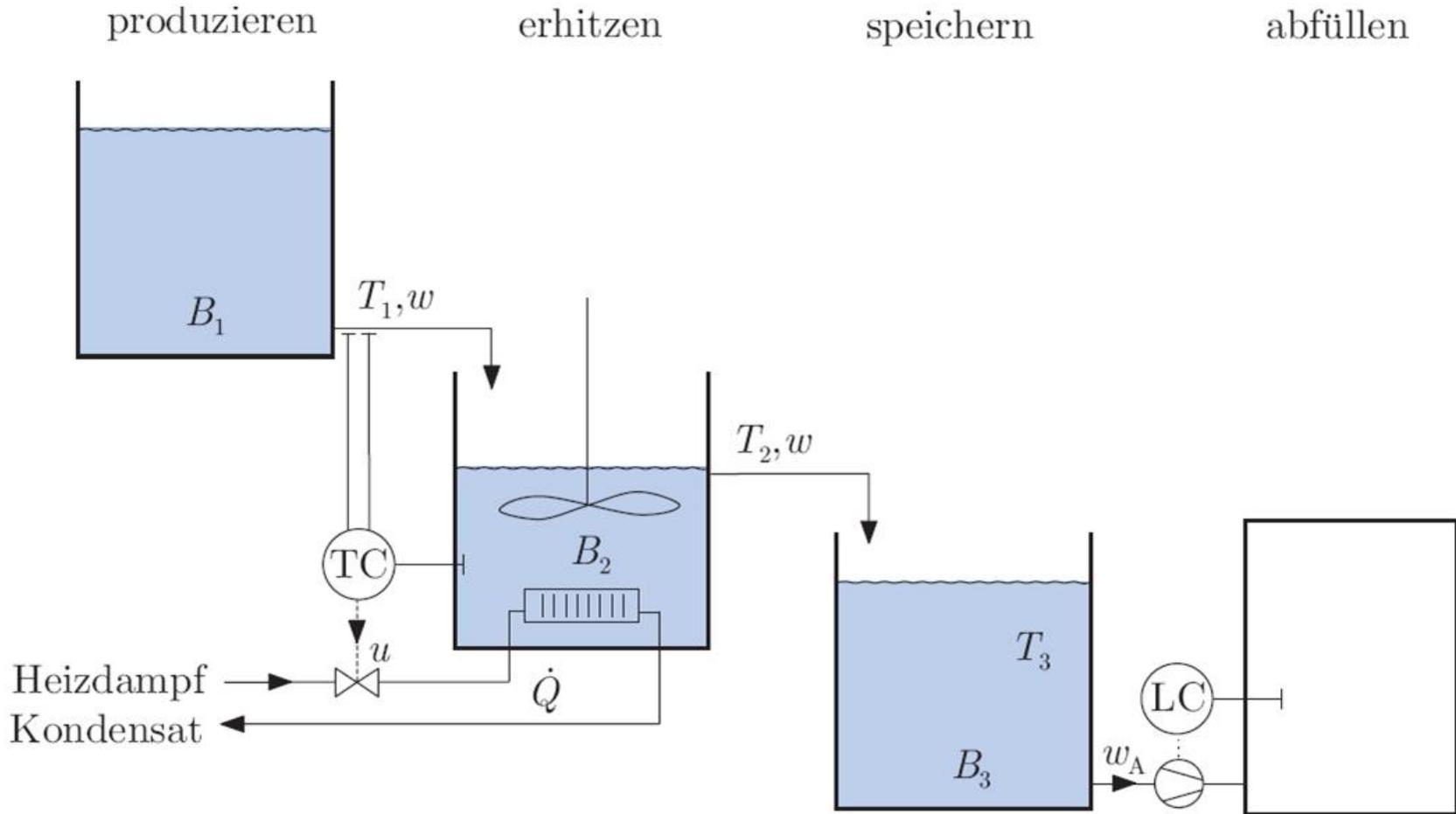
$$V = \{TS_1, TS_2, TS_3\}$$

$$E = \{(TS_1, TS_2), (TS_1, TS_3), (TS_2, TS_1)\}$$



2.3 Kausale Modellierung

2.3.2 Beispiel 2: Struktur einer Abfüllanlage (1)



J. Lunze: Automatisierungstechnik, Oldenbourg-Wissenschaftsverlag 2007



2.3

Kausale Modellierung

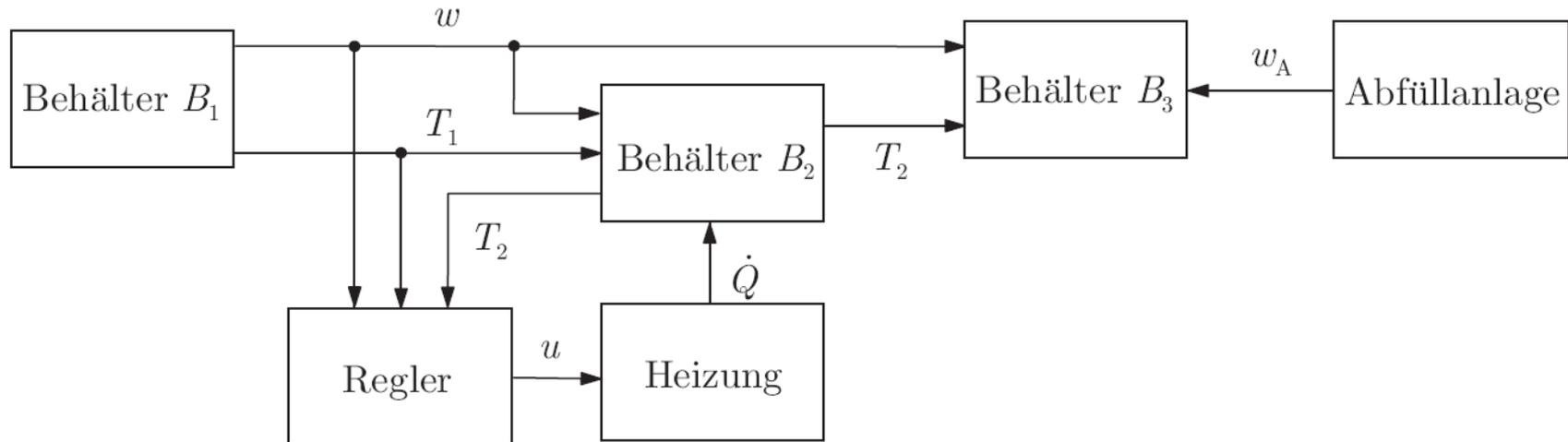
2.3.2

Beispiel 2: Struktur einer Abfüllanlage (2)



2.3 Kausale Modellierung

2.3.2 Beispiel 2: Struktur einer Abfüllanlage (3)



J. Lunze: Automatisierungstechnik, Oldenbourg-Wissenschaftsverlag 2007



2.4 Kopplungsanalyse

2.4.1 Zusammenhängende Teilsysteme

Definition: Stark zusammenhängende Teilsysteme

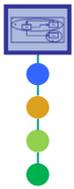
Die Teilsysteme i und j heißen stark zusammenhängend, wenn es im Blockschaltbild sowohl einen Pfad vom Teilsystem i zum Teilsystem j als auch einen Pfad vom Teilsystem j zum Teilsystem i gibt.

Konsequenzen:

Die Teilsysteme eines Blockschaltbildes lassen sich in disjunkten Mengen M_i zusammenfassen, wobei zwei Teilsysteme genau dann zu M_i gehören, wenn sie stark zusammenhängend sind.

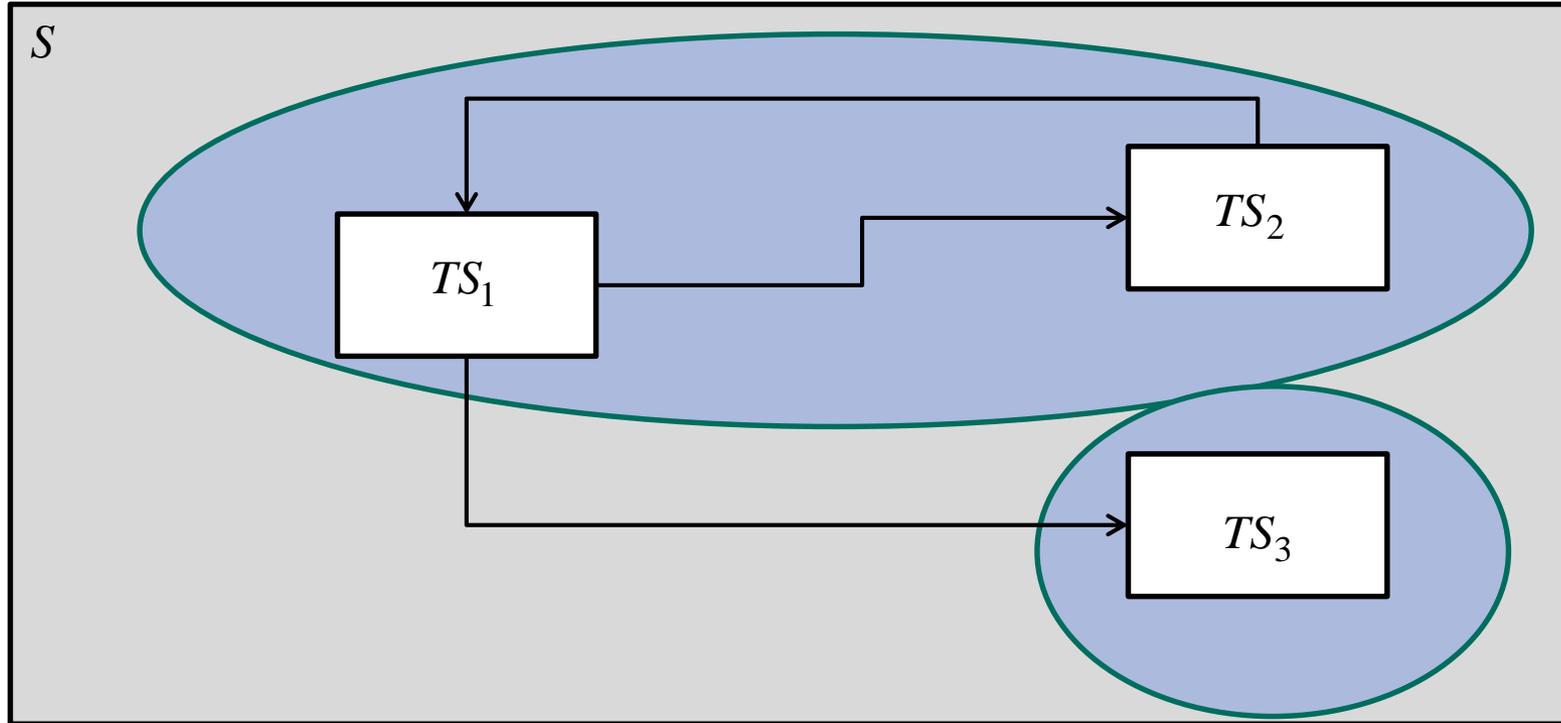
Fasst man nun diese Mengen M_i als neue (zusammengefasste) Teilsysteme (starke Komponenten) TMS_i auf, dann treten zwischen den TMS_i nur noch einseitige Kopplungen auf.

Die zusammengefassten Teilsysteme TMS_i können somit bei der Modellierung voneinander getrennt betrachtet werden.



2.4 Kopplungsanalyse

2.4.2 Beispiel

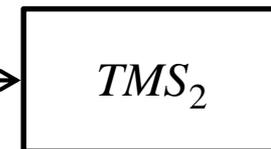


$$TMS_1 = M_1 = \{TS_1, TS_2\}$$

$$TMS_2 = M_2 = \{TS_3\}$$



Aggregation





2.4 Kopplungsanalyse

2.4.3 Bestimmung der starken Systemkomponenten

Ziel:

Mathematische Methode zur Bestimmung der stark zusammenhängenden Teilsysteme (starke Systemkomponenten).

Sinnvoll, wenn die Rückkopplungs- bzw. Vernetzungsstruktur komplex und unübersichtlich ist.

Vorgehen:

1. Bestimmung aller möglichen Nachfolgerpfade in einem Graphen
2. Bestimmung der Vereinigung aller möglichen Nachfolgerpfade
3. Berücksichtigung, dass ein Teilsystem selbst zu dem aggregierten System gehören kann
4. Bestimmung aller möglichen Vorgängerpfade in einem Graphen
5. Bestimmung der Vereinigung aller möglichen Vorgängerpfade sowie des einzelnen Teilsystems
6. Prüfung, welche Vorgängerpfade aus 3. und Nachfolgerpfade aus 5. übereinstimmen



2.4 Kopplungsanalyse

2.4.3 Bestimmung der starken Systemkomponenten

Kondensation G^K eines Graphen G :

Die **Kondensation** G^K eines gerichteten Graphen G stellt einen aus G gewonnenen reduzierten Graphen dar, dessen Knoten den starken Komponenten von G entsprechen und dessen Kanten den Zusammenhang zwischen den starken Komponenten untereinander wiedergeben.



2.4 Kopplungsanalyse

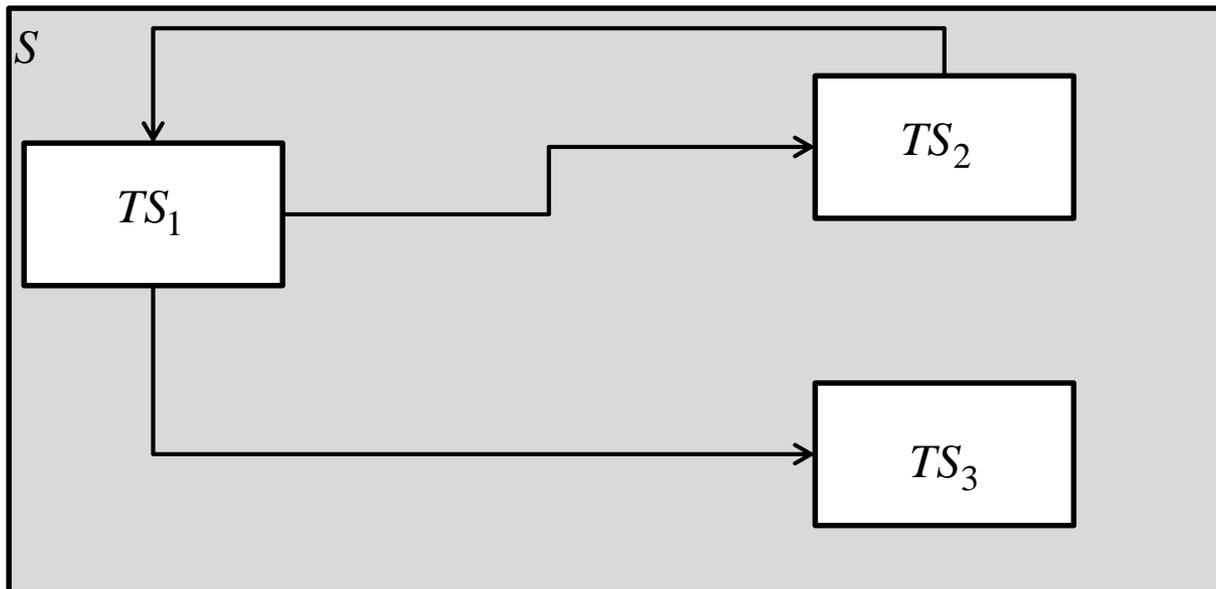
2.4.3 Bestimmung der starken Systemkomponenten

Analytische Beschreibung des Graphen:

Definition: Adjazenzmatrix

$$\underline{A}_{(n,n)} = [a_{ij}], \quad a_{ij} = \begin{cases} 1, & \text{falls Teilsystem } j \text{ Nachfolger von Teilsystem } i \\ 0 & \text{sonst} \end{cases}$$

Beispiel:

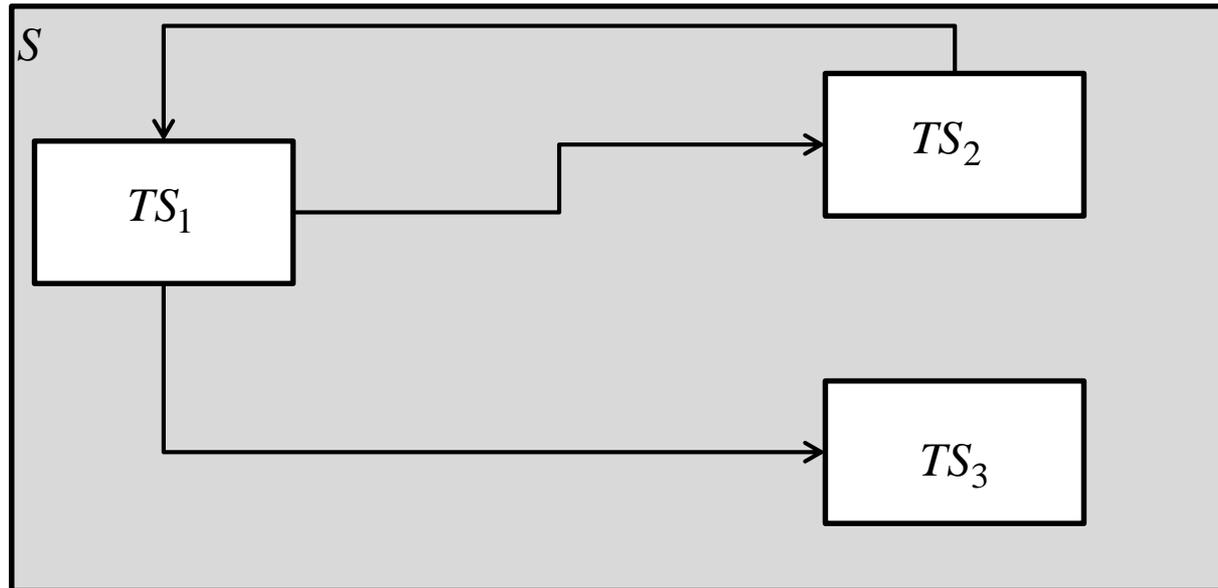


$$\underline{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



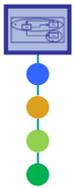
2.4 Kopplungsanalyse

2.4.3 Bestimmung der starken Systemkomponenten



1. Bestimmung aller möglichen Nachfolgerpfade $\underline{A}, \underline{A}^2, \dots, \underline{A}^k$ in einem Graphen mit n als längstem Pfad

$$\underline{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \underline{A}^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \left(\underline{A}^3 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)$$



2.4 Kopplungsanalyse

2.4.3 Bestimmung der starken Systemkomponenten

2. Bestimmung der Vereinigung aller möglichen Nachfolgerpfade (sog. transitive Hülle)

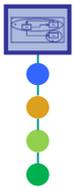
$$\underline{A}^+ := \bigcup_{k=1}^{\infty} \underline{A}^k = \bigcup_{k=1}^n \underline{A}^k = \underline{A} \cup \underline{A}^2 \cup \dots \cup \underline{A}^n$$

Beispiel: $\underline{A} \cup \underline{A}^2 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cup \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$

3. Berücksichtigung, dass ein Teilsystem selbst zu dem aggregierten System gehören kann (sog. reflexive transitive Hülle)

$$\underline{A}^* := \underline{A}^+ \cup \underline{I} = \underline{A} \cup \underline{A}^2 \cup \dots \cup \underline{A}^n \cup \underline{I}$$

Beispiel: $\underline{A}^* = \underline{A} \cup \underline{A}^2 \cup \underline{I} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$



2.4 Kopplungsanalyse

2.4.3 Bestimmung der starken Systemkomponenten

4. Bestimmung aller möglichen Vorgängerpfade in einem Graphen erhält man durch Transposition von \underline{A}

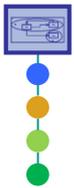
$$\underline{A}^T, (\underline{A}^2)^T, \dots, (\underline{A}^k)^T$$

5. Bestimmung der reflexiven transitiven Hülle der Vorgängerpfade

$$(\underline{A}^*)^T = (\underline{A} \cup \underline{A}^2 \cup \dots \cup \underline{A}^n \cup \underline{I})^T$$

6. Die starken Komponenten können folgendermaßen abgelesen werden

- Gilt für zwei Nebendiagonalelemente von \underline{A}^* die Bedingung $a_{ij}^* = a_{ji}^* = 1$, so gehören die Knoten i und j zu der gleichen starken Komponente.
- Ist lediglich das Hauptdiagonalelement $a_{ii}^* = 1$, gilt jedoch für die übrigen Nebendiagonalelemente $a_{ij}^* = 0$, so bildet der Knoten i eine eigene starke Komponente.
- Ist $a_{ij}^* = 1$ und $a_{ji}^* = 0$, so liegt in der Kondensation eine gerichtete Kante vor von der starken Komponente mit dem Knoten i zu der starken Komponente mit dem Knoten j .



2.4 Kopplungsanalyse

2.4.3 Bestimmung der starken Systemkomponenten

7. Algorithmus zur grafischen Bestimmung der Kondensation

1. Wahl eines Startknotens

Man wähle einen beliebigen Knoten $i \in V$ und kennzeichne ihn sowohl mit \oplus als auch mit \ominus .

2. Vorwärtsmarkierung

Man kennzeichne jeden Knoten j , der von i aus erreicht werden kann, d.h. für den ein Pfad $\rho = (i, \dots, j)$ mit $j \in V$ existiert, mit \oplus .

3. Rückwärtsmarkierung

Man kennzeichne jeden Knoten j , von dem aus i erreicht werden kann, d.h. für den ein Pfad $\rho = (j, \dots, i)$ mit $j \in V$ existiert, mit \ominus .

4. Alle diejenigen Knoten, die mit \oplus und \ominus gekennzeichnet sind, bilden eine starke Komponente von G .

5. Für die Knoten des Graphen, die nicht in bereits ermittelten starken Komponenten enthalten sind, sind die Schritte 1. - 4. beginnend mit einem beliebigen dieser Knoten zu wiederholen.

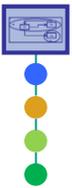
6. Zwei starke Komponenten sind dann durch eine Kante verbunden, wenn es mindestens eine Kante zwischen ihren jeweils enthaltenen Knoten gibt.



2.4 Kopplungsanalyse

2.4.4 Weitere Analysemöglichkeiten

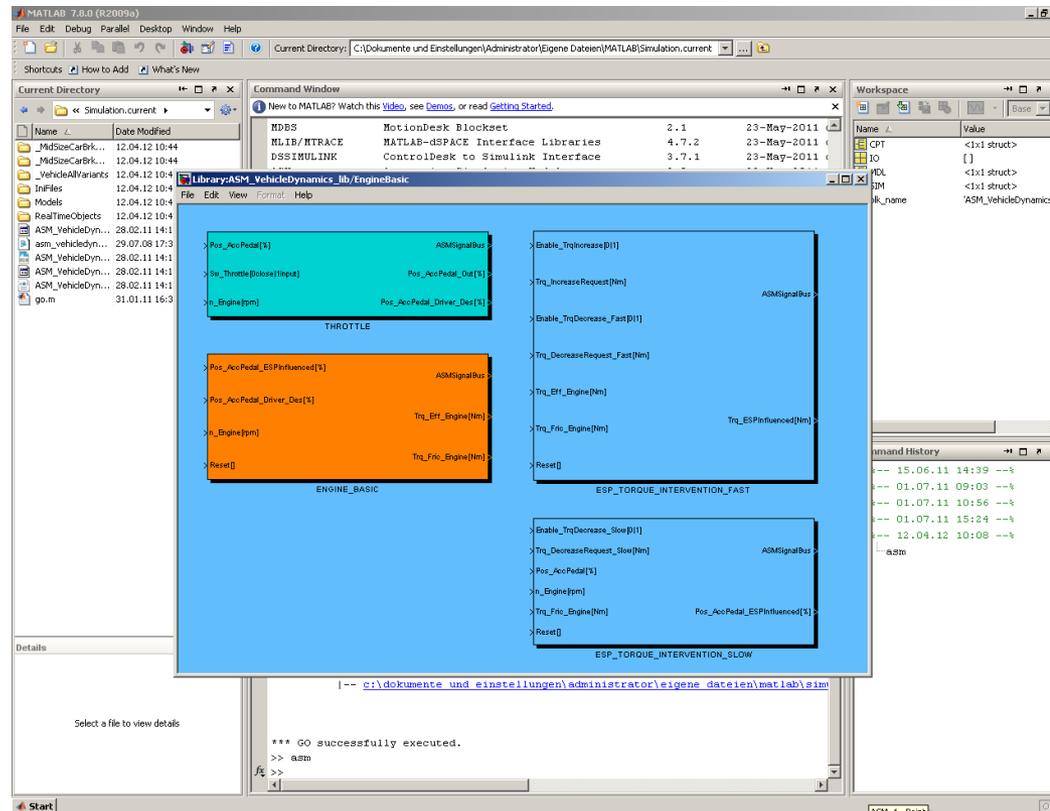
- auf Basis einer graphenbasierten Systembeschreibung können weitere Analysen durchgeführt werden
- diese Beschreibung liegt auf einem hohen Abstraktionslevel
- ein Spezialfall sind Automaten und Petrinetze
- weitergehende Informationen siehe AEH

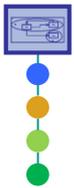


2.5 Strukturierung mit Matlab/Simulink

2.5.1 Modellierung in Matlab/Simulink (MS)

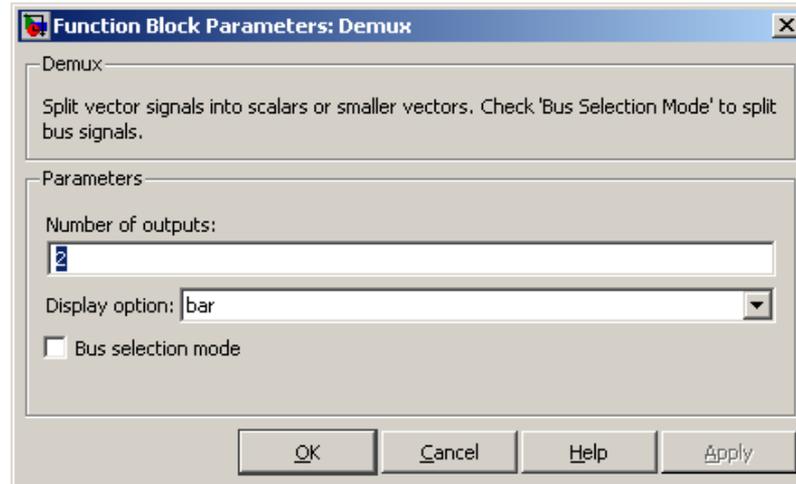
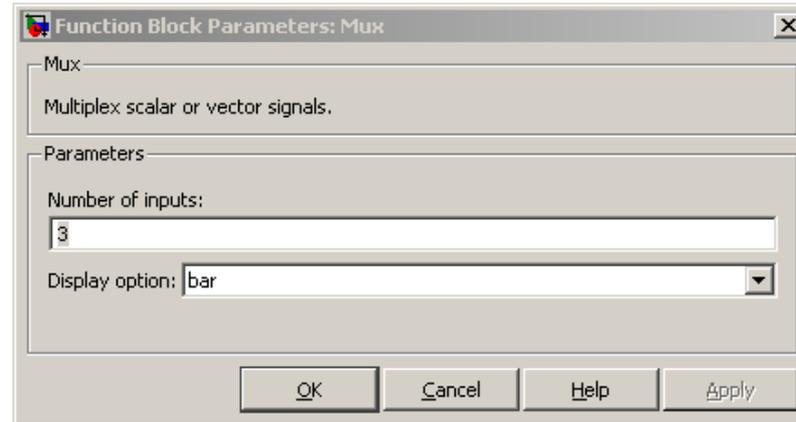
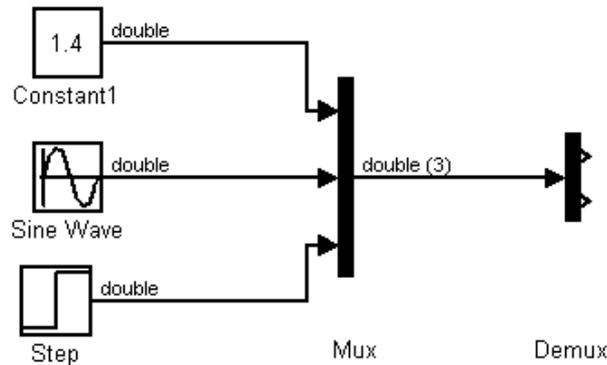
- Anlehnung an das Blockschaltbild aus SRT
- Modellbibliothek für Teilsysteme
- Kanten des Graphen sind Signale



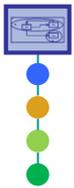


2.5 Strukturierung mit Matlab/Simulink

2.5.2 Signale in MS (1)

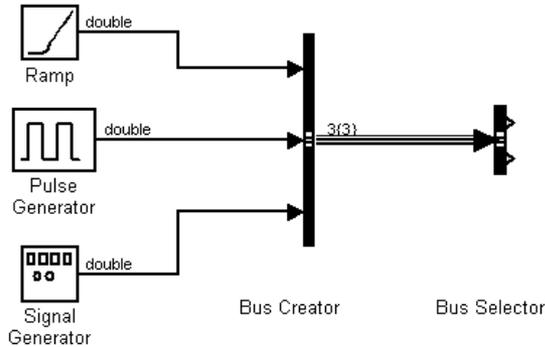


- Signale können mit individueller Namensbezeichnung versehen werden
- Signale können mehrdimensional definiert werden



2.5 Strukturierung mit Matlab/Simulink

2.5.2 Signale in MS (2)



Function Block Parameters: Bus Creator

BusCreator
This block creates a bus signal from its inputs.

Parameters

Inherit bus signal names from input ports

Number of inputs: 3

Signals in bus

- signal1
- signal2
- signal3

Rename selected signal:

Specify properties via bus object

Bus object: BusObject

Output as nonvirtual bus

Buttons: OK, Cancel, Help, Apply

Function Block Parameters: Bus Selector

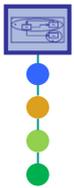
BusSelector
This block accepts a bus as input which can be created from a Bus Creator, Bus Selector or a block that defines its output using a bus object. The left listbox shows the signals in the input bus. Use the Select button to select the output signals. The right listbox shows the selections. Use the Up, Down, or Remove button to reorder the selections. Check 'Output as bus' to output a single bus signal.

Parameters

Signals in the bus	Find	Selected signals	Up
signal1	Select>>	signal1	Down
signal2	Refresh	signal2	Remove
signal3			

Output as bus

Buttons: OK, Cancel, Help, Apply



2.5 Strukturierung mit Matlab/Simulink

2.5.3 Signalgenerierung in MS

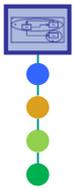
The diagram illustrates four different configurations of the Constant block in Simulink, each with its corresponding parameter dialog box:

- Block 1:** Constant value [1]. Output: 1.
- Block 2:** Constant value [1 2 3], Interpret vector param. as 1-D = on. Output: 3.
- Block 3:** Constant value [1 2 3], Interpret vector param. as 1-D = off. Output: 1x3 matrix.
- Block 4:** Constant value [1; 2; 3], Interpret vector param. as 1-D = on. Output: 3x1 matrix.

The parameter dialog boxes show the following settings:

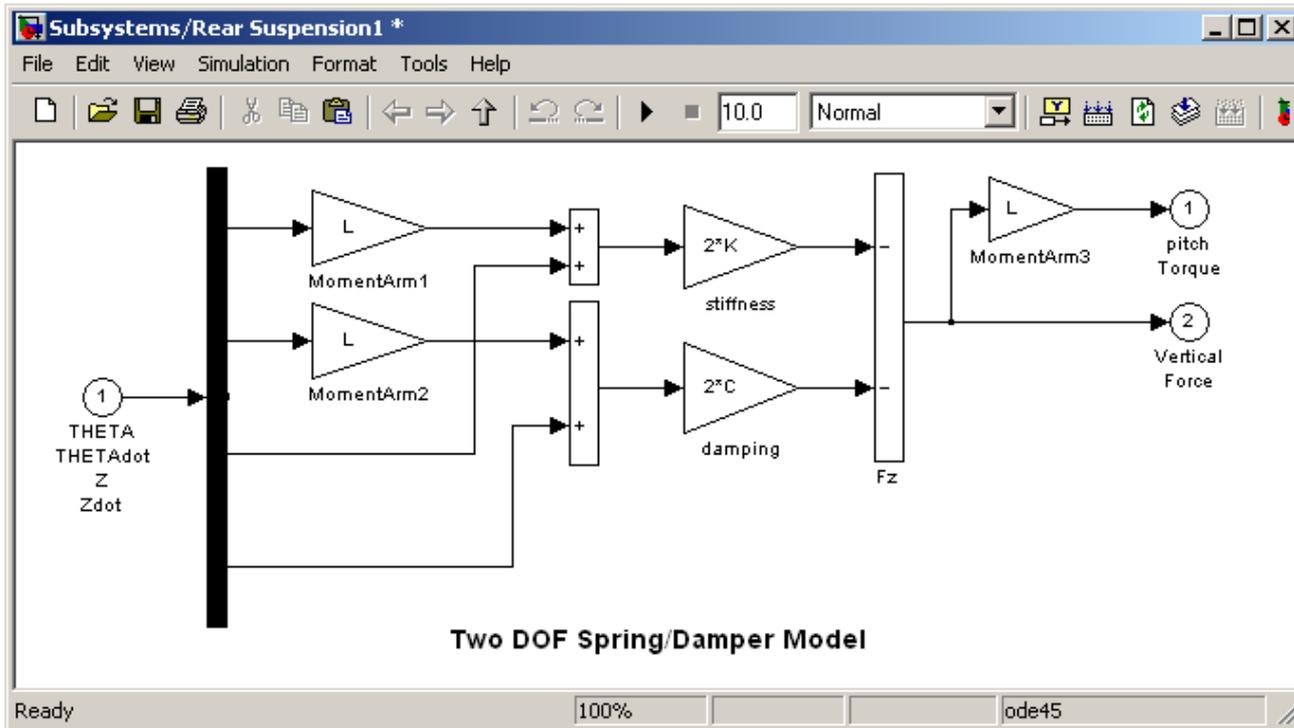
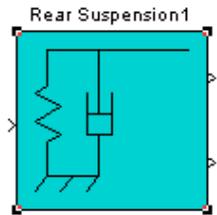
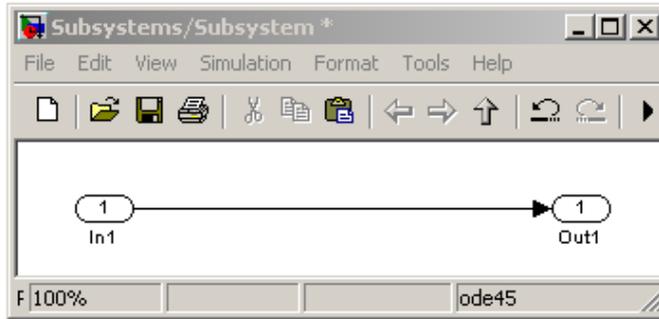
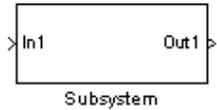
- Dialog 1:** Constant value [1; 2; 3], Interpret vector parameters as 1-D (checked).
- Dialog 2:** Constant value [1; 2; 3], Interpret vector parameters as 1-D (checked).
- Dialog 3:** Constant value [1 2 3], Interpret vector parameters as 1-D (unchecked).
- Dialog 4:** Constant value [1; 2; 3], Interpret vector parameters as 1-D (checked).

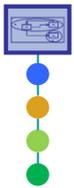
- Constant-Block:
 - zeitunabhängiges Signal
 - Zahlenwerte oder Variablen aus dem Workspace als Eingabewert möglich
 - Skalar, Vektor oder Matrix als Eingabewert möglich
 - Mit der Option „Interpret vector parameters as 1-D“ werden Zeilen- oder Spaltenvektor als Array weiterverarbeitet



2.5 Strukturierung mit Matlab/Simulink

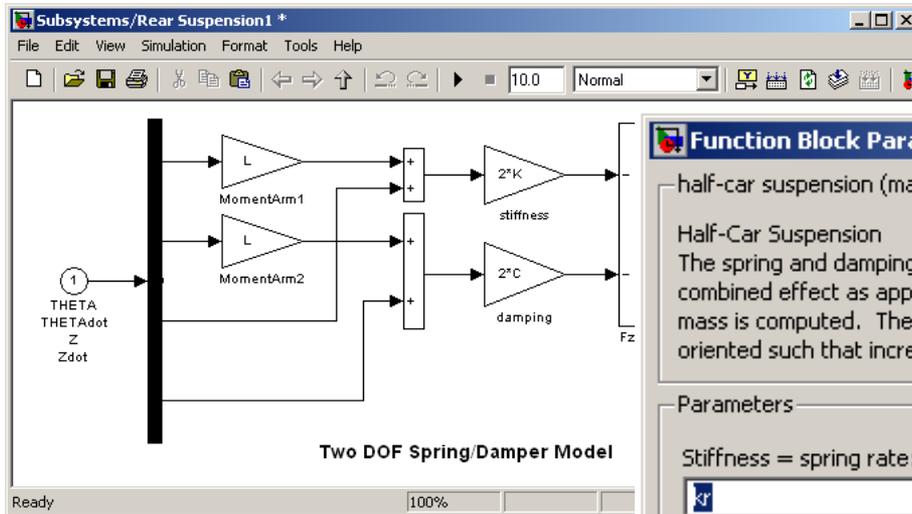
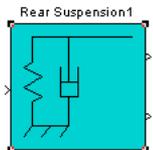
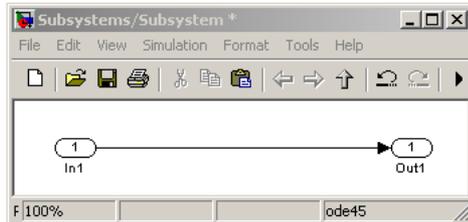
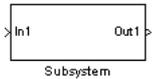
2.5.4 Teilsysteme in MS (1)





2.5 Strukturierung mit Matlab/Simulink

2.5.4 Teilsysteme in MS (2)



Function Block Parameters: Rear Suspension1

half-car suspension (mask)

Half-Car Suspension
The spring and damping rates for an individual wheel are entered below. Their combined effect as applied to two wheels at the specified distance from the center of mass is computed. The distance should be specified as positive if the subsystem is oriented such that increasing z corresponds to increasing theta, negative otherwise.

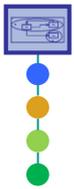
Parameters

Stiffness = spring rate:

Damping rate:

Moment arm:

Buttons: OK, Cancel, Help, Apply



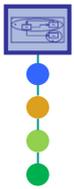
2.5 Strukturierung mit Matlab/Simulink

2.5.4 Teilsysteme in MS (3)

The screenshots show the Mask Editor for 'Rear Suspension1' in four different states:

- Top Left:** 'Icon' tab. Shows 'Icon options' (Visible, Transparency: Opaque, Rotation: Rotates, Units: Autoscale) and 'Drawing commands' (plot(-1, 0, 13, 12, [1 12], [11 11], [1 5], [2 ...]).
- Top Right:** 'Parameters' tab. Shows a table of dialog parameters.
- Bottom Left:** 'Initialization' tab. Shows 'Dialog variables' (K, C, L) and 'Initialization commands'.
- Bottom Right:** 'Documentation' tab. Shows 'Mask type' (half-car suspension), 'Mask description' (Half-Car Suspension. The spring and damping rates for an individual wheel are enter...), and 'Mask help' (See "Using Simulink and Stateflow in Automotive Applications" : ...).

Prompt	Variable	Type	Evaluate	Tunable
Stiffness = spring rate:	K	edit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Damping rate:	C	edit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Moment arm:	L	edit	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

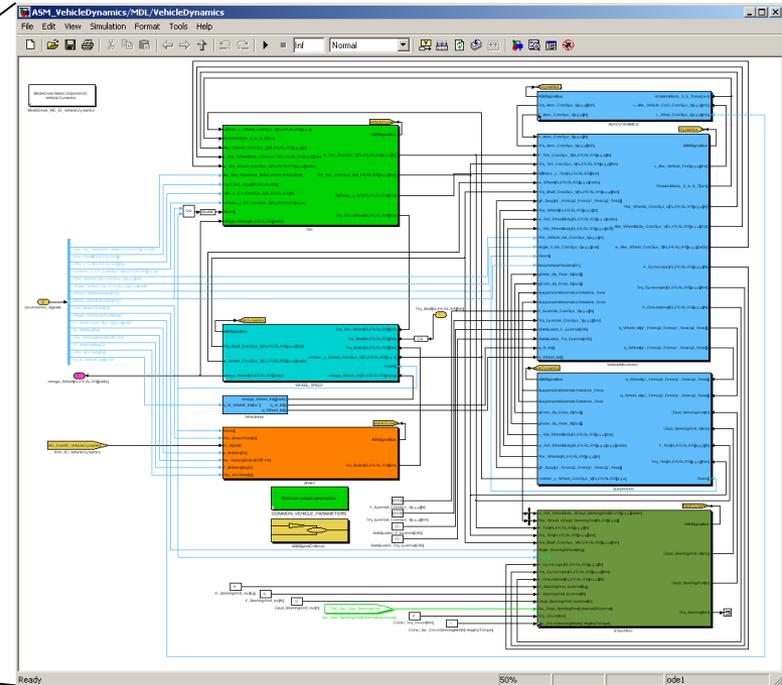
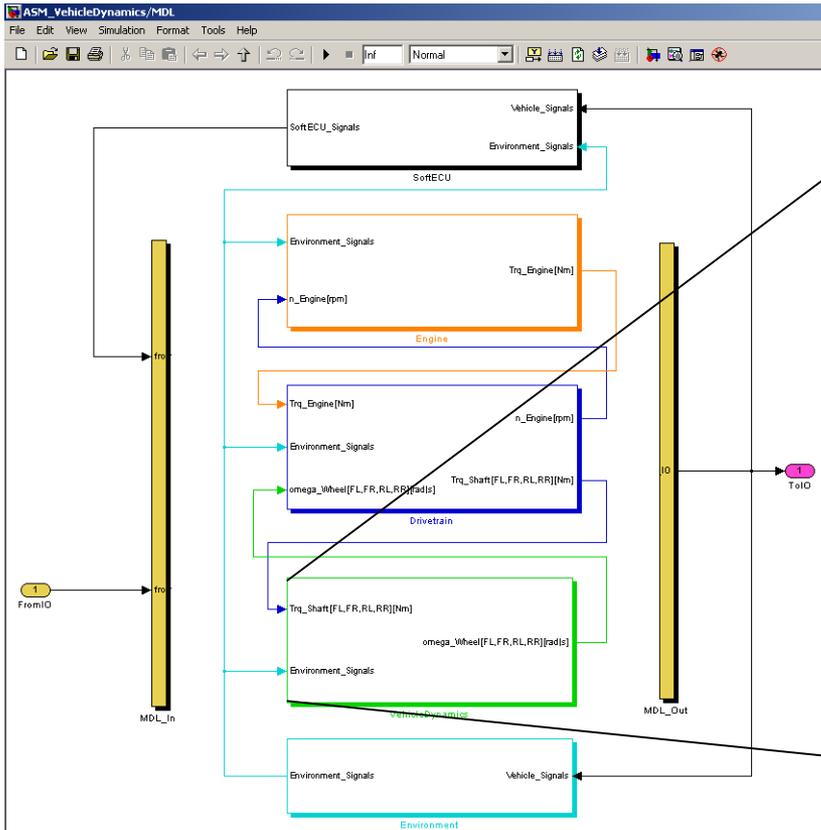


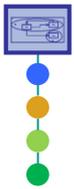
2.5

Strukturierung mit Matlab/Simulink

2.5.5

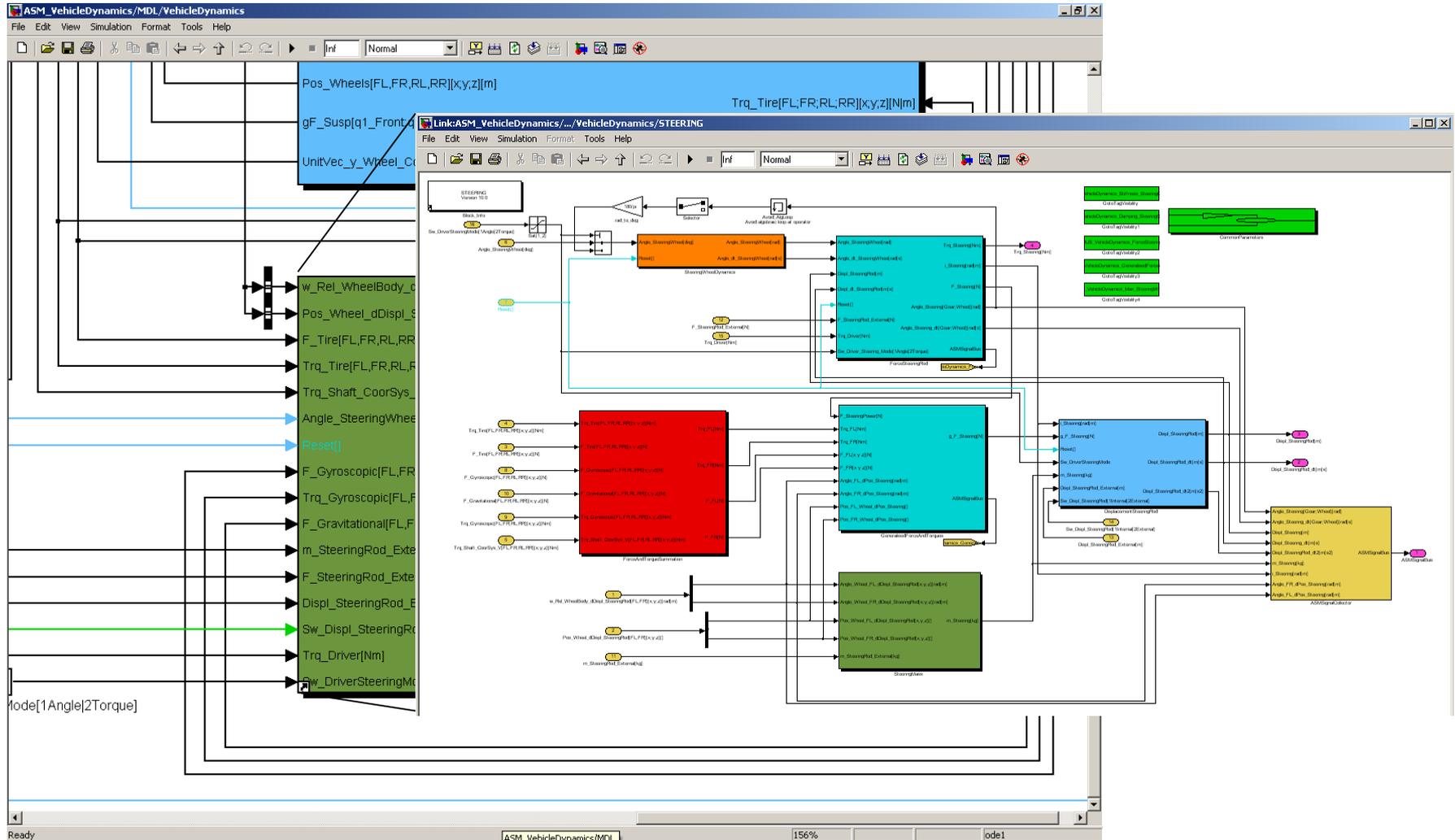
Beispiel: Fahrzeugmodell in MS (1)





2.5 Strukturierung mit Matlab/Simulink

2.5.5 Beispiel: Fahrzeugmodell in MS (2)





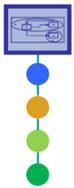
2.6

Objektorientierte Modellierung



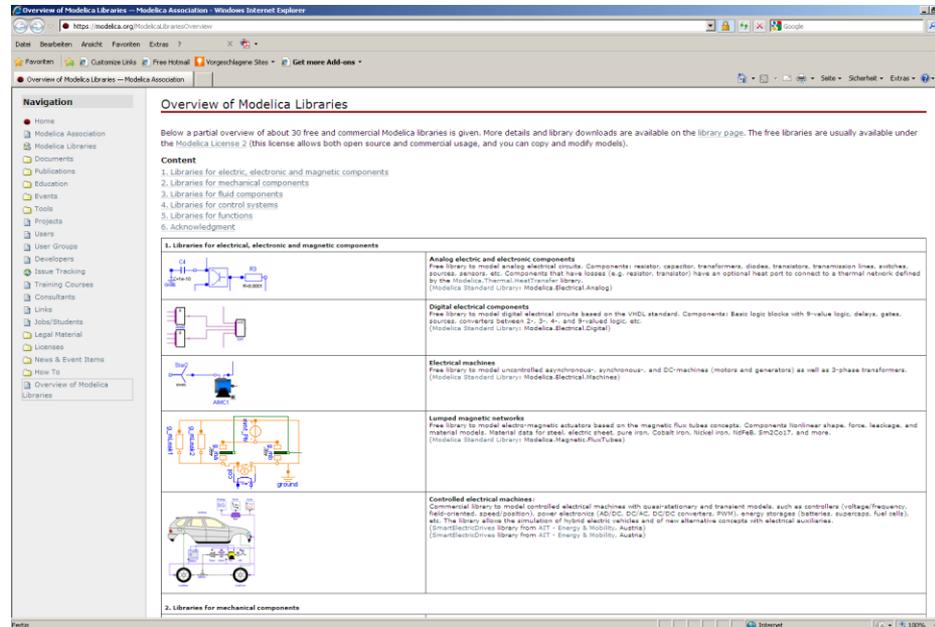
2.6.1

Akausale Modellierung



2.6 Objektorientierte Modellierung

2.6.2 Modelica

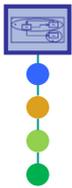


■ <https://modelica.org/ModelicaLibrariesOverview>

■ Verschiedene Entwicklungsumgebungen, z.B. <http://www.openmodelica.org/>

■ Beispiele für Modellierung und Simulation mit Modelica:

https://www.modelica.org/events/modelica2011/Proceedings/pages/tutorials/008/Tutorial_8_Multibody_Heckmann.pdf



2.6 Objektorientierte Modellierung

2.6.3 Dymola - Simulation

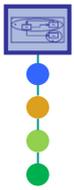
The screenshot shows the Dymola software interface. At the top, the window title is "Furuta - Furuta (Read-Only) - [Animation]". The menu bar includes "File", "Edit", "Simulation", "Plot", "Animation", "Commands", "Window", and "Help". The toolbar contains various icons for file operations, simulation control, and viewing options. Below the toolbar, there are playback controls (play, stop, previous, next) and a "Time: 5.18" display, along with a "Speed: 0.1" dropdown menu.

The main workspace displays a 3D model of a Furuta robot arm. The model consists of several colored links: a red base link, a green intermediate link, and a blue end effector link. A 3D coordinate system with x, y, and z axes is visible, with the origin at the base of the robot.

On the left side, there is a "Variable Browser" window. It has a tree view showing the following structure:

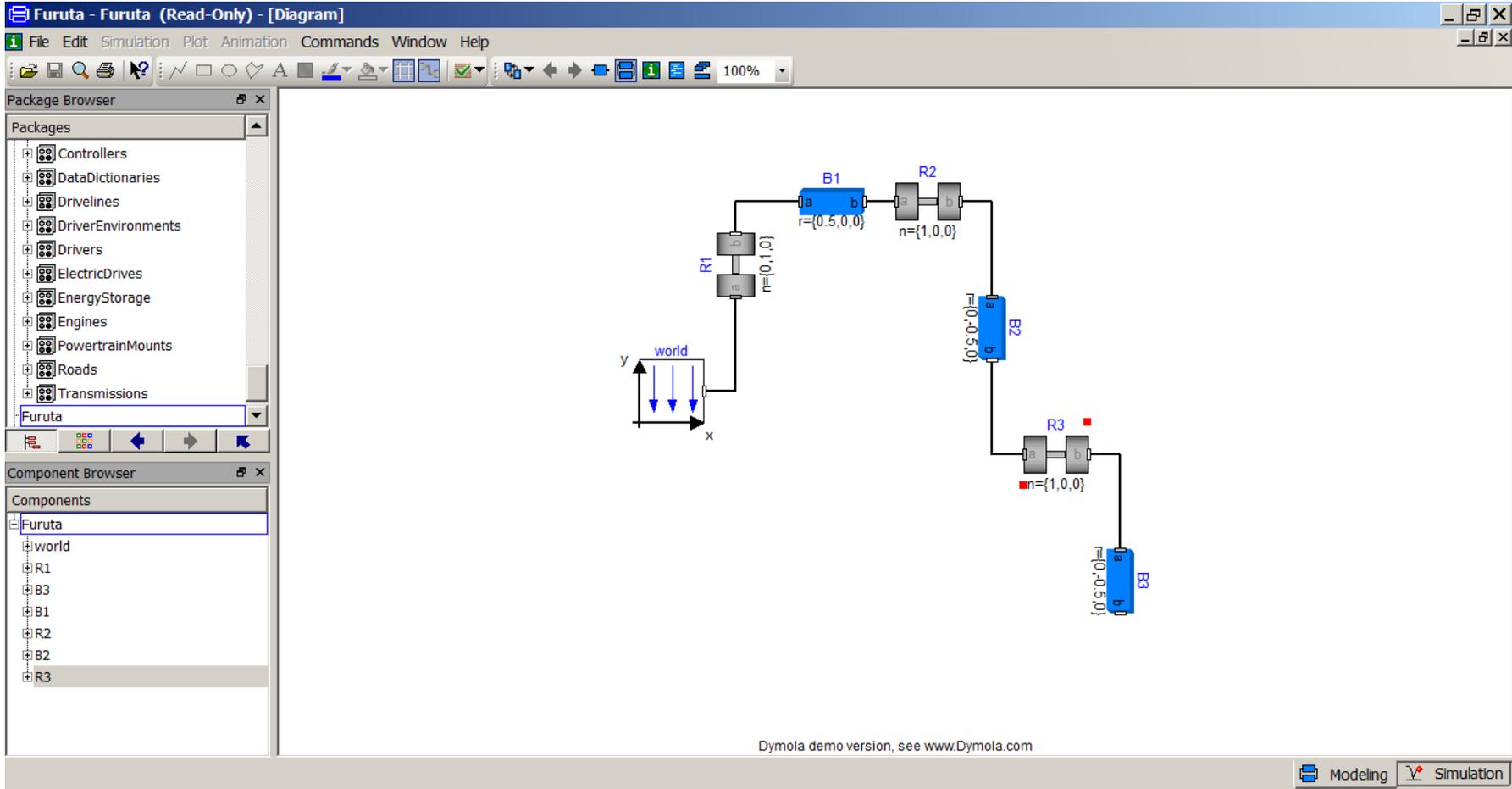
- dsres 1
 - world
 - R1
 - B3
 - B1
 - R2
 - B2
 - R3

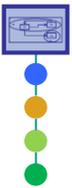
Demo-Download: <http://www.3ds.com/products/catia/portfolio/dymola/demo>



2.6 Objektorientierte Modellierung

2.6.3 Dymola - Modellierung





2.6 Objektorientierte Modellierung

2.6.3 Dymola - Quellcode

Modelica definition

```
model Furuta "Furuta pendulum with 3D animation"
```

```
inner Modelica.Mechanics.MultiBody.World world;  
Modelica.Mechanics.MultiBody.Joints.Revolute R1(  
  n={0,1,0},  
  a(fixed=false),  
  phi(fixed=true),  
  w(fixed=true));
```

```
Modelica.Mechanics.MultiBody.Parts.BodyBox B1(color={255,0,0}, r={0.5,0,0});
```

```
Modelica.Mechanics.MultiBody.Joints.Revolute R2(  
  n={1,0,0},  
  a(fixed=false),  
  w(fixed=true),  
  phi(  
    fixed=true,  
    start=2,  
    displayUnit="rad"));
```

```
Modelica.Mechanics.MultiBody.Parts.BodyBox B2(color={0,180,0}, r={0,-0.5,0});
```

```
Modelica.Mechanics.MultiBody.Joints.Revolute R3(  
  n={1,0,0},  
  a(fixed=false),  
  phi(fixed=true),  
  w(fixed=true));
```

```
Modelica.Mechanics.MultiBody.Parts.BodyBox B3(color={0,0,255}, r={0,-0.5,0});
```

```
equation
```

```
connect(R3.frame_b, B3.frame_a);  
connect(world.frame_b, R1.frame_a);  
connect(R1.frame_b, B1.frame_a);  
connect(B1.frame_b, R2.frame_a);  
connect(R2.frame_b, B2.frame_a);  
connect(B2.frame_b, R3.frame_a);
```

```
end Furuta;
```

