

Matlab/Simulink- Kurs Teil III

Übungen zum Modul Mechatronik im SoSe 2015

Prof. Dr.-Ing. Gerd-J. Menken
(g.menken@hs-bremen.de)

&

Dipl.-Ing. Ralf Blechschmidt
(ralf.blechschmidt@hs-bremen.de)



1 Umgang mit Modellparametern

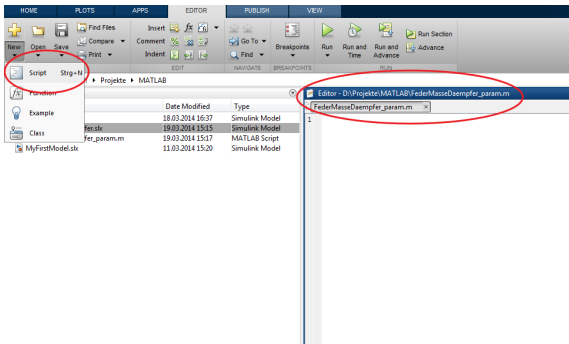
2 Übertragungsfunktionen & Zustandsraum

Umgang mit Modellparametern

- die Parameter der einzelnen Blöcke eines Modells werden als **Systemparameter** bezeichnet
- Idee: alle Systemparameter werden als *MATLAB*-Variablen angelegt
- Zuordnung von Zahlenwerten erfolgt dann in einer *MATLAB*-Skript-Datei (*.m)
- Festlegung: einheitliche Konvention für Dateinamen (z.B. *MyFirstModel_param.m*)
- in der *MATLAB*-Skript-Datei können auch Kommentare zur Dokumentation untergebracht werden
- zu einem Modell können mehrere Parameterdateien gehören

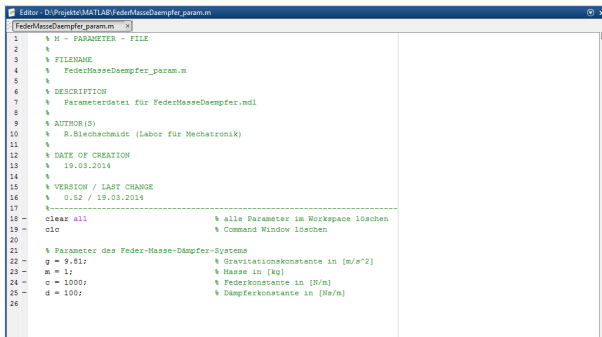
Beispiel: Feder–Masse–Dämpfer–System

- neue Script–Datei in *MATLAB* anlegen
- beim Speichern Namenskonvention beachten



Beispiel: Feder-Masse-Dämpfer-System

- alle Systemparameter aufführen
- übersichtliche Darstellung wählen
- sinnvolle Kommentare (z.B. Autor, Einheiten der Parameter)
- oftmals sind auch Berechnungen sinnvoll



```
Editor - D:\Projekte\MATLAB\FederMasseDaempfer_param.m
[FederMasseDaempfer_param.m]
1 % M - PARAMETER - FILE
2 %
3 % FILENAME
4 % FederMasseDaempfer_param.m
5 %
6 % DESCRIPTION
7 % Parameterdatei für FederMasseDaempfer.mdl
8 %
9 % AUTHOR(S)
10 % R.Bleeschmidt (Labor für Mechatronik)
11 %
12 % DATE OF CREATION
13 % 19.03.2014
14 %
15 % VERSION / LAST CHANGE
16 % 0.52 / 19.03.2014
17 %-----
18 - clear all % alle Parameter im Workspace löschen
19 - clc % Command Window löschen
20
21 % Parameter des Feder-Masse-Dämpfer-Systems
22 g = 9.81; % Gravitationskonstante in [m/s^2]
23 m = 1; % Masse in [kg]
24 c = 1000; % Federkonstante in [N/m]
25 d = 100; % Dämpferkonstante in [Ns/m]
26
```

Beispiel: Feder–Masse–Dämpfer–System

- Script-Datei unter *MATLAB* ausführen
- Systemparameter werden im Workspace gespeichert
- *Simulink*-Modell greift auf Workspace-Variablen zu

The screenshot illustrates the workflow of loading parameters into the workspace for use in a Simulink model. The top toolbar shows the 'Run' button (a green play icon) circled in red, with a red arrow pointing to the 'Workspace' window. Below the toolbar, the text 'Skript-Datei ausführen' is displayed in red. The 'Editor' window shows a script file named 'FederMasseDaempfer_param.m' with the following content:

```

1 % SI - PARAMETER - FILE
2 %
3 % FILENAME
4 % FederMasseDaempfer_param.m
5 %
6 % DESCRIPTION
7 % Parameterdatei für FederMasseDaempfer.m
8 %
9 %
10 % AUTHOR(S)
11 % W.Bleichschmidt (Labor für Mechatronik)
12 %
13 % DATE OF CREATION
14 % 19.03.2014
15 %
16 % VERSION / LAST CHANGE
17 % 0.52 / 19.03.2014
18 %
19 %
20 %
21 % Parameter des Feder-Masse-Dämpfer-System
22 - g = 9.81; % Grav.
23 - m = 1; % Masse
24 - c = 1000; % Feder
25 - d = 10; % Dämpf
26

```

The 'Workspace' window shows the variables loaded from the script:

Name	Value	Min	Max
c	1000	1000	1000
d	10	10	10
g	9.8100	9.8100	9.8100
m	1	1	1

The 'FederMasseDaempfer' Simulink model is shown in the foreground. It features a 'Constant' block for gravity (g), an 'Integrator' block for displacement (x_p(t)), and another 'Integrator' block for velocity (x(t)). The displacement signal is fed into a 'Scope' block. The velocity signal is fed into two gain blocks labeled 'Dämpferkonstante' (d/m) and 'Federkonstante' (c/m). The 'Command Window' at the bottom left shows the prompt 'A >>'.

Beispiel: Feder–Masse–Dämpfer–System

- *Simulink*–Modell parametrieren und ausführen
- Ergebnis verifizieren

The screenshot illustrates the process of running a Simulink model. The main window shows the Simulink model for a mass-spring-damper system. The model consists of a constant input, a summing junction, an integrator, a gain block for the damper constant (d/m), another summing junction, another integrator, a gain block for the spring constant (c/m), and a scope to observe the output $x(t)$.

The parameter editor window shows the following parameters:

```

% Parameter des Feder-Masse-Dämpfer-Systems
%
% FILENAME
% FederMasseDaempfer
%
% DESCRIPTION
% Parameterdatei
%
% AUTHOR(S)
% S. Bleschschmidt
%
% DATE OF CREATION
% 19.03.2014
%
% VERSION / LAST CHANGED
% 0.52 / 19.03.2014
%
% clear all
% cdc
%
% Parameter des Feder-Masse-Dämpfer-Systems
%
% g = 9.81;
% m = 1;
% c = 1000;
% d = 10;
  
```

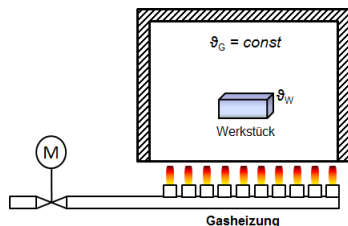
The workspace window shows the following variables:

Name	Value	Min	Max
c	1000	1000	1000
d	10	10	10
g	9.8100	9.8100	9.8100
m	1	1	1
test	<100hd double>	1.000e-1	

The scope window displays the time response of the system, showing a damped oscillation starting from zero and settling around a steady-state value of approximately 0.01.

Beispiel: Aufheizen eines Werkstückes im Glühofen

- Werkstück hat homogene Temperaturverteilung
- Temperatur im Ofen wird konstant gehalten
- Werkstück wird zum Zeitpunkt $t = 0s$ in den Ofen eingebracht



Übertragene Wärmeleistung:

$$\frac{c \cdot m}{\alpha \cdot A} \cdot \dot{\vartheta}_W + \vartheta_W = \vartheta_G, \quad T = \frac{c \cdot m}{\alpha \cdot A}$$

- 1) Erstellen Sie für das Feder–Masse–Dämpfer-Modell eine Parameterdatei und passen Sie das *Simulink*–Modell entsprechend an.
- 2) Erstellen Sie für das Aufheizen eines Werkstücks ein *Simulink*–Modell mit entsprechender Parameterdatei. Verwenden Sie folgende Systemparameter:

$$m = 80 \text{ kg}$$

Werkstückmasse

$$c = 0.6 \text{ kJ}/(\text{kgK})$$

spezifische Wärmekapazität

$$A = 0.646 \text{ m}^2$$

Werkstückoberfläche

$$\alpha = 12 \text{ W}/(\text{m}^2 \text{ kg})$$

Wärmeübergangskoeffizient

$$\vartheta_G = 700^\circ \text{ C}$$

Ofentemperatur

$$\vartheta_{W,t=0} = 20^\circ \text{ C}$$

Anfangstemperatur Werkstück

Übertragungsfunktionen & Zustandsraum

Übertragungsfunktionen in *Simulink*

- Differentialgleichungen werden mittels *Laplace*-Transformation in Übertragungsfunktionen überführt
- viele Analyseverfahren vorhanden (z.B. für Stabilität)
- viele Verfahren für Reglerentwurf bekannt
- einfache Modellierung in *Simulink*

Beispiel: System zweiter Ordnung

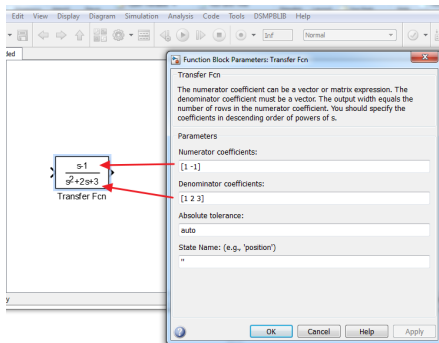
$$\ddot{y}(t) + a_1\dot{y}(t) + a_0y(t) = b_1\dot{x}(t) + b_0x(t)$$

↓

$$G(s) = \frac{Y(s)}{X(s)} = \frac{b_1s + b_0}{s^2 + a_1s + a_0}$$

Übertragungsfunktionen in *Simulink*

- *Transfer Fcn*-Block aus *Continuous*-Bibliothek
- Eingabe der Koeffizienten als Zeilenvektor
- Parameter des höchsten Polynomgrades zuerst (z.B. $[a_2 \ a_1 \ a_0]$)



Einschub: Was ist ein Polynom?

$$f(x) = a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \dots + a_1 \cdot x + a_0$$

mit n : Grad des Polynoms

Polynom in *Vektor*-Schreibweise:

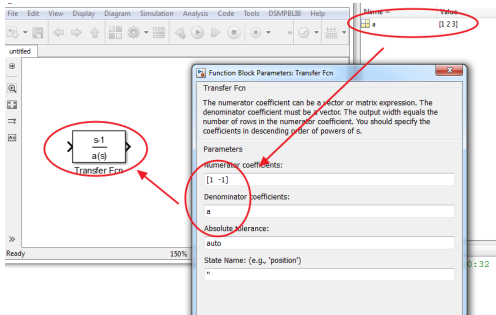
$$f(x) = [a_n \ a_{n-1} \ \dots \ a_1 \ a_0] \cdot \begin{bmatrix} x^n \\ x^{n-1} \\ \vdots \\ x \\ 1 \end{bmatrix} = \underline{a} \cdot \underline{x}$$

- der *Zeilenvektor* \underline{a} enthält die Parameter des Polynoms
- Eingabe eines Zeilenvektors in *MATLAB*

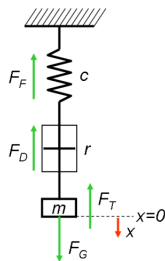
```
>> a = [1 2 3];
```

Übertragungsfunktionen in *Simulink*

- Vektoren können direkt als Parameter eingegeben werden
- besser Vektor mit Koeffizienten in Parameterdatei als *MATLAB*-Variable definieren



Beispiel: Feder–Masse–Dämpfer–System (inhomogene DGL)



Die Differentialgleichung für ein System bestehend aus Masse, Feder und Dämpfer lautet

$$m \cdot g = c \cdot x(t) + r \cdot \dot{x}(t) + m \cdot \ddot{x}(t)$$

Beispiel: Feder–Masse–Dämpfer–System

$$m \cdot \ddot{x}(t) + r \cdot \dot{x}(t) + c \cdot x(t) = F_g(t)$$

Nach *Laplace*–Transformation lautet die Übertragungsfunktion

$$\begin{aligned} G(s) &= \frac{\frac{1}{c}}{\frac{m}{c} \cdot s^2 + \frac{r}{c} \cdot s + 1} \\ &= \frac{\frac{1}{m}}{s^2 + \frac{r}{m} \cdot s + \frac{c}{m}} \quad (\text{Normalform}) \end{aligned}$$

mit

$\frac{r}{m}$: spezifischer Dämpfungsbeiwert

$\frac{c}{m}$: spezifischer Steifigkeitsbeiwert

Beispiel: Feder–Masse–Dämpfer–System

- Umsetzung mit *Transfer Fcn*-Block aus *Continuous*-Bibliothek
- Berechnungen der Parameter in *Transfer Fcn*-Block

Editor - D:\Projekte\MATLAB\FederMasseDaempfer_param.m

```

7 % Parameterdatei für FederMasseDaempfer.mdl
8 %
9 % AUTHOR(S)
10 % R.Blehschmidt (Labor für Mechatronik)
11 %
12 % DATE OF CREATION
13 % 19.03.2014
14 %
15 % VERSION / LAST CHANGE
16 % 0.52 / 19.03.2014
17 %-----
18 clear all
19 clc
20
21 % Parameter des Feder-Mass-
22 g = 9.81;
23 m = 1;
24 c = 1000;
25 r = 10;
26
27
28
29
30
31 Command Window
32 >>

```

Workspace

Name	Value	Min	Max
c	1000	1000	1000
g	9.8100	9.8100	9.8100
m	1	1	1
r	10	10	10

Function Block Parameters: Transfer Fcn

Transfer Fcn

The numerator coefficient can be a vector or matrix expression. The denominator coefficient must be a vector. The output width equals the number of rows in the numerator coefficient. You should specify the coefficients in descending order of powers of s.

Parameters

Numerator coefficients:
[1/m]

Denominator coefficients:
[1 r/m c/m]

Absolute tolerance:
auto

State Name: (e.g., 'position')
*

OK Cancel Help Apply

Simulink Diagram: Constant (g) → Transfer Fcn ($\frac{1/m}{s^2 + r/m \cdot s + c/m}$) → Scope

Beispiel: Feder–Masse–Dämpfer–System

- alle Parameter werden in *MATLAB* berechnet
- konsequente Umsetzung mit Parameterdatei–Konzept

The image shows a MATLAB Simulink workspace with three main windows:

- Editor (FederMasseDaempfer2_param.m):** Contains MATLAB code for parameter calculation. Red circles highlight:
 - Lines 27-31: `b0 = 1/m;`, `a0 = c/m;`, `a1 = r/m;`, `a2 = 1;`
 - Line 26: Comment `% Parameter für Transfer Funct`
- Block Diagram (FederMasseDaempfer2):** Shows a Simulink model with a 'Constant' block (g) connected to a 'Transfer Fcn' block. The transfer function is displayed as $\frac{b0}{a2 \cdot s^2 + a1 \cdot s + a0}$. Red circles highlight the transfer function block and the parameter `b0` in the numerator.
- Function Block Parameters: Transfer Fcn:** A dialog box for configuring the transfer function block. Red circles highlight:
 - The 'Denominator coefficients' field, which contains the vector `[a2 a1 a0]`.
 - The 'Numerator coefficients' field, which contains the vector `[b0]`.
- Workspace:** A table showing parameter values:

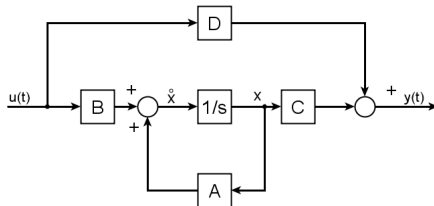
Name	Value
a0	1000
a1	10
a2	1
b0	1
c	1000
g	9.8100
m	1
r	10

Red arrows indicate the flow of information: from the MATLAB code to the block diagram, and from the workspace table to the parameter editor.

Darstellung von Systemen im Zustandsraum

- Differentialgleichungen höherer Ordnung können in Systeme von Differentialgleichungen 1. Ordnung umgeformt werden
- viele Analyseverfahren (z.B. Steuerbarkeit, Beobachtbarkeit) und Regelungskonzepte für *Zustandsraum* vorhanden

Allgemeine Zustandsraumdarstellung



$$\dot{\underline{x}}(t) = \underline{A} \cdot \underline{x}(t) + \underline{B} \cdot \underline{u}(t)$$

$$\underline{y}(t) = \underline{C} \cdot \underline{x}(t) + \underline{D} \cdot \underline{u}(t)$$

Darstellung von Systemen im Zustandsraum

- Zustandsgrößen werden als Vektoren abgebildet
- Parameter werden als Vektoren oder Matrizen abgebildet

Beispiel: System zweiter Ordnung (1 Eingang, 2 Ausgänge)

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \cdot u(t)$$

$$y(t) = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \cdot \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \cdot u(t)$$

mit

$$\underline{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \underline{b} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \underline{C} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}, \underline{d} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

Beispiel: Feder–Masse–Dämpfer–System

- Zustandsgröße Weg $x(t)$
- Zustandsgröße Geschwindigkeit $v(t) = \dot{x}(t)$

Differentialgleichung

$$\ddot{x}(t) = -\frac{c}{m} \cdot x(t) - \frac{r}{m} \cdot \dot{x}(t) + \frac{1}{m} \cdot F_g(t)$$

Darstellung im Zustandsraum

$$\begin{bmatrix} v(t) \\ a(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{r}{m} \end{bmatrix} \cdot \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \cdot u(t)$$
$$\begin{bmatrix} x(t) \\ v(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \cdot u(t)$$

Beispiel: Feder–Masse–Dämpfer–System

Parametermatrizen und -vektoren

$$\underline{A} = \begin{bmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{r}{m} \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}, \quad \underline{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \underline{d} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

mit

A : Systemmatrix, b : Eingangsvektor

C : Ausgangsmatrix, d : Durchgangsvektor

Eingabe in *MATLAB*

» A = [0 1; -c/m -r/m];

» b = [0; 1/m];

» C = [1 0; 0 1];

» D = [0; 0];

Beispiel: Feder–Masse–Dämpfer–System

- Umsetzung mit *State Space*-Block aus *Continuous*-Bibliothek
- Parameter für Matrizen in *MATLAB* berechnen

The screenshot displays the MATLAB environment with a script file named 'FederMasseDaempfer_param.m'. The script defines the following parameters and matrices:

```

10 % R. Biechenschlag (LADY FÜR MECHATRONIK)
11 %
12 % DATE OF CREATION
13 % 19.03.2014
14 %
15 % VERSION / LAST CHANGE
16 % 0.16 / 31.03.2014
17 %
18 -----
19 clear all
20 c1c
21 %
22 % Parameter des Feder-Masse-Dämpfer-Systems
23 g = 9.81;
24 m = 1;
25 c = 1000;
26 r = 10;
27 % Parameter für Zustandsraum
28 A = [0 1; -c/m -r/m];
29 b = [0; 1/m];
30 C = [1 0; 0 1];
31 d = [0; 0];

```

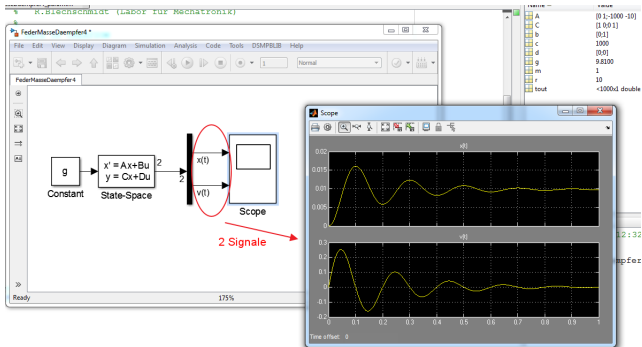
The Simulink diagram shows a 'Constant' block with value 'g' connected to a 'State Space' block. The State Space block is configured with the following parameters:

- State-space model: $dx/dt = Ax + Bu$, $y = Cx + Du$
- Parameters:
 - A: $\begin{bmatrix} 0 & 1 \\ -c/m & -r/m \end{bmatrix}$
 - B: $\begin{bmatrix} 0 \\ 1/m \end{bmatrix}$
 - C: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
 - D: $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
- Initial conditions: 0
- Absolute tolerance: auto
- State Name: (e.g., 'position')

Red circles and arrows highlight the parameter definitions in the MATLAB script and their corresponding values in the Simulink State Space block parameters dialog.

Beispiel: Feder–Masse–Dämpfer–System

- *State Space*-Block liefert Ausgangsvektor
- Aufteilen des Signals mit *Demux*-Block sinnvoll



- 1) Erstellen Sie das Feder–Masse–Dämpfer-Modell mit dem Transfer Fcn–Block.
- 2) Erstellen Sie das Feder–Masse–Dämpfer-Modell mit dem State Space–Block.
- 3) Vergleichen Sie die Sprungantworten der unterschiedlich modellierten Systeme.

Systemparameter Feder–Masse–Dämpfer:

$m = 1\text{kg}$	Masse
$g = 9.81\text{kgm/s}^2$	Gravitationsbeschleunigung
$c = 1000\text{N/m}$	Federkonstante
$r = 10\text{Ns/m}$	Dämpferkonstante