Laborprotokoll SSY

Anwendung von Systemen: Filter

Daniel Schrenk, Andreas Unterweger, ITS 2004

1. Einleitung

Ziel der Übung

Bei dieser Übung ging es um den Vergleich von verschiedenen Filtern mittels Matlab. Ein herkömmliches RC-Glied filtert auch bei höherer Ordnung nur sehr unzureichend (großer Übergangsbereich), weswegen es Filter wie z.B. den Tschebyscheff-Filter gibt, der zwar eine wesentlich steilere Flanke aufweist, was man allerdings mit einer gewissen Welligkeit im Durchlassbereich bezahlen muss.

Hintergrund

Filter sind Übertragungssysteme und können daher auch durch eine Übetragungsfunktion (vgl. erstes Laborprotokoll) beschrieben werden. Matlab bietet für die hier betrachteten Filter Funktionen an, die anhand von Parametern (z.B. Ordnung des Filters) entsprechende Übertragungsfunktionen generieren, die zur weiteren Betrachtung (z.B. Pol-Nullstellendiagramm, vgl. zweites Laborprotokoll) verwendet werden können.

2. Aufgabenstellungen

1.1 Kritisch gedämpfter Tiefpass (RC)

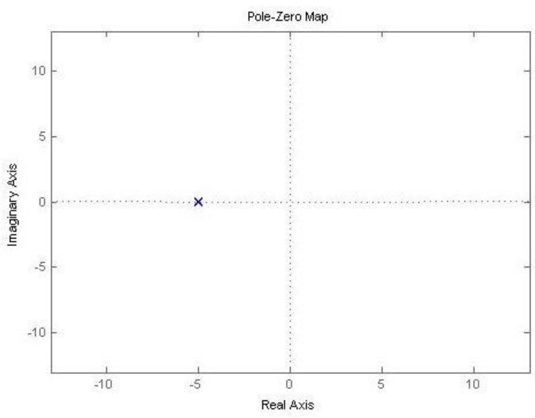
Der wohl einfachste Filter ist ein RC-Tiefpass, der (in der Praxis) auf einem Widerstand und einem Kondensator besteht. Die Übertragungsfunktion lautet $\frac{1}{RCs+1}$ wobei RC auch als Abklingkonstante Tau (τ) bezeichnet wird. Tau soll in diesem Beispiel 0,2 sein, die Transferfunktion lautet also $\frac{1}{0{,}2s+1}$ bzw. in Matlab-Schreibweise

```
G = tf(1, [0.2 1])
```

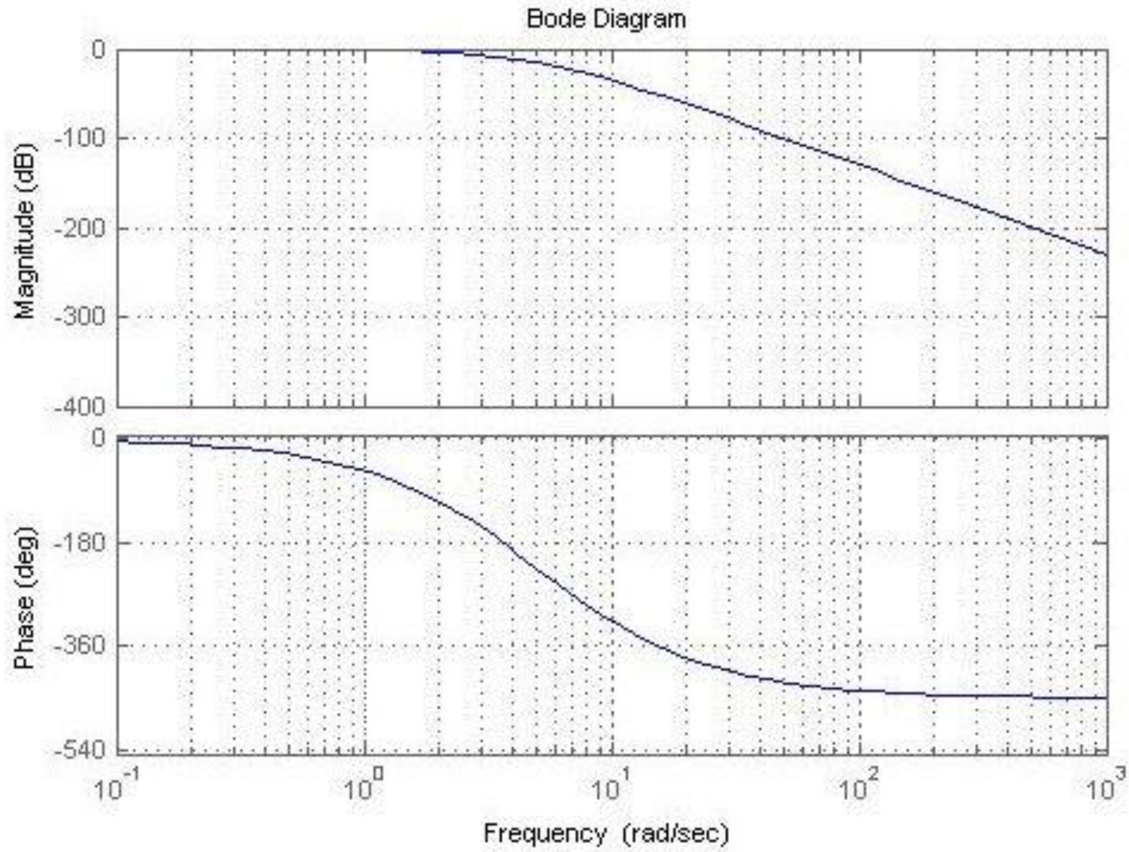
Da es ein Tiefpass 5. Ordnung sein soll, wird G fünfmal mit sich selbst multipliziert:

```
gOrdnung5 = G^5
```

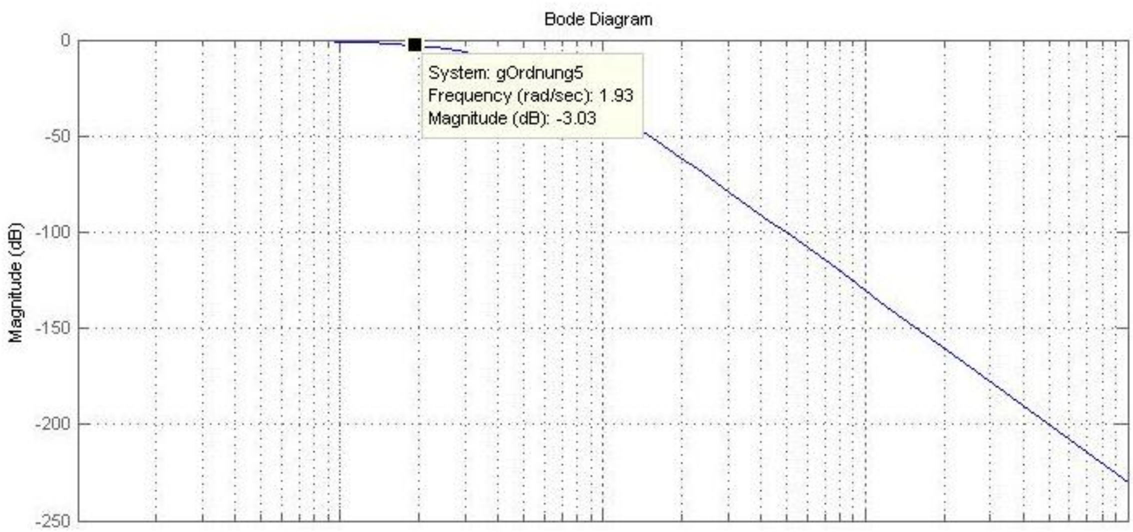
Das Pol-Nullenstellen-Diagramm weist eine fünffache Polstelle bei -5 auf (Nenner 0):



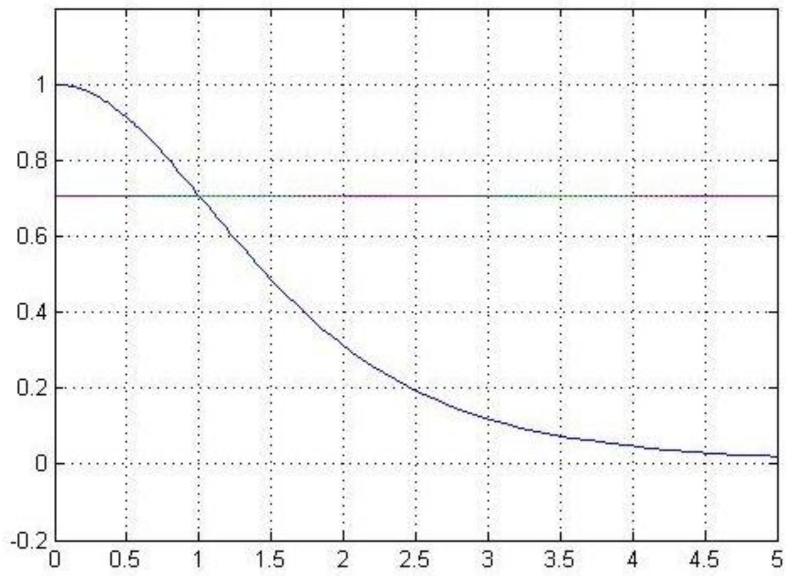
Das Bode-Diagramm zeigt das für einen RC-Tiefpass typische Bild: eine Dämpfung von 20dB pro Dekade sowie einen Phasenverlauf, der bedingt durch die Ordnung 5 (entspricht 5 hintereinandergeschalteten RC-Tiefpässen) ein Phasenverschiebung gegen 5*(-90) = -450° aufweist.



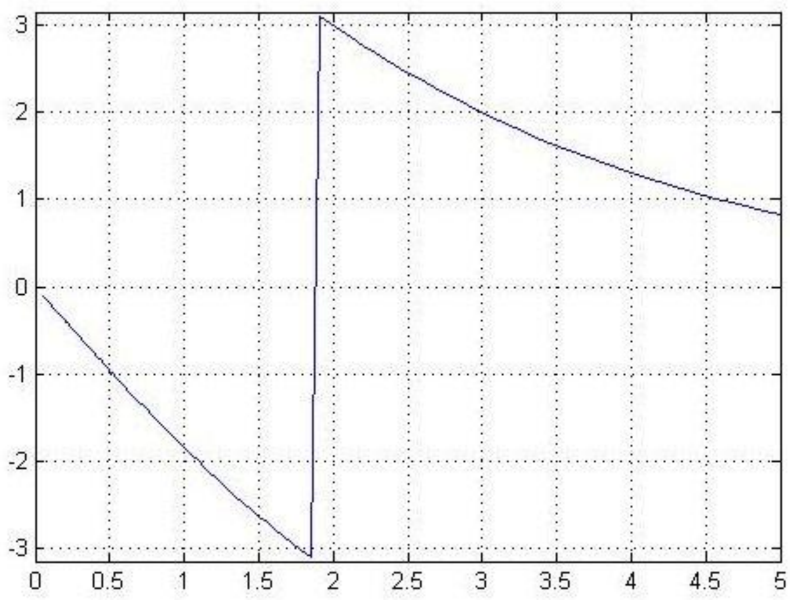
In der Grafik lässt sich durch das Setzen und Auslesen eines Markers jener Punkt bestimmen, an dem die Dämpfung 3dB beträgt (= Grenzfrequenz). Die nachstehende Grafik zeigt den oberen Teil des Bodediagramms mit dem Marker – die Grenzfrequenz liegt bei 1,93 rad/s – um den Tiefpass mit den anderen (auf 1 rad/s genormten) Filtern vergleichen zu können, wird nachfolgend jeder Wert im linearen (selbst erstellten, vgl. unten) „Bode"-Diagramm von w durch 1,93 dividiert.



Die nachstehenden Grafiken zeigen ähnlich dem Bodediagramm den durch den Filter hervorgerufenen Betrags- und Phasenverlauf – mit dem Unterschied, dass die Skalierung linear ist:



Auf der x-Achse ist die normierte Frequenz Ω aufgetragen, auf der y-Achse die relative Amplitude zu der des Eingangssignals (1 = keine Veränderung, 0,5 = halbe Amplitude, … etc.). Die horizontale Linie markiert die Dämpfung von 3dB (= $10^{\frac{-3}{20}} = \frac{1}{\sqrt{2}}$); am Schnittpunkt dieser Linie mit dem Filterverlauf lässt sich die Grenzfrequenz (1) ablesen.



Diese Grafik zeigt den Phasenverlauf (ebenfalls mit linearen Achsen). Der Sprung ist dadurch zu erklären, dass die angle-Funktion von Matlab nur Werte zwischen –Pi und Pi zurückgibt, was zu einem Sprung führt, wenn dieser Wertebereich über- bzw. unterschritten wird.

Zum Plotten dieses linearen „Bode"-Diagramms ist es notwendig, die Übertragungsfunktion zu teilen (in Zähler und Nenner); der Zähler ist 1, der Nenner kann durch die get-Funktion mit dem Parameter 'den' (= denominator = Nenner) extrahiert werden. Da die Get-Funktion eine sogenannte Cell zurückgibt, muss der eigentliche Nenner aus deren ersten Feld gelesen werden ({1}).

```
zaehler = 1
nenner_cell = get(gOrdnung5, 'den')
nenner = nenner_cell{1}
```

Mithilfe der freqs-Funktion werden für bestimmte w Betrag und Phase in Form einer komplexen Zahl berechnet (H). Um an Betrag und Phase von H zu gelangen, werden die beiden Funktionen abs bzw. angle verwendet.
Wie oben erwähnt, muss zum Plotten der Wert von w durch 1,93 dividiert werden, um einen Vergleich mit den auf 1 normierten, vordefinierten Filtern zu ermöglichen:

```
[H, w] = freqs(zaehler, nenner)
H_betrag = abs(H)
H_phase = angle(H)
subplot(2, 2, 3)
plot(w / 1.93, H_betrag)
```

Der vollständige Matlab-Code für die durchgeführten Berechnungen lautet:

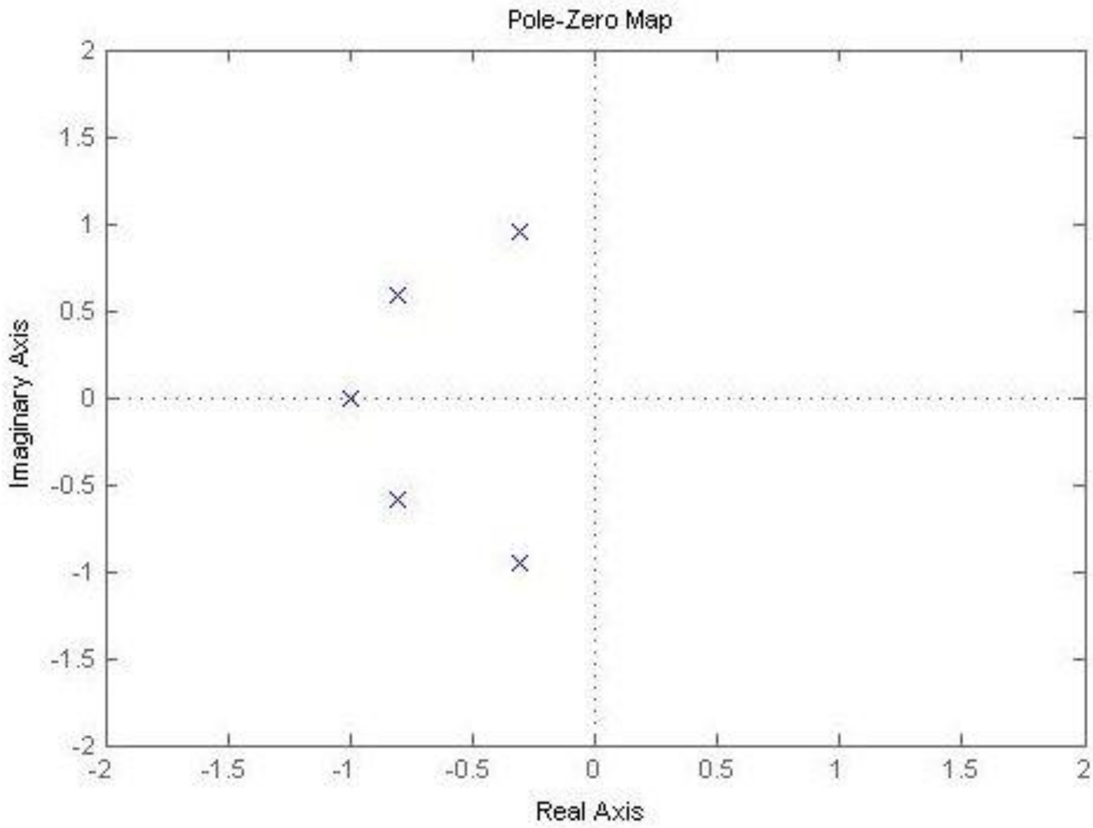
```
G = tf(1, [0.2 1]) %RC-Tiefpass (Tau = R*C)
gOrdnung5 = G^5 %5. Ordnung
subplot(2, 2, 1)
pzmap(gOrdnung5) %PN-Diagramm
axis([-13 13 -13 13]) %Achsenbereich ändern
%figure %Neues Fenster
subplot(2, 2, 2)
bode(gOrdnung5) %Bode-Diagramm
grid on %Gitter anzeigen
zaehler = 1 %Zähler von G
nenner_cell = get(gOrdnung5, 'den') %Nenner von G (Cell-Format)
nenner = nenner_cell{1} %Nenner ("herkömmliches" Format)
[H, w] = freqs(zaehler, nenner) %Bode händisch berechnen
H_betrag = abs(H) %Betrag
H_phase = angle(H) %Phase
subplot(2, 2, 3)
plot(w / 1.93, H_betrag) %Oberen Teil des Bodediagramms plotten %w / 1,93 für Normierung!
axis([0 5 -0.2 1.2]) %Achsenbereich anpassen
grid on %Gitter anzeigen
hold on %Gerade in selbe Grafik einzeichnen
plot(0:0.001:5, 1/sqrt(2)) %Grenzfrequenz ist hier bei 10^(-3dB/20) = 1/sqrt(2)
hold off
subplot(2, 2, 4)
%figure %Neues Fenster
plot(w / 1.93, H_phase) %Unteren Teil des Bodediagramms plotten % w / 1,93 für Normierung!
axis([0 5 -pi pi]) %Achsenbereich anpassen (Sprung bei -pi wegen Phase!)
grid on %Gitter anzeigen
```

1.2 Butterworth-Tiefpass

Nun wurden zu Vergleichszwecken die selben Grafiken wie in 1.1 von einem Butterworth-Filter 5. Ordnung erstellt. Die Matlab-Funktion buttap erzeugt einen solchen Filter, gibt aber die Pol- und Nullstellen anstelle einer Übertragungsfunktion zurück. Daher müssen die Pol- und Nullstellen zusätzlich mit der Funktion zp2tf in zwei Übertragungsfunktionen (je eine für Zähler und Nenner) umgewandelt und anschließend zu einer einzigen Übertragungsfunktion zusammengefasst werden:

```
[z, p, k] = buttap(5)
[zaehler, nenner] = zp2tf(z, p, k)
gOrdnung5 = tf(zaehler, nenner)
```

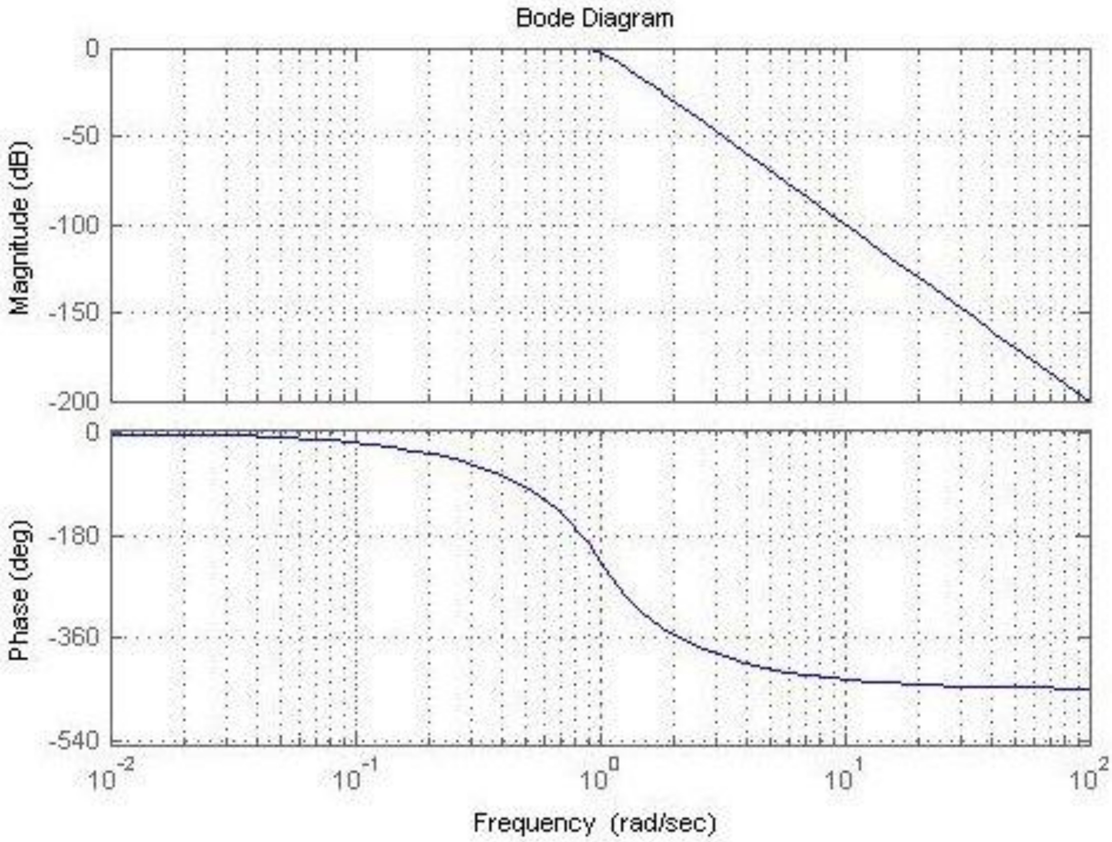
Es ergibt sich folgendes (für den Butterworth-Filter typisches) Pol-Nullstellen-Diagramm:



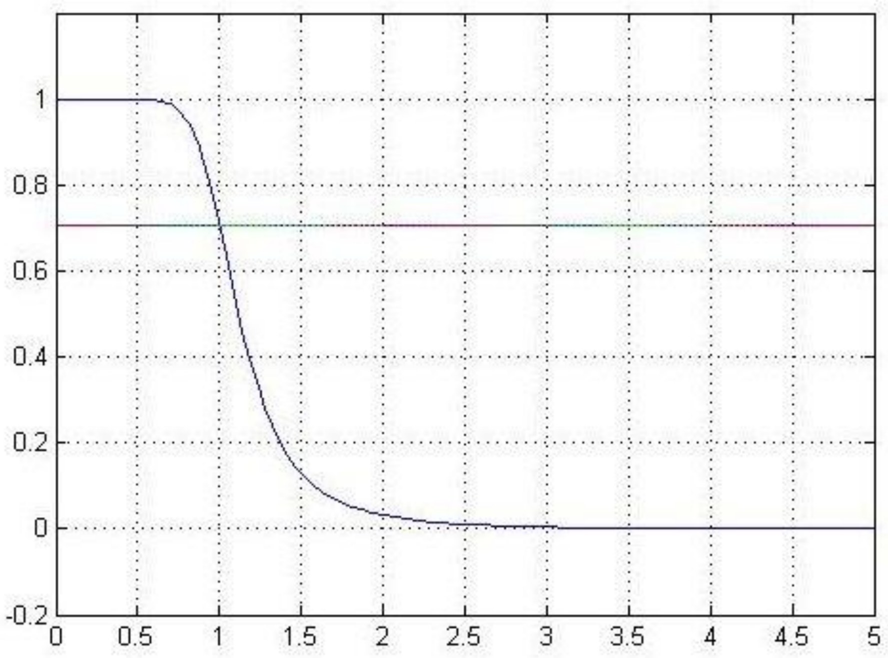
Charakteristisch sind hier die halbkreisförmig angeordneten Polstellen links von der imaginären Achse. Wären sie rechts davon, wäre das System instabil (vgl. zweites Laborprotokoll).

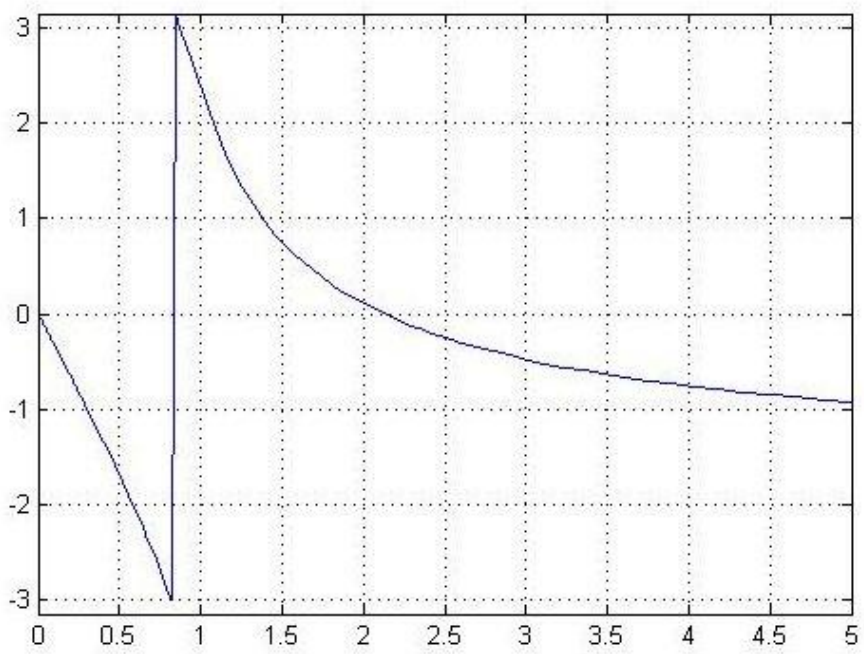
Das Bodediagramm (siehe unten) zeigt relativ deutlich, dass der Butterworth-Filter eine steilere Flanke aufweist als der RC-Tiefpass – noch besser ist das im Diagramm mit linearer Skala sichtbar. Der Phasenverlauf ist sehr ähnlich.

Bode-Diagramm:



Lineares „Bode"-Diagramm (zu den Skalen auf den Achsen siehe 1.1):





Der vollständige Matlab-Code für die durchgeführten Berechnungen lautet:

```
[z, p, k] = buttap(5) %Butterworth-Filter 5. Ordnung
[zaehler, nenner] = zp2tf(z, p, k) %Umwandeln in Transferfunktion
gOrdnung5 = tf(zaehler, nenner) %Als Transferfunktion darstellen
subplot(2, 2, 1)
pzmap(gOrdnung5) %PN-Diagramm
axis([-2 2 -2 2]) %Achsenbereich ändern
%figure %Neues Fenster
subplot(2, 2, 2)
bode(gOrdnung5) %Bode-Diagramm
grid on %Gitter anzeigen
[H, w] = freqs(zaehler, nenner) %Bode händisch berechnen
H_betrag = abs(H) %Betrag
H_phase = angle(H) %Phase
subplot(2, 2, 3)
plot(w, H_betrag) %Oberen Teil des Bodediagramms plotten
axis([0 5 -0.2 1.2]) %Achsenbereich anpassen
grid on %Gitter anzeigen
hold on %Gerade in selbe Grafik einzeichnen
plot(0:0.001:5, 1/sqrt(2)) %Grenzfrequenz ist hier bei 10^(-3dB/20) = 1/sqrt(2)
hold off
subplot(2, 2, 4)
%figure %Neues Fenster
plot(w, H_phase) %Unteren Teil des Bodediagramms plotten
axis([0 5 -pi pi]) %Achsenbereich anpassen (Sprung bei -pi wegen Phase!)
grid on %Gitter anzeigen
```

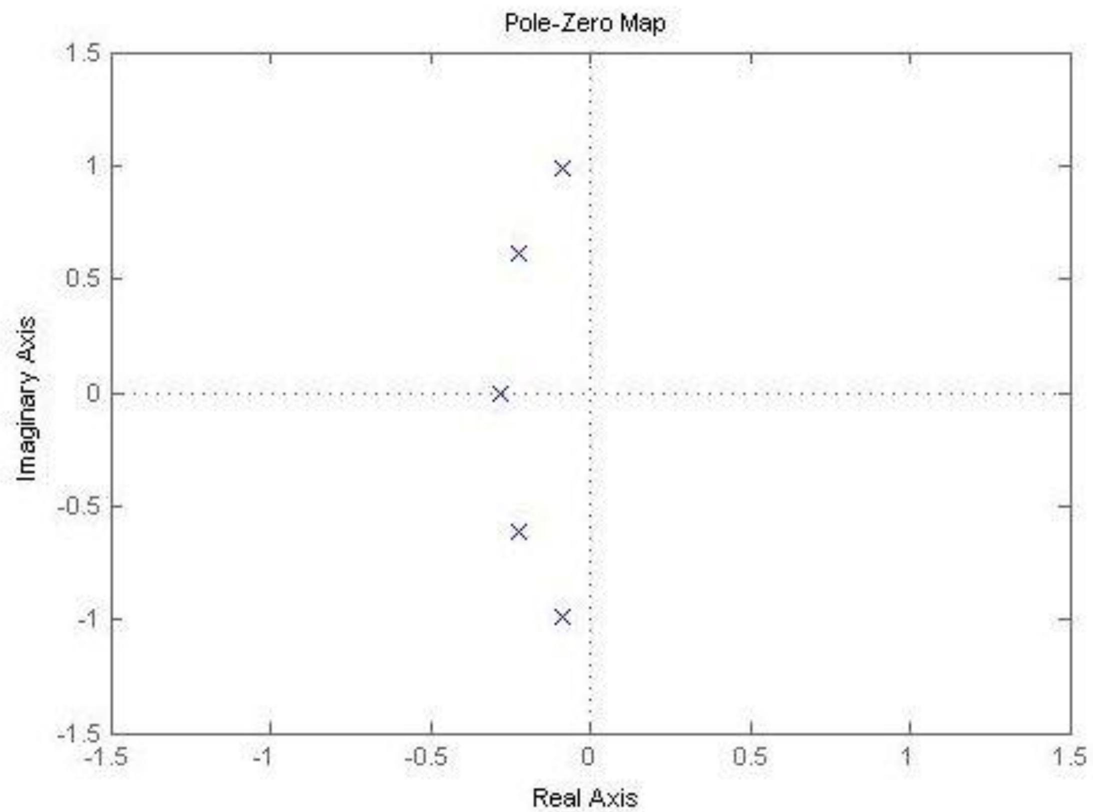
1.3 Tschebyscheff-Tiefpass

Analog zum Butterworth- sollte nun der Tschebyscheff-Filter untersucht werden. Matlab stellt für die Erzeugung von letzterem die Funktion cheb1ap bereit, die (zusätzlich zur Ordnung) die Welligkeit als Parameter erwartet – letztere sollte in unserem Fall 1dB sein:

```
[z, p, k] = cheb1ap(5, 2^(1 / 6))
```

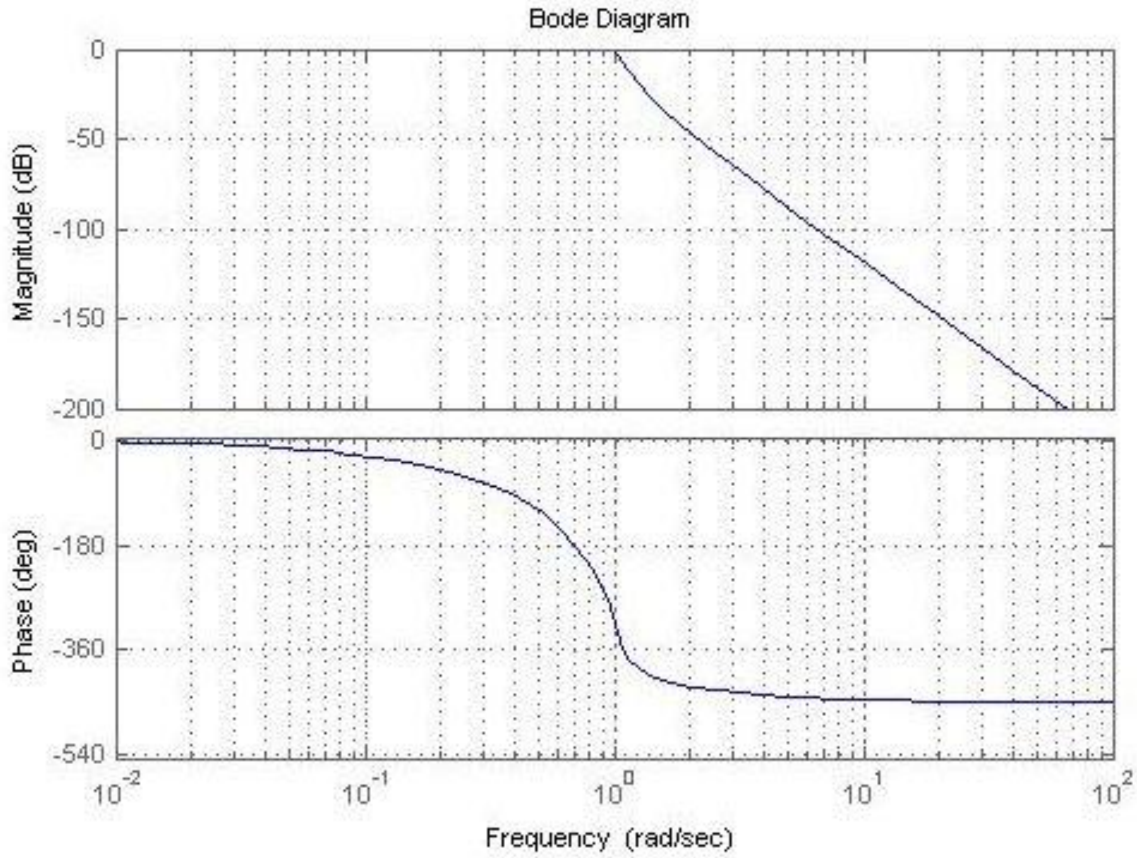
Da die Funktion die Welligkeit nicht in dB erwartet, muss der Wert entsprechend umgerechnet werden: da 3dB = $\frac{1}{\sqrt{2}}$ = 1dB + 1dB + 1dB folgt x*x*x = $\frac{1}{\sqrt{2}}$ (Addition wird zur Multiplikation). x ist daher $\frac{1}{\sqrt[6]{2}}$ bzw. $2^{\frac{1}{6}}$.

Das Pol-Nullstellen-Diagramm zeigt eine elliptische Anordnung der Polstellen (vgl. kreisförmige Anordnung bei Butterworth in 1.2):

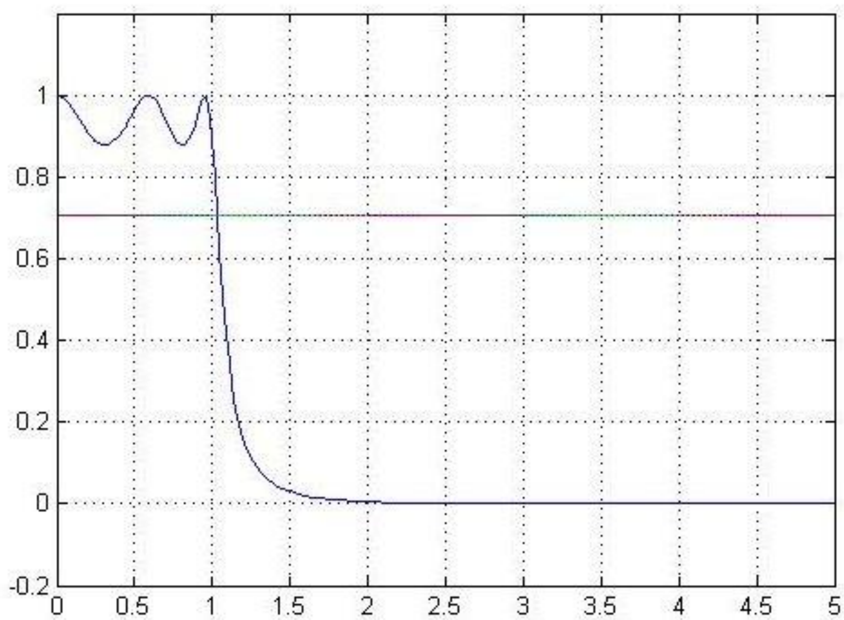


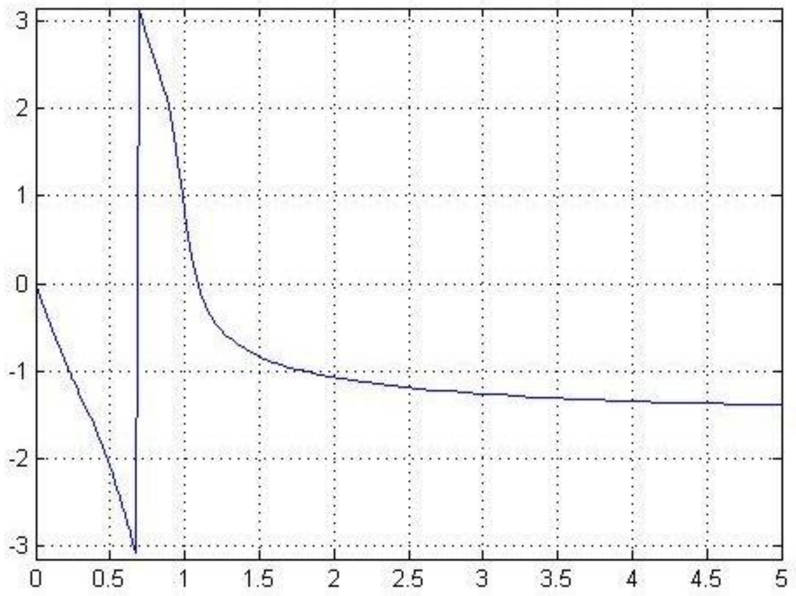
Diese Anordnung erlaubt zwar eine noch steilere Flanke (vgl. linearer Betragsverlauf unten), wird allerdings durch eine Welligkeit im Durchlassbereich erkauft (sehr schön im linearen „Bode"-Diagramm zu erkennen). Dies kann nachteilig sein, ist aber in manchen Fällen nicht so relevant wie die Steilheit der Flanke.

Bode-Diagramm:



Lineares „Bode"-Diagramm:





Der vollständige Matlab-Code für die durchgeführten Berechnungen lautet:

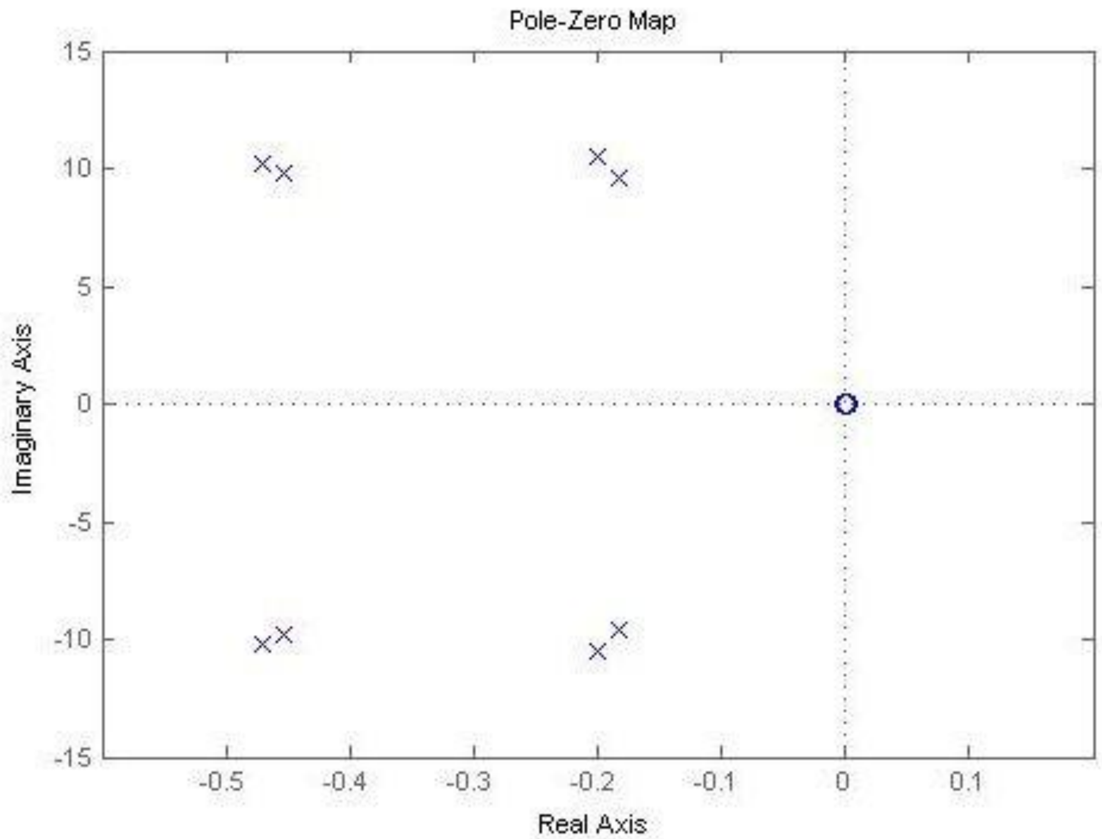
```
[z, p, k] = cheb1ap(5, 2^(1 / 6)) %Tschebyscheff-Filter 5. Ordnung
[zaehler, nenner] = zp2tf(z, p, k) %Umwandeln in Transferfunktion
gOrdnung5 = tf(zaehler, nenner) %Als Transferfunktion darstellen
subplot(2, 2, 1)
pzmap(gOrdnung5) %PN-Diagramm
axis([-1.5 1.5 -1.5 1.5]) %Achsenbereich ändern
%figure %Neues Fenster
subplot(2, 2, 2)
bode(gOrdnung5) %Bode-Diagramm
grid on %Gitter anzeigen
[H, w] = freqs(zaehler, nenner) %Bode händisch berechnen
H_betrag = abs(H) %Betrag
H_phase = angle(H) %Phase
subplot(2, 2, 3)
plot(w, H_betrag) %Oberen Teil des Bodediagramms plotten
axis([0 5 -0.2 1.2]) %Achsenbereich anpassen
grid on %Gitter anzeigen
hold on %Gerade in selbe Grafik einzeichnen
plot(0:0.001:5, 1/sqrt(2)) %Grenzfrequenz ist hier bei 10^(-3dB/20) = 1/sqrt(2)
hold off
subplot(2, 2, 4)
%figure %Neues Fenster
plot(w, H_phase) %Unteren Teil des Bodediagramms plotten
axis([0 5 -pi pi]) %Achsenbereich anpassen (Sprung bei -pi wegen Phase!)
grid on %Gitter anzeigen
```

1.4 Butterworth-Bandpass

Zuletzt sollte noch ein Butterworth-Bandpass betrachtet werden. Ein Bandpass lässt sich mit der Matlab-Funktion lp2bp sehr einfach aus einem Tiefpass erzeugen. Als zusätzliche Parameter benötigt diese Funktion die Grenzfrequenz (10Hz) und die Bandbreite (1Hz).

```
[z, p, k] = buttap(4)
[zaehler, nenner] = zp2tf(z, p, k)
[zaehler, nenner] = lp2bp(zaehler, nenner, 10, 1)
```

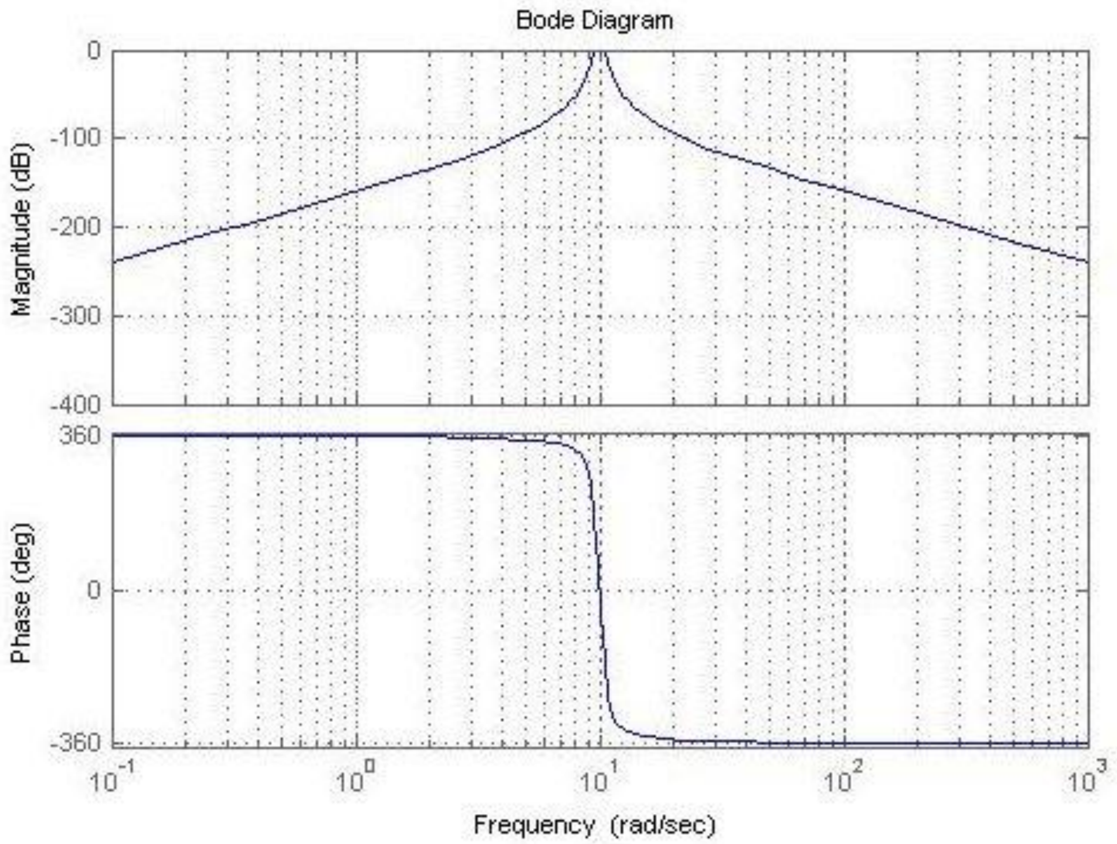
Das Pol-Nullstellen-Diagramm zeigt hier neben einer mehrfachen Nullstelle im Ursprung auch mehrere konjugiert komplexe Polstellen links von der imaginären Achse:



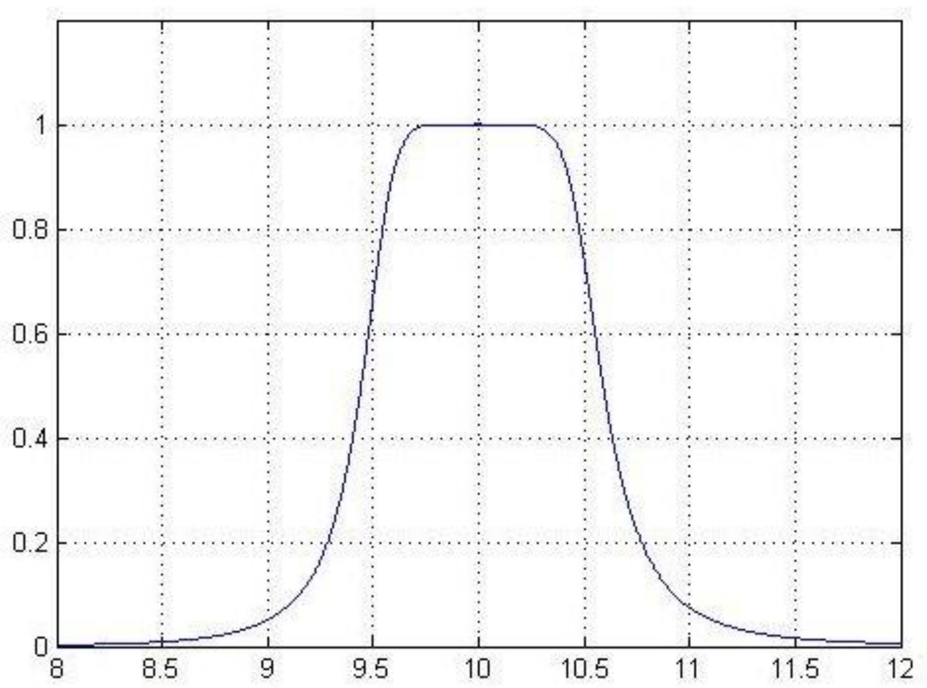
Das lineare „Bode"-Diagramm (vgl. unten) zeigt eine sehr starke Dämpfung außerhalb des Bandes (Anmerkung: die Zeichengenauigkeit lässt sich durch Verkleinerung der Intervalle (vgl. Matlab-Code unten) erhöhen). Auch im Bode-Diagramm ist das sichtbar (über 200dB Dämpfung 2 Dekaden vor bzw. nach dem Band).

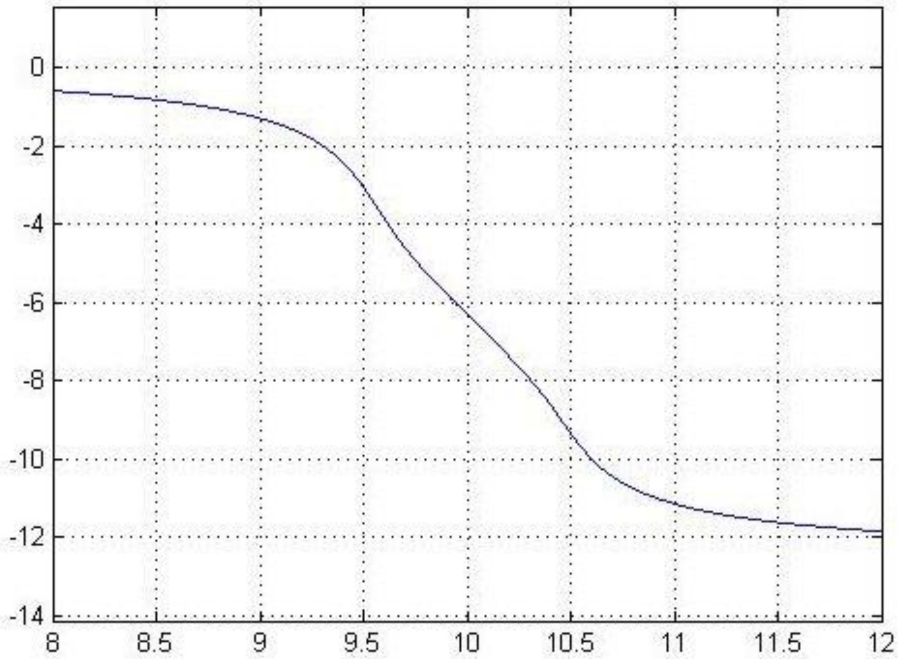
Anmerkung: zur besseren Übersichtlichkeit wurde im Phasenverlauf mit linearer Skala die Funktion unwrap verwendet, um Sprünge durch Über- bzw. Unterschreiten des Wertebereichs (-Pi bis Pi) zu vermeiden.

Bode-Diagramm:



Lineares „Bode"-Diagramm:

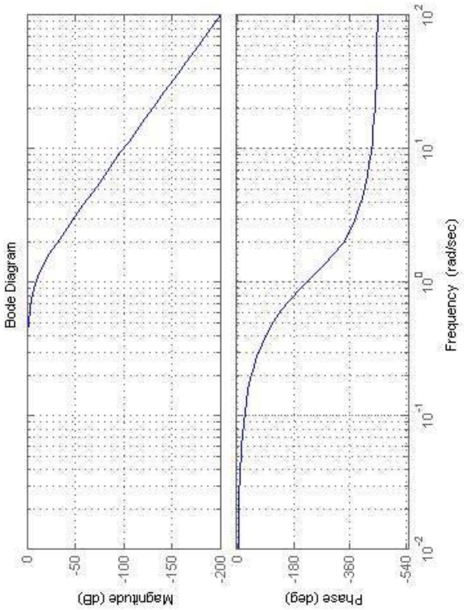


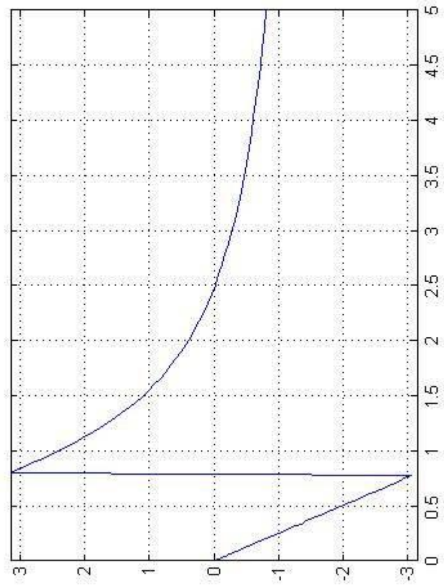


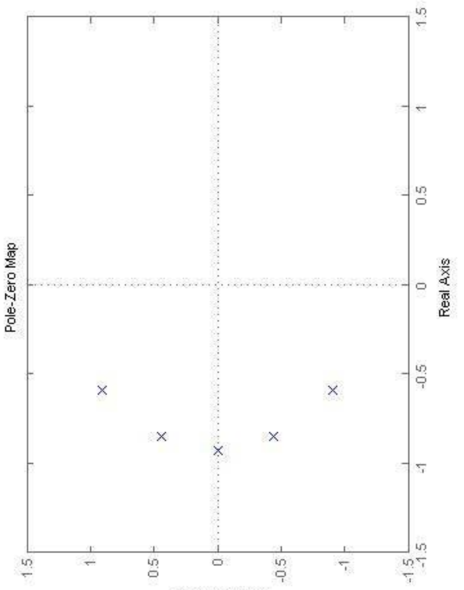
Der vollständige Matlab-Code für die durchgeführten Berechnungen lautet:

```
[z, p, k] = buttap(4) %Butterworth-Filter 4. Ordnung
[zaehler, nenner] = zp2tf(z, p, k) %Umwandeln in Transferfunktion
[zaehler, nenner] = lp2bp(zaehler, nenner, 10, 1) %TP in Bandpass umwandeln
gOrdnung5 = tf(zaehler, nenner) %Als Transferfunktion darstellen
subplot(2, 2, 1)
pzmap(gOrdnung5) %PN-Diagramm
axis([-0.6 0.2 -15 15]) %Achsenbereich ändern
%figure %Neues Fenster
subplot(2, 2, 2)
bode(gOrdnung5) %Bode-Diagramm
grid on %Gitter anzeigen
[H, w] = freqs(zaehler, nenner, 0:0.001:20) %Bode händisch berechnen (höhere Genauigkeit!)
H_betrag = abs(H) %Betrag
H_phase = angle(H) %Phase
subplot(2, 2, 3)
plot(w, H_betrag) %Oberen Teil des Bodediagramms plotten
axis([8 12 0 1.2]) %Achsenbereich anpassen
grid on %Gitter anzeigen
subplot(2, 2, 4)
%figure %Neues Fenster
plot(w, unwrap(H_phase)) %Unteren Teil des Bodediagramms plotten
axis([8 12 -4.5*pi pi/2]) %Achsenbereich anpassen (Sprung bei -pi wegen Phase!)
grid on %Gitter anzeigen
```

Anmerkung: der Bessel-Filter wurde zwar ebenfalls berechnet, der Übersichtlichkeit halber aber nicht mehr im Detail dem Protokoll hinzugefügt. Auf der nächsten Seite findet sich ein Plot mit allen 4 Grafiken (wie in 2.1-2.3), allerdings ohne Matlab-Code. Letzterer unterscheidet sich vom Butterworth-Code nur durch den Namen der Funktion, die den Filter erzeugt: `[z, p, k] = besselap(5)`
Der Bessel-Filter ist etwas steiler als der RC-Tiefpass, weist aber einen schöneren Phasenverlauf auf, was durch Welligkeit im Durchlass- und im Sperrbereich erkauft wird.


Bode Diagram
Magnitude (dB)
Phase (deg)
Frequency (rad/sec)




Pole-Zero Map
Real Axis
Imaginary Axis