Technik

Informatik & Medien

Hochschule Ulm

University of

Applied Sciences

Computerlab 3: Car Suspension System II

- Analysis of the Car Suspension System using a MatLab program -

Prof. Dr. A. Beckmann

---

1 Model of the Car Suspension 1

2 External force as input 1

2.1 Non-normalized System 2

2.2 Normalized System 2

2.3 Parameterfile of the nonnormalized car suspension system 4

2.4 Parameterfile of the normalized car suspension system 5

2.5 Main program for analysis of a LTI-system 6

2.6 Simulations 8

2.7 Discussion of the results 9

2.7.1 Discussion of the non-normalized system 9

2.7.2 Discussion of the normalized system: 10

3 Road Profile as Input 11

3.1 Parameterprogram of the normalized system 12

3.2 Simulations on the normalized system 12

3.3 Discussion of the results 12

Prof. Dr. A. Beckmann

Hochschule Ulm
University of Applied Sciences

Computerlab 3:
Car Suspension System II

1 Model of the Car Suspension

An automobile suspension system consists of a spring and a hydraulic shock absorber as a damping part. The differential equation of the system is:

$$m\frac{d^2y}{dt^2} + b\frac{dy}{dt} + ky = F_{ext}$$

input force
restoring force
friction force

b = friction coefficient

k = spring coefficient

$m = m_{Car}/4$ (car has 4 wheels)

This is a 2nd order LTI-system. The input of the system is the external force, the output of the system is the deflection y of the car. The system parameters of the <u>non</u>normalized system are:

$$a_2 = m;\ a_1 = b;\ a_0 = k;\ b_2 = 0;\ b_1 = 0;\ b_0 = 1$$

The car suspension system shall be characterized by the following values:

- friction coefficient: b = 700 Ns/m (in case of underdamping),
- spring coefficient: k = 130000 N/m,
- mass of the car: m_{Car} = 1580 kg.

As the car has 4 wheels so we have to use $m = m_{Car} / 4$.

2 External force as input

We want to study the car suspension system with respect to

1. Unit impulse response $y_I(t)$,
2. Unit step response $y_S(t)$,
3. Frequency response (Bode plots, Nyquist plot),
4. Poles and zeros of the transfer function.

For impulse response and step response we performed simulations using a Simulink model in ComputerLab2. Now we want to write a MatLab program that covers all 4 aspects.

Computerlab 3:
Car Suspension System II

2.1 Non-normalized System

The differential equation of the car suspension system is:

$$m\frac{d^2y}{dt^2} + b\frac{dy}{dt} + ky = F_{ext} \qquad => \qquad \frac{d^2y}{dt^2} + \frac{b}{m}\frac{dy}{dt} + \frac{k}{m}y = \frac{F_{ext}}{m}$$

Using $\omega_o{}^2 = \frac{k}{m}$ and $\delta = \frac{b}{2m}$ we obtain: $\frac{d^2y}{dt^2} + 2\delta\frac{dy}{dt} + \omega_o{}^2 y = \frac{F_{ext}}{m}$

(with $\omega_o =$ natural frequency of the system; $\delta =$ decay coefficient)

The system parameters are:

$$a_2 = 1;\ a_1 = 2\delta;\ a_0 = \omega_o{}^2;\ b_2 = 0;\ b_1 = 0;\ b_0 = \frac{1}{m}$$

2.2 Normalized System

We normalize the system with respect to maximum values. From the steady state condition we derive:

$$\omega_o{}^2 y_t = \frac{F_{ext}}{m}.$$

From this we get for maximum input: $\omega_o{}^2 y_{t,\max} = \frac{F_{ext,\max}}{m}$

After normalization to maximum values we get:

$$\frac{1}{\omega_o{}^2}\frac{d^2}{dt^2}\left(\frac{y}{y_{t,\max}}\right) + \frac{2\delta}{\omega_o{}^2}\frac{d}{dt}\left(\frac{y}{y_{t,\max}}\right) + \frac{y}{y_{t,\max}} = \frac{F_{ext}}{F_{ext,\max}}$$

$$\frac{1}{\omega_o{}^2}\frac{d^2y'}{dt^2} + \frac{2\delta}{\omega_o{}^2}\frac{dy'}{dt} + y' = F_{ext}'$$

$$T_2{}^2\frac{d^2y}{dt^2} \quad + T_1\frac{dy}{dt} + y = K_P \cdot u$$ **general PT2-element**

The differential equation of the normalized system corresponds to a PT2-element.

The time constants of the system are: $T_1 = \frac{2\delta}{\omega_o{}^2}$; $T_2 = \frac{1}{\omega_o}$.

The normalized values of the input $F_{ext}' = F_{ext}/F_{ext,max}$ and output $y' = y/y_{t,max}$ are values between 0 and 1. Therefore $y' = y/y_{t,max} = 0{,}5$ means, that the displacement will 50% of the maximum value given by:

$$y_{t,\max} = \frac{F_{ext,\max}}{m\omega_o^{\ 2}}$$

The system parameters of the normalized system are:

$$a_2 = \frac{1}{\omega_o^{\ 2}}\ ;\ a_1 = \frac{2\delta}{\omega_o^{\ 2}}\ ;\ a_0 = 1;\ b_2 = 0;\ b_1 = 0;\ b_0 = 1$$

We need the system parameters to write down the transfer function of the system:

$$G(s) = \frac{b_2 s^2 + b_1 s + b_0}{a_2 s^2 + a_1 s + a_0}$$ **transfer function of a 2nd order LTI-system**

All parameters will be defined in parameter files:

- **CSS_parameters.m** for parameters of the car suspension
- **Dia_parameters.m** for the parameters of the diagrams

The transfer function of the system will be defined in system files:

- **CSS_nonnormalized.m** for the non-normalized system
- **CSS_normalized.m** for the normalized system

From the system file we will call the main program for analysis **lti_analysis_4.m**.

2.3 Parameterfile of the nonnormalized car suspension system

Write the following MatLab program **CSS_parameters.m** to define the parameters of the car suspension:

```
% Parameters of the car suspension system

% constant coefficients of the systems
k = 130000;          % constant of the spring in N/m
m_car = 1580;        % mass of the car

m = m_car/4;         % car has 4 wheels => 4 suspensions

F_ext_max = 500;

disp('natural angular frequency of the system in 1/s: ')
w_0 = sqrt(k/m)

disp('natural frequency of the system in Hz: ')
f_0 = w_0/(2*pi)

disp('oscillation period of the undamped system')
T_0 = 1/f_0

disp('maximum deflection in m: ')
y_max = F_ext_max/(m*w_0^2)

% decay coefficients in 1/s
% delta(1) = underdamping
% delta(2) = less underdamping in 1/s
% delta(3) = critical damping in 1/s
% delta(4) = overdamping in 1/s

delta = [w_0/5  w_0/2  w_0  2*w_0];
```

Write the MatLab program **Dia_parameters.m** to define the parameters that will used for the diagrams:

```
% Parameters for the diagrams
% number of periods in time intervall for simulation
N_periods = 5;

% sample time for simulation
N_sample = 100  % number of samples per period T_ext
t_sample = 1/N_sample / f_0; % sampling time

% time vector for impulse response and step response
t_max = N_periods*N_sample*t_sample;
t_dia = 0:t_sample:t_max;

% create vector of the angular frequencies N_w values
% from 10-4 to 10+6
N_w = 1000;   % Number of samples in frequency range
w = logspace(-4,6,N_w);
```

Write the following MatLab program **CSS_nonnormalized.m** to define the system by its transfer function and to call the analysis program:

```
% Definition of the transfer function
% and call of the analysis programm
clear               % delete all variables
close all           % close all figure windows

% Parameters of the suspension system
CSS_parameters

% Parameters for diagrams
Dia_parameters

% system parameters of the non-normalized system
% coefficients of numerator of transfer function
num = [0  0  1/m;...
       0  0  1/m;...
       0  0  1/m;...
       0  0  1/m];

% coefficients of denominator of transfer function as a matrix
den = [1   2*delta(1)  w_0^2;...
       1   2*delta(2)  w_0^2;...
       1   2*delta(3)  w_0^2;...
       1   2*delta(4)  w_0^2];

% define system by its transfer function
G1  = tf(num(1,:),den(1,:));
G2  = tf(num(2,:),den(2,:));
G3  = tf(num(3,:),den(3,:));
G4  = tf(num(4,:),den(4,:));

lti_analysis_4   % call main program lti_analysis_4.m
% *** end of program***
```

2.4 Parameterfile of the normalized car suspension system

Rename the above parameterfile: **CSS_normalized.m.** Change the system parameters according to the theory of the normalized system in section 2.2.

```
% system parameters of the normalized system
% coefficients of numerator of transfer function
num = [ 0  0  1 ;...
        0  0  1 ;...
        0  0  1 ;...
        0  0  1 ];

% coefficients of denominator of transfer function
% as a matrix
den = [1/w_0^2   2*delta(1)/w_0^2   1 ;...
       1/w_0^2   2*delta(2)/w_0^2   1 ;...
       1/w_0^2   2*delta(3)/w_0^2   1 ;...
       1/w_0^2   2*delta(4)/w_0^2   1 ];
```

2.5 Main program for analysis of a LTI-system

MatLab offers a variety of commands to analyse LTI-systems. Write the program **lti_analysis_4.m** that will be used for comparison of 4 different systems G1, G2, G3 and G4.

```
% Program for analysis of LTI-Systems
% Comparison of 4 systems

close all
%%
% Bode plots of the frequency response
% for given angular frequencies in 1/s
figure(1)    % new window for the diagram
bode(G1,'-b',G2,'-g',G3,'-r',G4,'-k',w)
grid         % draws the grid
title('CSS - Bode Diagrams','FontSize',12,'FontWeight','bold')
%%
% Bode plots of the frequency response
% for given frequencies in Hz
figure(2)    % new window for the diagram
h = bodeplot(G1,'-b',G2,'-g',G3,'-r',G4,'-k',w);
grid         % draws the grid
title('CSS - Bode Diagrams','FontSize',12,'FontWeight','bold')
setoptions(h,'FreqUnits','Hz')  % use Hz instead of rad/s
%%
% plot of the frequency response locus
figure(3)   % new window for the diagram
nyquist(G1,'-b',G2,'-g',G3,'-r',G4,'-k',w)
title('CSS - Frequency Response . . .
Locus','FontSize',12,'FontWeight','bold')
%%
% plot of poles and zeros in the complex plane
figure(4)   % new window for the diagram
pzmap(G1,G2,G3,G4)
title('CSS - Poles and Zeros','FontSize',12,'FontWeight','bold')
%%
% plot of unit impulse response
figure(5)   % new window for the diagram
impulse(G1,'-b',G2,'-g',G3,'-r',G4,'-k',t_dia)
grid         % draws the grid
title('CSS - Unit Impulse Response')
%%
% plot of unit step response
figure(6)   % new window for the diagram
step(G1,'-b',G2,'-g',G3,'-r',G4,'-k',t_dia)
grid         % draws the grid
title('CSS - Unit Step Response','FontSize',12,'FontWeight','bold')

% end of main program
```

Please note, that the command “impulse” will use unit impulse input and “step” will use unit step input.

Customizing the plots:

- **Bodeplots:** If you want to change scaling and units in bode plots or color and size of the title or the axislabels, use the following lines:

```
% Bode plots of the frequency response customized
figure   % new window for the diagram
h1 = bodeplot(G1,'-b',G2,'-g',G3,'-r',G4,'-k',w);
grid
p1 = getoptions(h1);
% Frequency
p1.FreqUnits = 'Hz';      % default: rad/s
p1.FreqScale = 'linear'; % default: log
% Magnitude
p1.MagUnits = 'abs';      % default: dB
p1.MagScale = 'linear';   % default: log
% Phase
p1.PhaseUnits = 'rad';    % default: deg
% title
p1.Title.String  =  'CSS - Bodediagrams';
p1.Title.FontSize = 12;
p1.Title.Color = [0 0 1];     % RGB here: blue
% Labels of x-axis and y-axis
p1.Xlabel.FontSize = 12;
p1.Xlabel.Color = [1 0 0];    % RGB here: red
p1.Ylabel.FontSize = 12;
p1.Ylabel.Color = [0 1 0];    % RGB here: green
setoptions(h1,p1)
```

- **Customizing Nyquistplots:**

```
% plot of the frequency response locus
figure  % new window for the diagram
h2 = nyquistplot(G1,'-b',G2,'-g',G3,'-r',G4,'-k',w);
p2 = getoptions(h2);
p2.Title.String  =  'CSS - Frequency Response Locus';
p2.Title.FontSize = 12;
p2.Title.Color = [0 0 1];     % RGB here: blue
p2.Xlabel.FontSize = 12;
p2.Xlabel.Color = [1 0 0];    % RGB here: red
p2.Ylabel.FontSize = 12;
p2.Ylabel.Color = [0 1 0];    % RGB here: green
setoptions(h2,p2)
```

or

```
xlabel('Time in s','FontSize',12,'FontWeight','bold','Color','r')
…
```

2.6 Simulations

Run the program for the non-normalized system and for the normalized system. Check your result for the step response by calculating the period T_o from the natural frequency:

$$\omega_o = \sqrt{\frac{k}{m}} = \sqrt{\frac{\qquad\qquad}{\qquad\qquad}} = \_\_\_\_\_\_\_\_\_\_\_\_$$

The corresponding period is:

$$T_o = \frac{2\pi}{\omega_o} = \_\_\_\_\_\_\_\_\_\_\_\_$$

The step response of the critically damped system should have reached its terminal value roughly after this time.

The terminal value of the step response of the non-normalized system should be:

$$y_{t,\max} = \frac{F_{ext,\max}}{m\omega_o^{\ 2}} = \frac{\qquad\qquad}{\qquad\qquad} = \_\_\_\_\_\_\_\_\_\_\_\_$$

In the simulations on the normalized system the terminal value of the step response should be equal to 1.

- **Customize the plots (title, labels, remarks, font size…).**
- **Transfer all diagrams into Word.**
- **Print out all diagrams as an example for a PT2-element.**
- **Discuss your results on the following pages.**

Try the LTI-Viewer of MatLab as well to display the plots. Use the following command in the matlab command line after having run the parameter program:

```
ltiview({'bode','nyquist','step','impulse','pzmap'},G1,G2,G3,G4)
```

2.7 Discussion of the results

2.7.1 Discussion of the non-normalized system

Impulse response: ____

____

____

____

Step response: ____

____

____

____

Bode plots: ____

____

____

____

Nyquist plot: ____

____

____

____

Zeros and poles: ____

____

____

____

2.7.2 Discussion of the normalized system:

Impulse response: ____

____

____

____

Step response: ____

____

____

____

Bode plots: ____

____

____

____

Nyquist plot: ____

____

____

____

Zeros and poles: ____

____

____

____

3 Road Profile as Input

The system can be described by the following differential equation:

$$m\frac{d^2y}{dt^2}+b\frac{dy}{dt}+ky=b\frac{dy_{Road}}{dt}+ky_{Road}$$

We divide by m:

$$\frac{d^2y}{dt^2}+\frac{b}{m}\frac{dy}{dt}+\frac{k}{m}y=\frac{b}{m}\frac{dy_{Road}}{dt}+\frac{k}{m}y_{Road}$$

System input: $u=y_{Road}$, **system output**: $y=y$

Using $\omega_o=\sqrt{\frac{k}{m}}$ and $2\delta=\frac{b}{m}$ we obtain:

$$\boxed{\frac{d^2y}{dt^2}+2\delta\frac{dy}{dt}+\omega_o{}^2y=2\delta\frac{dy_{Road}}{dt}+\omega_o{}^2y_{Road}}$$

The system parameters of the nonnormalized system are:

$a_2=1,\ a_1=2\delta,\ a_0=\omega_0{}^2$

$b_2=0,\ b_1=2\delta,\ b_0=\omega_0{}^2$

Show, that the differential equation of the normalized system is:

$$\frac{1}{\omega_o{}^2}\frac{d^2}{dt^2}\left(\frac{y}{y_{t,\max}}\right)+\frac{2\delta}{\omega_o{}^2}\frac{d}{dt}\left(\frac{y}{y_{t,\max}}\right)+\frac{y}{y_{t,\max}}=\frac{2\delta}{\omega_o{}^2}\frac{d}{dt}\left(\frac{y_{Road}}{y_{Road,\max}}\right)+\frac{y_{Road}}{y_{Road,\max}}$$

$$\frac{1}{\omega_o{}^2}\frac{d^2y'}{dt^2}+\frac{2\delta}{\omega_o{}^2}\frac{dy'}{dt}+y'=\frac{2\delta}{\omega_o{}^2}\frac{dy_{Road}'}{dt}+y_{Road}'$$ **DT2-element**

3.1 Parameterprogram of the normalized system

Rename the parameter program of the normalized car suspension system **CSS_normalized.m** to **CSS_road_profile.m** and modify the coefficients of the numerator of the transfer function:

```
% system parameters of the normalized system
% coefficients of numerator of transfer function
num = [0  2*delta(1)/w_0^2  1; . . .
       0  2*delta(2)/w_0^2  1; . . .
       0  2*delta(3)/w_0^2  1; . . .
       0  2*delta(4)/w_0^2  1];
```

The coefficients of the denominator are the same as before.

3.2 Simulations on the normalized system

Run the new program **CSS_road_profile.m**, transfer all diagrams into Word and print them out as an example for a DT2-element.

3.3 Discussion of the results

Discuss your results with respect to “What do you see in the plot?” and “What did you expect?”

Impulse response: ____

____

____

____

____

Step response: ____

____

____

____

____

Bode plots: ____

____

____

____

____

____

____

____

____

Nyquist plot: ____

____

____

____

____

Zeros and poles: ____

____

____

____