

# Laborbericht Regelungstechnik

## Versuch Nr. 1

Jan Hoegen\*

Maileen Schwenk†

22. April 2024

## 1 Darstellung von Sinussignalen

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

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

$$x_2(t) = 2 \cdot \sin(2\pi \cdot 6 \text{ kHz} \cdot t - \frac{\pi}{4}) \quad (2)$$

Darüber hinaus wird das Zusammengesetzte Signal  $x_3(t) = x_1(t) \cdot x_2(t)$  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  $x_1(t)$  beträgt.

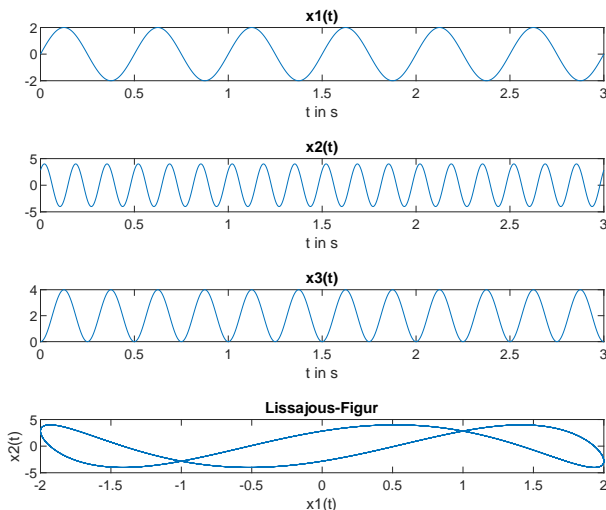


Abbildung 1: Darstellung der Sinussignale

Legende: Darstellung in  $10^3$  Intervallen

### 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 MISSING Beide Änderungen sind in Abbildung 2 gezeigt.

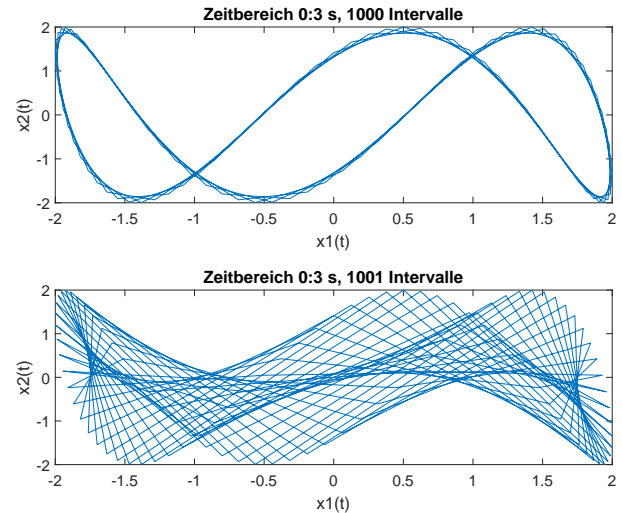


Abbildung 2: Fehlerhafte Lissajous-Figuren

## 2 Tiefpassanalyse

Für einen Tiefpass erster Ordnung gilt:

$$\frac{U_a}{U_e} = \frac{1}{1 + j\omega RC} \quad (3)$$

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

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

$$\Rightarrow R = \frac{1}{2\pi \cdot 10^5 \text{ Hz} \cdot 10^{-9} \text{ F}} = 1591,55 \Omega \quad (5)$$

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.

## 3 Temperaturregler

Nun wird ein Temperaturregler simuliert.

\*Matrikel-Nr. 82358. E-Mail [hoja1028@h-ka.de](mailto:hoja1028@h-ka.de)

†Matrikel-Nr. 83802. E-Mail [scma1315@h-ka.de](mailto:scma1315@h-ka.de)

## 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.

## A Anhang

../versuch1/sinus.m

```
clear

% x-Axis
time = linspace(0, 3e-3, 10e3); % seconds

% declare sine function
function f1 = myfun(amplitude, frequency, time, offset)
    f1 = amplitude * sin(2 * pi * frequency * time - offset);
end

f1 = myfun(2, 2e3, time, 0);
f2 = myfun(4, 6e3, time, -pi./4);
f3 = f1 .* f1;

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

exportgraphics(sinplots, "sinus.pdf", 'ContentType', 'vector')

% task d
figure
t=linspace(0.3,1e3);
lissplots = tiledlayout(2,1);
nexttile
plot(myfun(2, 2e3, t, 0), myfun(2, 6e3, t, -pi./4))
xlabel('x1(t)')
ylabel('x2(t)')
title('Zeitbereich 0:3 s, 1000 Intervalle')

% task e
t=linspace(0.3,1e3+1);
nexttile
plot(myfun(2, 2e3, t, 0), myfun(2, 6e3, t, -pi./4))
xlabel('x1(t)')
ylabel('x2(t)')
title('Zeitbereich 0:3 s, 1001 Intervalle')
```

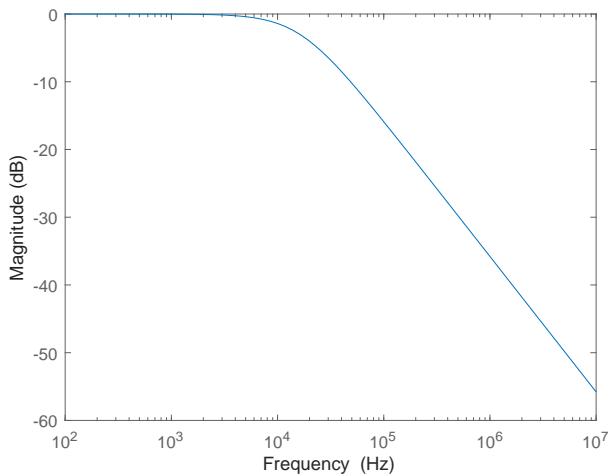


Abbildung 3: Bodediagramm des Tiefpasses

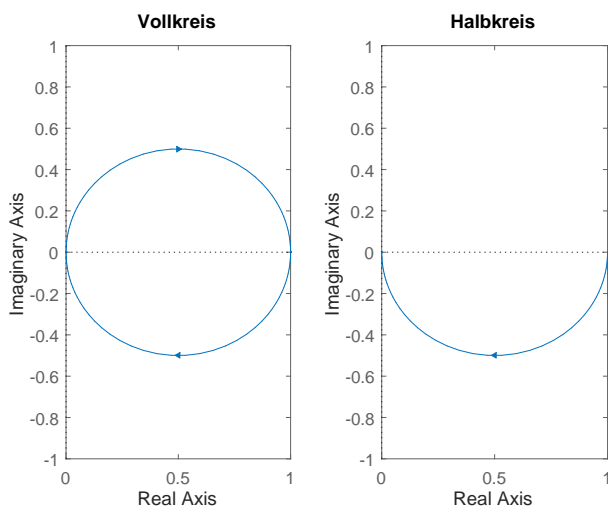


Abbildung 4: Ortskurven des Tiefpasses

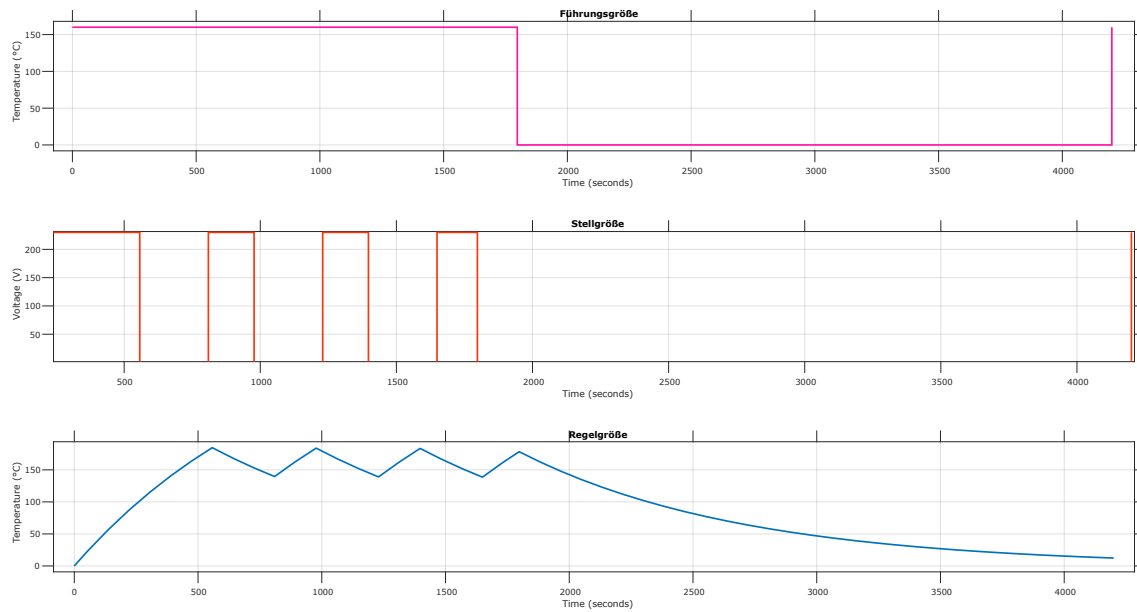


Abbildung 5: Zeitsignal der Regelgrößen

```
exportgraphics(lissplots, "lissjaou.pdf", 'ContentType', 'vector')
```

```
../versuch1/tiefpass.m
```

```
clear
```

```
FREQUENCY = 10^5;
RESISTOR = 1560;
CAPACITOR = 10^-9;
denominator = [RESISTOR*CAPACITOR*2*pi, 1];
system = tf(1, denominator);
```

```
figure;
bode = bodeplot(system);
setoptions(bode, ...
    'FreqUnits', 'Hz', ...
    'PhaseVisible', 'off', ...
    'xlim', {[10^2, 10^7]} ...
);
title('');
```

```
exportgraphics(gcf, "bode.pdf", 'ContentType', 'vector')
```

```
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');
```

```
exportgraphics(gcf, "ortskurve.pdf", 'ContentType', 'vector')
)
```