Hochschule Karlsruhe
University of Applied Sciences
Fakultät für Elektro- und Informationstechnik

Laborbericht Regelungstechnik

Versuch Nr. 1

Jan Hoegen* Maileen Schwenk†

26. April 2024

1 Darstellung von Sinussignalen

Die Funktionen aus der Versuchsanleitung [1] werden mit MATLAB simuliert und in Abbildung 1 dargestellt.

$$x_1(t) = 2 \cdot sin(2\pi \cdot 2\,\text{kHz} \cdot t) \tag{1}$$

$$x_1(t) = 2 \cdot sin(2\pi \cdot 6\,\text{kHz} \cdot t - \frac{\pi}{4}) \tag{2}$$

Darüber hinaus wird das zusammen gesetzte Signal $x_3(t) = x_1(t) \cdot x_1(t)$ sowie eine Lissajous-Figur mit $x_1(t)$ auf der x-Achse und $x_2(t)$ auf der y-Achse abgebildet. Es ist zu erkennen, dass die Frequenz das doppelte von $x_1(t)$ mit einem DC-Offset beträgt. Der Code zum Erstellen der Grafiken ist in Anhang A zu sehen.

1.1 Fehlerhafte Darstellungen der Lissajous-Figur

Wird der Zeitbereich auf 0 s bis 3 s gelegt und somit die Größenordnung um 10^3 erhöht, ist die Figur zur Abbildung 1 gleich. Wird der Zeitbereich auf leicht verschoben, entsteht ein nicht interpretierbares Bild. Diese Effekte sind durch den Aliasing-Effekt zu begründen. Beide Änderungen sind in Abbildung 2 gezeigt.

2 Tiefpassanalyse

Für einen Tiefpass erster Ordnung gilt:

$$\frac{U_a}{U_e} = \frac{1}{1 + jwRC} \tag{3}$$

Die Bauteilwerte mit einer Grenzfrequenz von 100 kHz und einem gewählten Kondensator C von $1 \cdot 10^{-9}\,\text{F}$ berechnen sich zu:

$$f_g = \frac{1}{2\pi RC} \stackrel{!}{=} 1 \cdot 10^5\,\text{Hz} \tag{4}$$

$$\Rightarrow R = \frac{1}{2\pi \cdot f_g \cdot C} = 1591{,}55\,\Omega \tag{5}$$

Das Bodediagramm ist in Abbildung 3 und die zugehörige Ortskurve in Abbildung 4 dargestellt. Da die Ortskurve achsensymmetrisch zur x-Achse ist, kann das Diagramm ohne den Verlust von Informationen um genau diese Spiegelung verkürzt werden. In MATLAB wird dies durch die Option `ShowFullContour='off'` des `nyquistplot`-Befehls erreicht. Der Code zum Erstellen der Diagramme findet sich in Anhang B.

*Matrikel-Nr. 82358. E-Mail hoja1028@h-ka.de
†Matrikel-Nr. 83802. E-Mail scma1315@h-ka-de

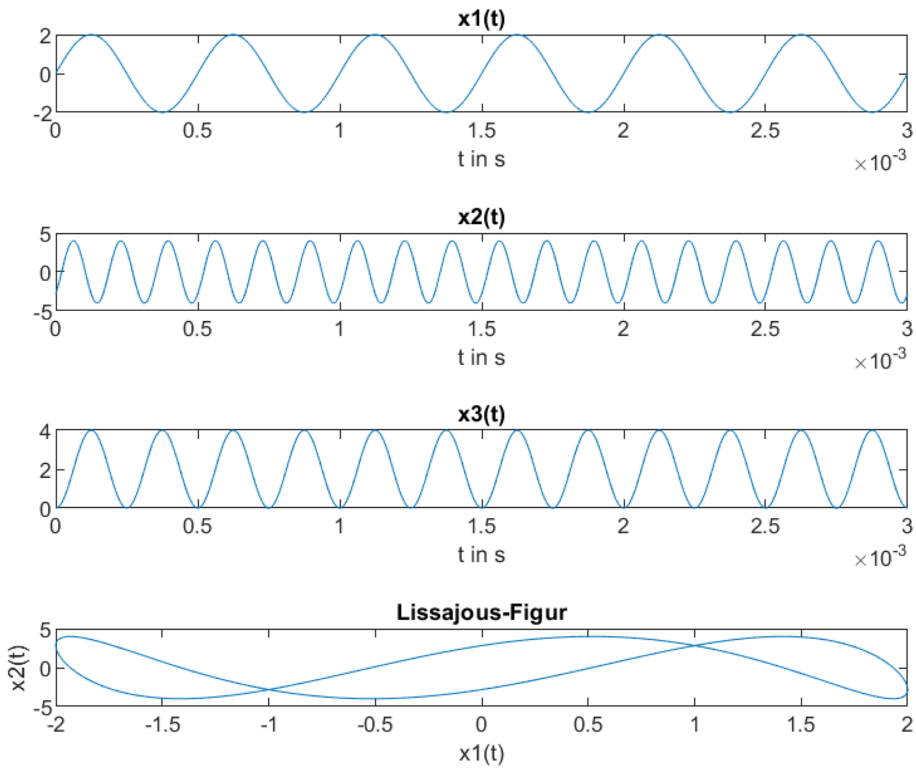

Abbildung 1: Darstellung der Sinussignale

Legende: Darstellung mit 10^3 Abtastpunkten

3 Temperaturregler

Nun wird ein Temperaturregler in Abbildung **??** simuliert. Der Zeitverlauf der eingestellten Solltemperatur, der Stellgröße und der Ausgangsgröße sind in Abbildung 6 dargestellt. Es ist zu erkennen, dass bei eingeschalteten Heizelement die Temperatur im Backofen schnell steigt bis zur Zieltemperatur von 160 °C. Anschließend wird periodisch auf- und abgewärmt, bis die Führungsgröße auf 0 °C verändert wird und die Temperatur absinkt. Die Parameter wurden im Anhang C definiert.

4 Literatur

[1] F. Keller, *Labor Regelungstechnik, Einführung in MATLAB/SIMULINK SS2024*, Karlsruhe: Hochschule Karlsruhe, 6. März 2024.

5 Autorenbeiträge

Maileen Schwenk und Jan Hoegen erstellten die Vorbereitung und Messauswertung. Jan Hoegen schrieb das Protokoll.

6 Verfügbarkeit des Codes

Der Code zum Auswerten der Daten und Erstellen der Diagramme findet sich unter `https://github.com/JaxRaffnix/Regelungstechnik`. Ebenfalls ist hier der Code zum Erstellen dieser Ausarbeitung hinterlegt.

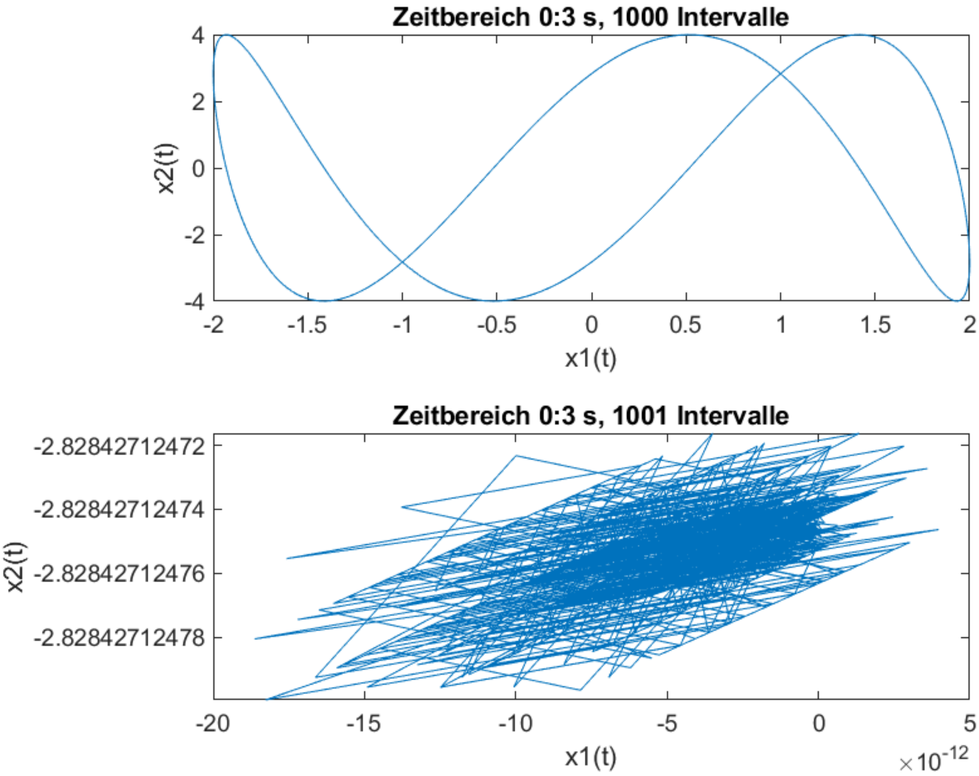

Abbildung 2: Fehlerhafte Lissajous-Figuren

A MATLAB-Code der Sinussignale

../versuch1/sinus.m

```
clear

LOCAL_DIRECTORY = "C:\Users\janho\Coding\Regelungstechnik\versuch1\";

% x-Axis
time = linspace(0, 3e-3, 1e3);

% declare functions
function f1 = sine1(time)
    f1 = 2 * sin(2 * pi * 2e3 * time - 0);
end
function f2 = sine2(time)
    f2 = 4 * sin(2 * pi * 6e3 * time - pi./4);
end

f1 = sine1(time);
f2 = sine2(time);
f3 = f1 .* f1;

% plot functions
sinplots = tiledlayout(4,1);
nexttile
plot(time, f1)
xlabel('t in s')
title('x1(t)')
nexttile
plot(time, f2)
xlabel('t in s')
title('x2(t)')
nexttile
plot(time, f3)
xlabel('t in s')
```

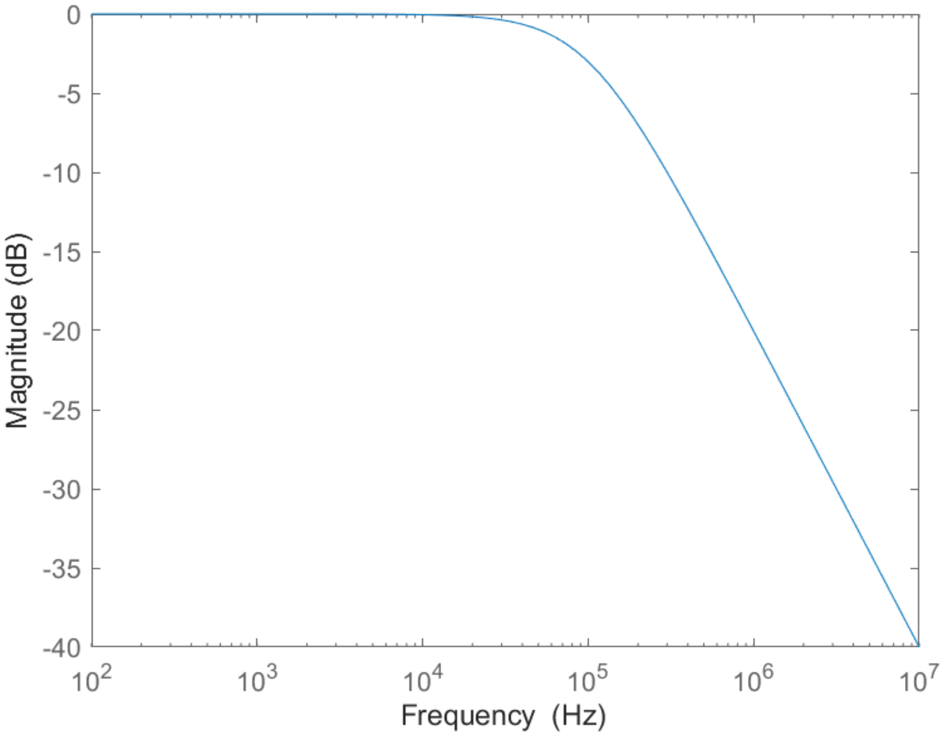

Abbildung 3: Bodediagramm des Tiefpasses

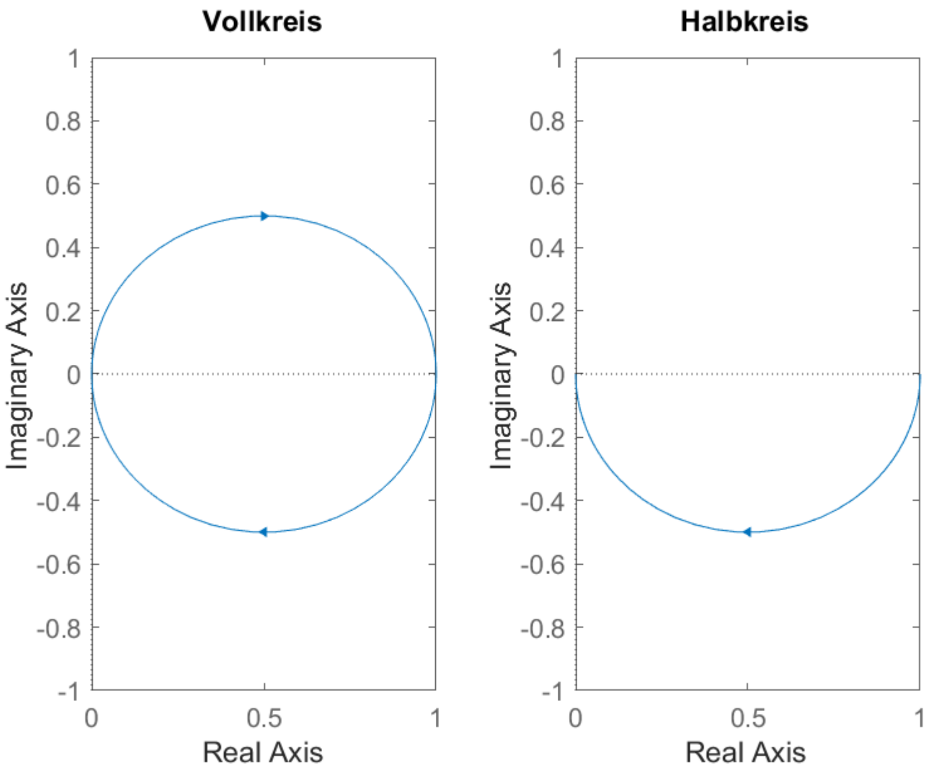

Abbildung 4: Ortskurven des Tiefpasses

Abbildung 5: Blockschaltbild des Temperaturreglers

Abbildung 6: Zeitsignal der Regelgrößen

```
title('x3(t)')
nexttile
plot(f1, f2)
xlabel('x1(t)')
ylabel('x2(t)')
title('Lissajous-Figur')

% exportgraphics(sinplots, "sinus.pdf", 'ContentType','vector') % known bug in matlab: exportgraphics won't
     save axis exponent!
saveas(sinplots, LOCAL_DIRECTORY + 'sinus.png')

% plot with wrong paramters
figure
lissplots = tiledlayout(2,1);
time = linspace(0, 3, 1e3);

nexttile
f1 = sine1(time);
f2 = sine2(time);
plot(f1, f2)
xlabel('x1(t)')
ylabel('x2(t)')
title('Zeitbereich 0:3 s, 1000 Intervalle')

% another set of wrong paramters
time = linspace(0, 3, 1e3+1);
nexttile
f1 = sine1(time);
f2 = sine2(time);
plot(f1, f2)
xlabel('x1(t)')
ylabel('x2(t)')
title('Zeitbereich 0:3 s, 1001 Intervalle')

saveas(lissplots, LOCAL_DIRECTORY + "lissjaou.png")
```

B MATLAB-Code zum Tiefpass

../versuch1/tiefpass.m

```
clear

LOCAL_DIRECTORY = "C:\Users\janho\Coding\Regelungstechnik\versuch1\";

FREQUENCY = 100e3;
CAPACITOR = 1e-9;
RESISTOR = 1 / (2.*pi.*FREQUENCY.*CAPACITOR)    % = 1.5915e+03
denominator = [RESISTOR*CAPACITOR, 1];
system = tf(1, denominator);

figure;
bode = bodeplot(system);
setoptions(bode, ...
    'FreqUnits','Hz', ...
    'PhaseVisible','off', ...
    'xlim', {[1e2, 1e7]} ...
);
title('');

saveas(gcf, LOCAL_DIRECTORY + 'bode.png')

figure;
subplot(1,2,1)      % rows, columns, position
    ny = nyquistplot(system);
    setoptions(ny, ...
        'Xlim', {[0,1]}, ...
        'YLim', {[-1, 1]} ...
    );
    title('Vollkreis');

subplot(1,2,2)
    ny_half = nyquistplot(system);
    setoptions(ny_half, ...
        'ShowFullContour', 'off', ...
        'Xlim', {[0,1]}, ...
        'YLim', {[-1, 1]} ...
    );
```

```
    title('Halbkreis');

saveas(gcf, LOCAL_DIRECTORY + 'ortskurve.png')
```

C MATLAB-Code zum Temperaturregler

../versuch1/temp_regler.m

```
% global parameters
LOCAL_DIRECTORY = "C:\Users\janho\Coding\Regelungstechnik\versuch1\";
STOPTIME = 70*60 - 1;   % hide last data point to hide second positive flank of the wave generator
POINTS = 1e2;

STEPSIZE = STOPTIME / POINTS;


% set model parameters
model = LOCAL_DIRECTORY +  'tempregler_modell.slx';
k1 = 10/400;
k2 = 400/230;
time_const = 15*60;

% run first simulation
controlller_on = 0.2;
controller_off = -0.2;
output = sim(model, "StopTime", num2str(STOPTIME), 'FixedStep', num2str(STEPSIZE));  % 1: control variable,
     2: manipulated_variable, 3: reference_variable

% second simulation with changed parameters
controlller_on = 0.01;
controller_off = -0.01;
output_new = sim(model, "StopTime", num2str(STOPTIME), 'FixedStep', num2str(STEPSIZE));  % 1: control
     variable, 2: manipulated_variable, 3: reference_variable

% plotting
variables = {"Führungsgröße", "Stellgröße", "Re"};
unnit = {"Spannung in V", };
figure;
% title(outputplot, "alte controller werte")
subplot(3,2,1)      % rows, columns, position
plot(output.yout{1}.Values.Time, output.yout{1}.Values.Data)
subplot(3,2,3)
plot(output.yout{2}.Values.Time, output.yout{2}.Values.Data)
subplot(3,2,5)
plot(output.yout{2}.Values.Time, output.yout{3}.Values.Data)

subplot(3,2,2)
plot(output_new.yout{1}.Values.Time, output_new.yout{1}.Values.Data)
subplot(3,2,4)
plot(output_new.yout{2}.Values.Time, output_new.yout{2}.Values.Data)
subplot(3,2,6)
plot(output_new.yout{2}.Values.Time, output_new.yout{3}.Values.Data)
```