1 Untersuchung eines passiven Tiefpassfilters

1.1 Bestimmung der Übertragungsfunktion und des Frequenzgangs

In Abb. 1 ist die Schaltung des Tiefpassfilters gezeigt. Es ist eine nicht optimierte Schaltung, die als Beispiel für den Umgang mit solchen und ähnlichen Schaltungen dienen soll. Für passive Filter sind in der Literatur abhängig vom Typ des Filters wie Tschebyschev, Butterworth etc. Methoden zur Entwicklung beschrieben [4], [3].

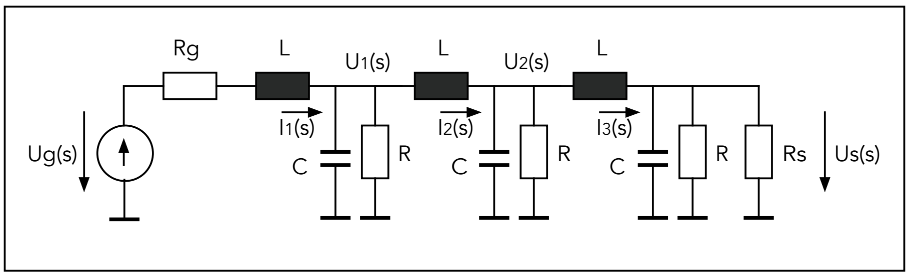

Abb. 1: Passives Tiefpassfilter 6. Ordnung

Wenn man versucht die Übertragungsfunktion im Bildbereich der Laplace-Transformation ausgehend von der Eingangsspannung zu bestimmen, wird es relativ kompliziert. Viel einfacher und auch leichter in MATLAB zu programmieren ist von der Ausgangsspannung auszugehen. Diese wird vorübergehend als bekannt angenommen, z.B. $U_s(s) = 1$, und danach wird die Eingangsspannung $U_g(s)$ berechnet, die zu dieser Ausgangsspannung führt. Zuletzt aus dem Verhältnis $U_s(s)/U_g(s)$ ergibt sich die gesuchte Übertragungsfunktion.

Mit den Bezeichnungen aus Abb. 1 ausgehend von $U_s(s)$ erhält man folgende Beziehungen :

$$\begin{aligned}
&U_s(s) = 1 \\
&I_3(s) = U_s(s)\Big(\frac{1}{R_s} + \frac{1}{R} + C \cdot s\Big); \qquad && U_2(s) = I_3(s) \cdot s \cdot L + U_s(s) \\
&I_2(s) = U_2(s)\Big(\frac{1}{R} + C \cdot s\Big) + I_3(s); && U_1(s) = I_2(s) \cdot s \cdot L + U_2(s) \\
&I_1(s) = U_1(s)\Big(\frac{1}{R} + C \cdot s\Big) + I_2(s); && U_g(s) = I_1(s)(R_g + s \cdot L) + U_1(s) \\
&H_s(s) = \frac{U_s(s)}{U_g(s)}
\end{aligned} \tag{1}$$

Im Skript `uebertrag_1.m` ist die Untersuchung programmiert. Es werden Anfänglich die Parameter der Schaltung initialisiert und danach mit folgenden Zeilen wird die Übertragungsfunktion als `transfer function` Objekt ermittelt:

```
s = tf('s');                 % s als Laplace Variable
```

```
% ------- Berechnung der Übertragungsfunktion vom Ausgang ausgehend
Us = 1;                       % Ausgangsspannung
I3 = Us*(1/Rs + 1/R + C*s);

U2 = I3*L*s + Us;
I2 = U2*(1/R + C*s) + I3;

U1 = I2*L*s + U2;
I1 = U1*(1/R + C*s) + I2;

Ug = I1*(L*s + Rg) + U1;

Hs = Us/Ug;              % Übertragungsfunktion (s System)

[num, den] = tfdata(Hs); % Cellen der Koeffizienten
b = num{:};     a = den{:};   % Koeffizienten des Zählers und Nenners
```

Mit `tf` wird die Variable s als die komplexe Variable der Laplace-Transformation definiert. Zuletzt wird die Übertragungsfunktion als `transfer function` Objekt erhalten:

```
Hs =
                                           1
    ------------------------------------------------------------------

  2.16e-46 s^6 + 1.476e-38 s^5 + 1.971e-30 s^4 + 9.551e-23 s^3 + 4.234e

                                         -15 s^2 + 1.15e-07 s + 1.13
 Continuous-time transfer function.
```

Die Koeffizienten des Zählers und des Nenners der Übertragungsfunktion werden aus dem Objekt `Hs` als `cell`-Strukturen mit `[num, den] = tfdata(Hs)` extrahiert. Weiter werden die numerischen Koeffizienten aus den Zellen mit `b = num{:};      a = den{:};` extrahiert.

Der Frequenzgang des Filters wird mit der Funktion `freqs` in einem Frequenzbereich von zwei Dekaden links und zwei Dekaden rechts der Resonanzfrequenz einer Stufe des Filters `f0` berechnet:

```
f = logspace(round(log10(f0/100)), round(log10(f0*100)), 200);
         % Frequenzbereich mit 2 Dekaden links und rechts
         % der Resonanzfrequenz
[H,w] = freqs(b, a, f*2*pi);  % Frequenzgang
```

In Abb. 2 ist der Frequenzgang dargestellt. Im Skript (`uebertrag_1.m`) werden auch die Pol- und Nullstellen der Übertragungsfunktion ermittelt und dargestellt. Mit

```
>> roots(a)
ans =
   1.0e+07 *
```

```
-0.4164 + 7.2168i
-0.4164 - 7.2168i
-0.8955 + 4.5914i
-0.8955 - 4.5914i
-2.1047 + 0.3810i
-2.1047 - 0.3810i
```

werden die Polstellen der Übertragungsfunktion, die alle in der linken komplexen Ebene liegen, berechnet.

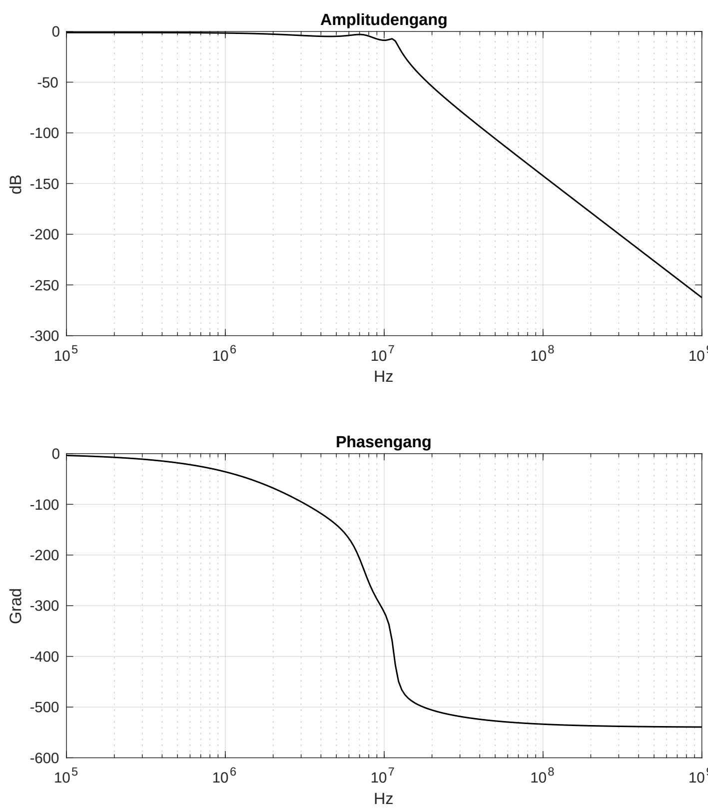

Abb. 2: Frequenzgang des passiven Tiefpassfilters 6. Ordnung

Der Frequenzgang kann auch direkt numerisch berechnet werden:

```
omega = 2*pi*f;                % Frequenzbereich (rad/s)
% ------- Berechnung der Übertragungsfunktion vom Ausgang ausgehend
Us = 1;                        % Ausgangsspannung
I3 = Us*(1/Rs + 1/R + C*j*omega);
U2 = I3.*(L*j*omega) + Us;
I2 = U2.*(1/R + C*j*omega) + I3;
U1 = I2.*(L*j*omega) + U2;
```

```
I1 = U1.*(1/R + C*j*omega) + I2;
Ug = I1.*(L*j*omega + Rg) + U1;
Hs2 = Us./Ug;              % Frequenzgang
```

Für denselben Frequenzbereich wird die Frequenz in rad/s in der Variable `omega` definiert. Danach werden ähnlich, wie im vorherigen Fall, für diesen Frequenzbereich alle Variablen der Schaltung berechnet. Zuletzt wird mit dem elementweiße Verhältnis `Us./Ug` der Frequenzgang ermittelt. Wie erwartet erhält man dasselbe Ergebnis. Man besitzt aber so nicht die Koeffizienten der Übertragungsfunktion für weitere Untersuchungen.

Zu bemerken sei, dass in diesem Fall alle Variablen Zeiger sind, die mit der Kreisfrequenz `omega` in der komplexen Ebene rotieren. Mit `Us = 1` ist die Lage des Zeigers der Ausgangsspannung auf der realen Achse angenommen. Man kann auch einen komplexen Wert, wie z.B. `Us = 1*exp(j*pi/3)` annehmen. Dann sind alle Zeiger mit diesem Winkel versetzt. Das Endergebnis für `Hs2` ist dasselbe.

1.2 Zustandsmodell

Bei der Berechnung der Antwort auf beliebige Eingangsspannungen wird immer ein Zustandsmodell [2] eingesetzt. Wenn $\mathbf{x}(t)$ der Zustandsvektor, $\mathbf{u}(t)$ der Eingangsvektor und $\mathbf{y}(t)$ der Ausgangsvektor sind, dann ist das Zustandsmodell durch folgende Gleichungen gegeben:

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= \mathrm{A}\mathbf{x}(t) + \mathrm{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathrm{C}\mathbf{x}(t) + \mathrm{D}\mathbf{u}(t) \end{aligned} \tag{2}$$

Die Matrizen des Zustandsmodells können mit der Funktion `tf2ss` ermittelt werden:

```
% -------- Zustandsmodell 1
[A1,B1,C1,D1] = tf2ss(b,a);
system1 = ss(A1,B1,C1,D1);     % Zustandssystem 1; Objekt ss (state space)
```

In dieser Schaltung sind die Zustandsvariablen die Spannungen der Kondensatoren und die Ströme der Induktivitäten. Die Zuordnung der Zustandsvariablen ist nach dieser Umwandlung nicht so einfach zu bestimmen. Ein besseren Einblick erhält man, wenn man den Zustandsmodell "zu Fuß" aus den Differentialgleichungen die man für die Zustandsvariablen schreiben kann, ermittelt:

$$\begin{aligned} \frac{du_s(t)}{dt} &= -\frac{u_s(t)}{C}\Big(\frac{1}{R} + \frac{1}{R_s}\Big) + \frac{i_3(t)}{C} \\ \frac{i_3(t)}{dt} &= -\frac{u_s(t)}{L} + \frac{u_2(t)}{L} \\ \frac{du_2(t)}{dt} &= -\frac{u_2(t)}{RC} - \frac{i_3(t)}{C} + \frac{i_2(t)}{C} \end{aligned} \tag{3}$$

und

$$\begin{aligned}
\frac{i_2(t)}{dt} &= \frac{u_1(t)}{L} - \frac{u_2(t)}{L} \\
\frac{du_1(t)}{dt} &= -\frac{u_1(t)}{RC} - \frac{i_2(t)}{C} + \frac{i_1(t)}{C} \\
\frac{i_1(t)}{dt} &= -\frac{u_1(t)}{L} - i_1(t)\frac{Rg}{L} + \frac{u_g(t)}{L}
\end{aligned} \tag{4}$$

Wenn man den Zustandsvektor mit folgendem Vektor definiert

$$\mathbf{x}(t) = [u_s(t), i_3(t), u_2(t), i_2(t), u_1(t), i_1(t)]^T \tag{5}$$

und $\mathbf{u}(t) = u_g(t)$ bzw. $\mathbf{y}(t) = u_s(t)$ erhält man folgendes Zustandsmodell:

$$\begin{bmatrix} \frac{du_s(t)}{dt} \\ \frac{di_3(t)}{dt} \\ \frac{du_2(t)}{dt} \\ \frac{di_2(t)}{dt} \\ \frac{du_1(t)}{dt} \\ \frac{di_1(t)}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{1}{C(R||R_s)} & \frac{1}{C} & 0 & 0 & 0 & 0 \\ -\frac{1}{L} & 0 & \frac{1}{L} & 0 & 0 & 0 \\ 0 & -\frac{1}{C} & -\frac{1}{RC} & \frac{1}{C} & 0 & 0 \\ 0 & 0 & -\frac{1}{L} & 0 & \frac{1}{L} & 0 \\ 0 & 0 & 0 & -\frac{1}{C} & -\frac{1}{RC} & \frac{1}{C} \\ 0 & 0 & 0 & 0 & -\frac{1}{L} & -\frac{R_g}{L} \end{bmatrix} \begin{bmatrix} u_s(t) \\ i_3(t) \\ u_2(t) \\ i_2(t) \\ u_1(t) \\ i_1(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u_g(t) \tag{6}$$

Daraus ergeben sich die ersten zwei Matrizen A und B. Die Matrizen C und D sind für diese Schaltung mit einem Eingang und einem Ausgang sehr einfach:

$$C = [1, 0, 0, 0, 0, 0] \quad \text{und} \quad D = 0 \tag{7}$$

Im Skript werden die Matrizen dieses zweiten Zustandsmodell in folgenden Zeilen berechnet:

```
% -------- Zustandsmodell 2
A2 = [-(1/C)*(1/R+1/Rs), 1/C 0 0 0 0;-1/L 0 1/L 0 0 0;
     0 -1/C -1/(R*C) 1/C 0 0; 0 0 -1/L 0 1/L 0;
     0 0 0 -1/C -1/(R*C) 1/C; 0 0 0 0 -1/L -Rg/L];

B2 = [0 0 0 0 0 1/L]';
C2 = [1 0 0 0 0 0];
D2 = 0;
system2 = ss(A2,B2,C2,D2);    % Zustandssystem 2; Objekt ss (state space)
```

Die Eigenwerte der Zwei Matrizen `A1` und `A2`, die auch die Pole der Übertragungsfunktion darstellen, sind gleich und werden mit der Function `eig` erhalten. Die Reihenfolge der Eigenwerte ist für die zwei Matrizen verschieden. Die Zustandsvariablen des ersten Modells, kann man nicht mit den physikalischen Zustandsvariablen, die die Spannungen der Kondensatoren und Ströme der Induktivitäten sind, verbinden. Das Zustandsmodell, das durch die vier Matrizen dargestellt wird, ist nicht einzigartig. Eine Übertragungsfunktion kann zu verschiedenen Zustandsmatrizen führen. Sicher muss die Ausgangsvariable von den beiden Modellen dieselbe sein. In `https://lpsa.swarthmore.edu/Representations/SysRepTransformations/TF2SS.html` sind einige Methoden zur Umwandlung einer Übertragungsfunktion in einem Zustandsmodell beschrieben.

1.3 Sprungantwort, Impulsantwort und die Antwort auf beliebige Eingangsspannungen

Die Antwort solcher LTI-Systeme (*Linear Time Invariant*) auf beliebige Anregung kann in MTLAB mit der Funktion `lsim` ermittelt werden.

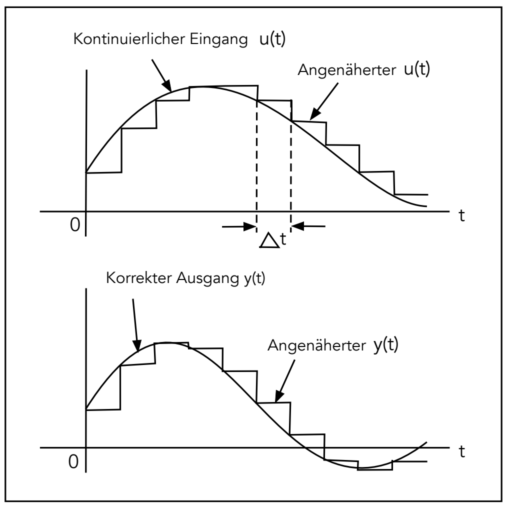

Abb. 3: Annäherungen beim Einsatz der Funktion **lsim**

In Abb. 3 sind die Annäherungen beim Einsatz der Funktion `lsim` dargestellt. Die Anregung als Eingangsvariable wird als konstant in kleinen Schritten Δt der Zeitvariable t angenommen. Der Ausgang und zusätzlich bei Bedarf auch die Zustandsvariablen werden am Ende dieser kleinen Intervallen ausgehend von den Werten der Zustandsvariablen am Anfang der Intervalle und der konstanten Anregung in diesen Intervallen berechnet. Es kann auch eine lineare Interpolation der Anregung festgelegt werden.

Die einfachste Methode aus den Zustandsvariablen zum Zeitpunkt t die neuen Zustandsvariablen zum Zeitpunkt $x(t + \Delta t)$ zu ermitteln ist die Euler-Methode [1]. Mit

$$\frac{d\mathbf{x}(t)}{dt} = \mathrm{A}\mathbf{x}(t) + \mathrm{B}\mathbf{u}(t) \tag{8}$$

wird eine sehr einfache Annäherung der Zustandsvariable $\mathrm{x}(t + \Delta t)$ aus der Zustandsvariable x(t) und der konstanten Anregung $u(t)$ im Intervall Δt berechnet:

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \Delta t\frac{d\mathbf{x}(t)}{dt} = \mathbf{x}(t) + \Delta t\big(\mathrm{A}\mathbf{x}(t) + \mathrm{B}\mathbf{u}(t)\big) \tag{9}$$

Aus den iterierten Zustandsvariablen werden die Ausgangsvariablen mit Hilfe der Matrizen C und D berechnet.

Als Beispiel für den Einsatz der Funktion `lsim` werden die Sprungantworten der zwei Systeme einmal mit dem Zustandsmodell mit den Matrizen `A2,B2,C2,D2` und danach mit dem Modell mit den Matrizen `A1,B1,C1,D1`. Es werden alle Zustandsvariablen ermittelt und dargestellt.

Im Skript werden die Zustandsvariablen für das System `system2` für ein Einheitsprung am Eingang mit folgenden Programmzeilen ermittelt:

```
Nt = 1000;
u = ones(Nt,1);              % Eingangssprung
delta_t = 1/10e8;            % Zeitschritt
t = (0:Nt-1)' *delta_t;      % Zeitvektor
x0 = zeros(6,1);             % Anfangszustandsvektor

[y2,t,x2] = lsim(system2,u,t,x0);
```

Im Spaltenvektor `u` sind die konstanten Werte des Eingangs für jedes Intervall des Zeitspaltenvektors `t` enthalten. Mit `x0` kann auch ein Anfangszustandsvektor (hier null) einbezogen werden. In Abb.4 sind die Zustandsvariablen dargestellt. Die erste Zustandsvariable `x2(:,1)` stellt auch die Ausgangsspannung `us` dar. Die Spannungen der anderen Kondensatoren erkennt man sehr leicht, weil sie annähern zu eins (Eingangssprung) tendieren, wie die Zustandsvariable `x2(:,3)`, die die Spannung `u2` des zweiten Kondensator darstellt und die Zustandsvariable `x2(:,5)`, die die Spannung `u1` des ersten Kondensator darstellt. Die anderen Zustandsvariablen stellen die Ströme der Induktivitäten dar.

Ähnlich werden die Zustandsvariablen des Systems `system1` ermittelt und dargestellt. In Abb. 5 sind diese Zustandsvariablen dargestellt. Hier ist die letzte Zustandsvariable `x1(:,6)` gewichtet mit dem Wert des Koeffizienten aus der Matrix `C1` das Ausgangssignal (`us`), wie erwartet gleich der Zustandsvariable `x2(:,1)` aus Abb. 4. Wie man sieht, kann man hier die restlichen Zustandsvariablen direkt nicht mit den physikalischen Zustandsvariablen verbinden.

Die Sprungantwort kann auch mit der Funktion `step` ermittelt werden:

```
>> [y, t] = step(system1);
>> figure
>> plot(t,y)
```

Ähnlich auch für das Modell `system2`.

Für die Ermittlung der Impulsantwort kann die Funktion `impulse` eingesetzt werden. Der Aufruf ist ähnlich dem Aufruf für die Sprungantwort mit `step`:

```
>> [y, t] = impulse(system1);
>>figure
>>plot(t,y)
```

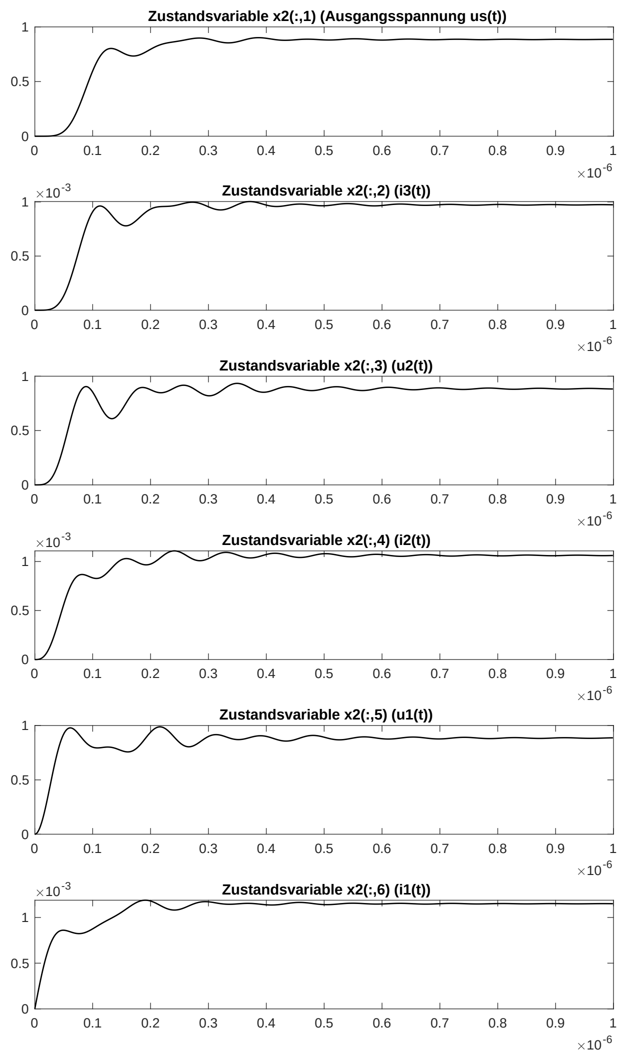

Abb. 4: Zustandsvariablen der Sprungantwort für das Modell system2

Die Ermittlung der Impulsantwort über die Funktion `lsim` ist lehrreich und wird im Skript mit folgenden Programmzeilen berechnet:

```
% ------- Impulsantwort mit lsim
N = 1000;
u = [1, zeros(1,N-1)]';
t = [0:N-1]'*delta_t;
y = lsim(Hs,u,t)/delta_t;
```

Als Eingangssignal `u` wird ein Puls der Höhe eins und Dauer `delta_t` mit der Fläche `delta_t` angesetzt. Die Antwort auf eine Delta-Funktion der Fläche eins wird dann durch Teilung der Antwort auf den Puls der Fläche `delta_t` mit dieser

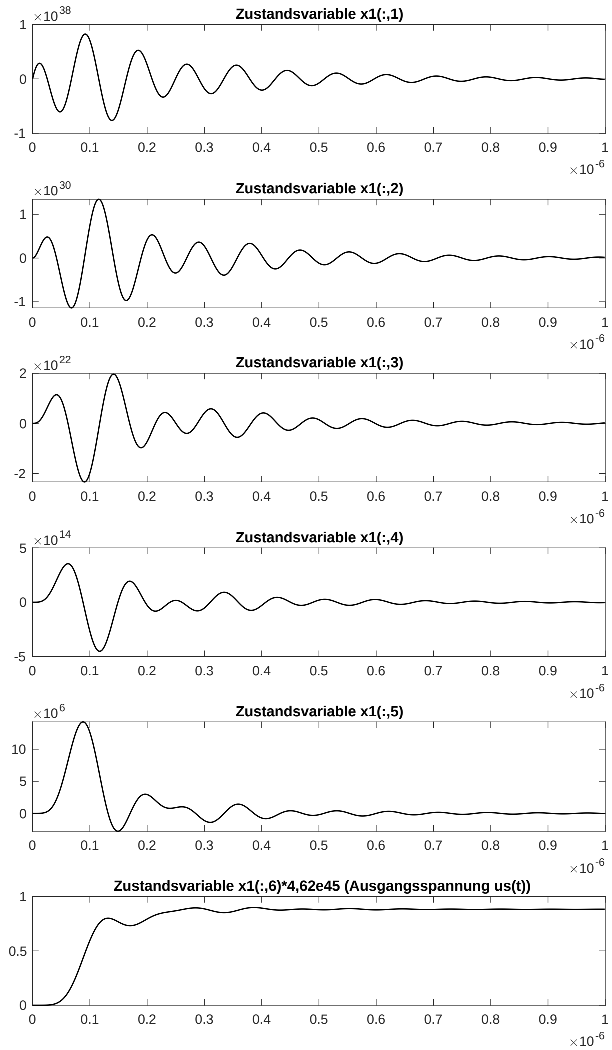

Abb. 5: Zustandsvariablen der Sprungantwort für das Modell system1

Fläche angenähert. Die Annäherung ist gut, wenn der Puls eine Dauer `delta_t` viel kleiner als die Dynamik des Systems ist. Man kann die Impulsantwort mit verschiedenen Dauern ermitteln, bis sich die Antwort nicht mehr ändert.

Die Antwort auf einen beliebigen Eingangssignal, wie z.B. weißes Rauschen, ist im Skript mit folgenden Zeilen programmiert:

```
N = 1000;
rng(1238);              % Startwert für den Rauschgenerator
u = randn(1,N);
t = (0:N-1)*delta_t;
x0 = zeros(1,6);        % Anfangszustandsvektor
```

```
y = lsim(Hs,u,t,x0);    % oder y = lsim(system1,u,t,x0);  bzw. y = lsim(system2,u,t,x0);
```

Für das System wurde hier das Objekt *Transfer function* `Hs` benutzt. Es können auch die Modelle (auch Objekte) `system1` oder `system1` benutzt werden. In Abb. 6 ist das Eingangs- und Ausgangssignal dargestellt.

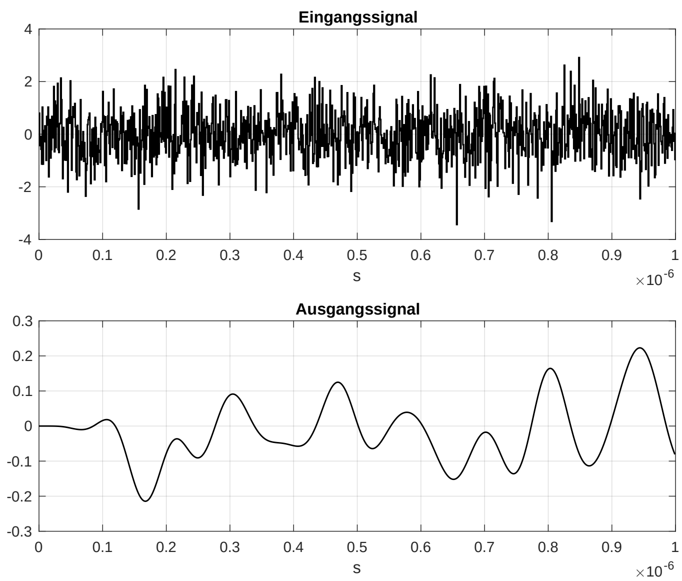

Abb. 6: Antwort des Filters auf weißen Rauschen

Die Antwort auf zwei sinusförmige Signale, wobei das eine Signal im Durchlassbereich und das zweite im Sperrbereich liegen, wird mit

```
fsig1 = 1e6;       ampl1 = 1;  % Signal im Durchlassbereich des Filters
fsig2 = 1e8;       ampl2 = 2;  % Signal im Sperrbereich des Filters
N = 5000;
t = (0:N-1)*delta_t;

u = ampl1*sin(2*pi*fsig1*t) + ampl1*sin(2*pi*fsig2*t);
y = lsim(Hs,u,t,x0);
```

programmiert. In Abb. 7 ist das Eingangs- und Ausgangssignal für diese Anregung dargestellt.
Das eine Signal wird unterdrückt und es bleibt nur das Signal, das im Durchlassbereich liegt, mit der entsprechenden Dämpfung, die man aus dem Frequenzgang entnehmen kann.

1.4 Zusammenfassung

Mit Hilfe eines passiven Tiefpassfilters werden einige MATLAB Funktionen zur Untersuchung von LTI-Systeme eingesetzt. Die Übertragungsfunktion wird im

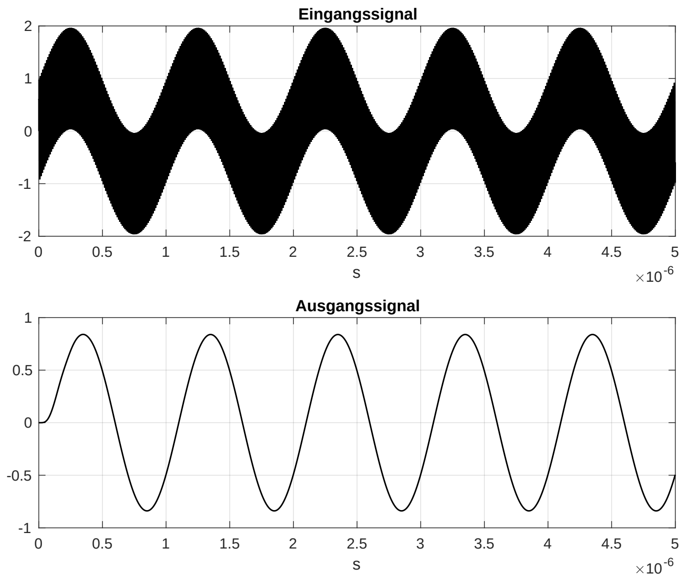

Abb. 7: Antwort des Filters auf zwei sinusförmige Signale

Gegensatz zur üblichen Art vom Eingang als Ursache zum Ausgang als Ergebnis die Gleichungen aufbauen, leichter von Ausgang zum Eingang Gleichungen aufbauen. Die Gleichungen werden mit einer *Transfer function* Variable geschrieben, so dass man am Ende ein Objekt dieser Art erhält. Weitere Funktionen erlauben den Einsatz dieses Objekts ohne dass man die numerischen Koeffizienten der Übertragungsfunktion extrahieren muss. Diese Koeffizienten kann man extrahieren für weitere MATLAB Funktionen, die nur diese Werte akzeptieren.

Der Frequenzgang wird mit der Funktion `freqs` ermittelt und auch direkt numerisch mit Hilfe der komplexen Gleichungen der Schaltung des Filters.

Für die Antwort auf beliebige Anregungen ist das Zustandsmodell des LTI-Systems dargestellt durch das passive Tiefpassfilter sehr wichtig. Man kann ein Zustandsmodell (*State space* mit der Funktion `tf2ss` ermitteln. Man kann in diesem Modell den gelieferten Zustandsvariablen nicht sehr einfach die physikalischen Zustandsvariablen. Diese sind in elektrischen Schaltungen die Spannungen der Kapazitäten und die Ströme der Induktivitäten.

Einen besseren Einblick was ein Zustandmodell bedeutet erhält man, wenn dieses Modell aus den Differentialgleichungen der Schaltung abgeleitet ist. Die Zustandsmodelle sind nicht einzigartig, müssen aber dieselben Ausgangsvariablen liefern. Sie unterscheiden sich nur durch die Zustandsvariablen.

Literatur

[1] Josef Hoffmann, Alfons Klönne: *Wechselstromtechnik. Anwendungsorientierte Simulationen in MATLAB.* Oldenbourg Verlag, 2012.

[2] Josef Hoffmann, Franz Quint: *Einführung in Signale und Systeme. Lineare zeitinvariante Systeme mit anwendungsorientierten Simulationen in MATLAB/Simulink.* Oldenbourg Verlag, 2013.

[3] Schaumann Rolf, Van Valkenburg, Mac E.: *Design of Analog Filters.* Oxford University Press, 2001.

[4] Zverev, Anatol I.: *Handbook of Filter Synthesis.* John Wiley, 1967.