Ereignisdiskrete Systeme

Praktikum Blatt 1

Jan Kristel, Alexandra Moritz

Aufsicht von Frau Rembold

22. April 2023

Inhaltsverzeichnis

1 Aufgabe 1: MATLAB Grundlagen

1.1 Was ist MATLAB?

Matlab ist eine Hochleistungssprache für technisches Rechnen. Es integriert Berechnung, Visualisierung und Programmierung in einer benutzerfreundlichen Umgebung, in der Probleme und Lösungen in vertrauter mathematischer Notation ausgedrückt werden.
Typische Verwendungen sind:

- Mathematik und Rechnen
- Entwicklung von Algorithmen
- Modellierungssimulation und Visualisierung
- wissenschaftliche und technische Grafiken

Dies ermöglicht es Ihnen, viele, technische Rechenprobleme, insbesondere solche mit Matrix- und Vektorformulierungen, in einem Bruchteil der Zeit zu lösen, die benötigt würde, um ein Programm in einer skalaren, nicht-interaktiven Sprache wie C oder Fortan zu schreiben. Der Name Matlab steht für Matrixlabor.

1.2 Wesentliche Komponenten der MATLAB-Oberfläche

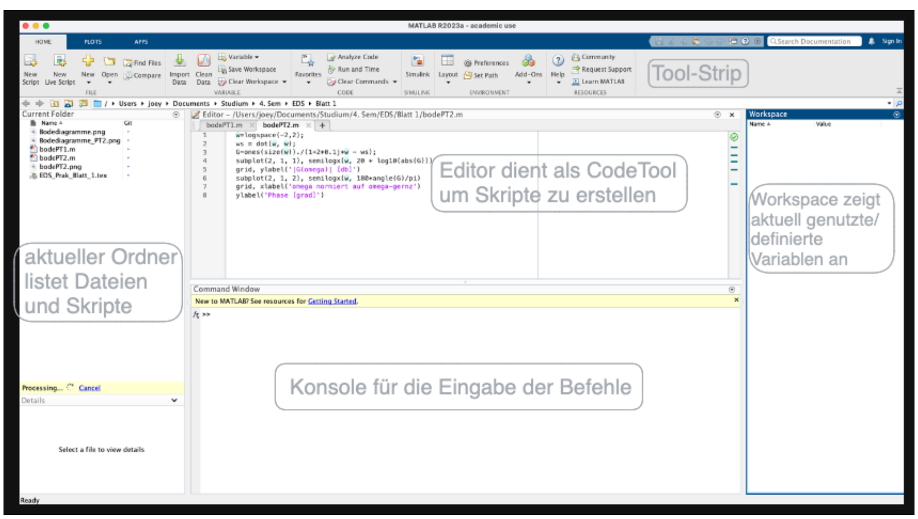


Abbildung 1: MATLAB Oberfläche

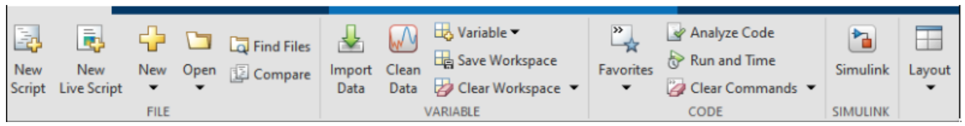


Abbildung 2: Tool-Strip im Reiter ”Home”. Hier lassen sich neue Skripte erstellen oder vorhandene öffnen.

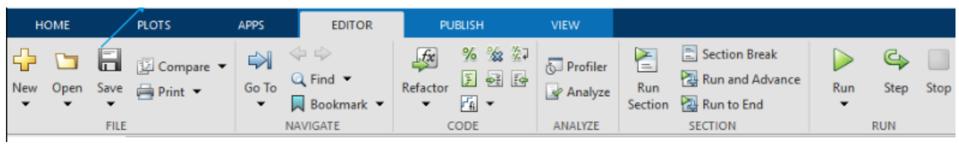


Abbildung 3: Tool-Strip im Reiter Ëditor”. Mit ”Run”lassen sich die Programme/Skripte starten, die man im Editor erstellt hat.

1.3 *Current Folder Browser* - Wozu? Was ist zu beachten?

Der Current Folder Browser zeigt Ordner und Dateien im aktuellen Arbeitsverzeichnis an. Man braucht es um den Zugriff auf die Dateien und Skripte zu erleichtern und um sicherzustellen das MATLAB die Dateien findet. Es ist wichtig die Dateien in der richtigen Stelle zu speichern, da die MATLAB-Programme standardgemäß im aktuellen Arbeitsverzeichnis ausgeführt werden.

1.4 *Comand Window* - Was verbirgt sich dahinter?

Das Command Window ist eine Konsole, die für die Eingabe von dem Benutzer verwendet wird und hier werden die MATLAB-Befehle eingegeben. Es wird auch eine Historie der davor eingegebenen Befehle gespeichert, sodass man sie immer wieder benutzen kann.

1.5 *Tool-Strip* - Was verbirgt sich dahinter?

Der Tool-Strip ist eine Symbolleiste, hier findet man auf häufig verwendete Funktionen. Es bietet beispielsweise schnellen Zugriff auf das Command Window oder die Hilfe-Funktionen.

1.6 Zweck des *Workspaces*

Der Workspace zeigt die aktuellen Variablen und ihre Werte an. Man kann es vergleichen mit einer Variablenansicht in anderen Programmierumgebungen. Wird diser gelöscht bzw. *gecleared*, sind alle benutzten Variablen nicht mehr definiert und nachfolgende Eingaben oder Scripte, die diese Variablen benutzten oder benutzt haben bringen einen Fehler.

1.7 Möglichkeiten Information von *MATLAB-Hilfe* zu bekommen

- Die Online-Hilfe von MATLAB, die über der Webseite erreichbar ist
- in MATLAB direkt die Hilfefunktion, die über der Schaltfläche „Hilfe“ auf der Symbolleiste aufgerufen werden kann. Oder durch Eingabe in der Kommandozeile mit „help“.

1.8 Simulink

- MATLAB öffnen
- in der Kommandozeile ßimulinkëingeben
- oder über die Symbolleiste (Tool-Strip) Simulink starten

1.9 *Control System Tollbox* - Was ist das? Wo findet man sie?

Die *Control System Toolbox* ist eine Add-On Bibliothek für MATLAB. Sie bietet Algorithmen und Apps zum systematischen Analysieren, Entwerfen und Optimieren linearer Steuerungssysteme. Sie können Ihr System als Übertragungsfunktion, Zustandsraum, Nullpolverstärkung oder Frequenzgangmodell spezifizieren. Beispielswiese mit dem Sprungantwortdiagramm und dem Bode-Diagramm lässt sich das Systemverhalten im Zeit- und Frequenzbereich analysieren und visualisieren.
Die Toolbox stimmt automatisch sowohl SISO- als auch MIMO-Kompensatoren ab, einschließlich PID-Regler. Sie können verstärkungsgeplante Regler optimieren und mehrere Optimierungsziele festlegen. Man kann Designs validieren, indem man Anstiegszeit, Überschwingen, Einschwingzeit, Verstärkungs- und Phasenreserven und andere Anforderungen überprüfen.
Zu finden ist die Erweiterung im *AddOn-Explorer*. Diesen wird geöffnet in dem man im Reiter "Homeïm Tool-Strip auf den Button *AddOns* klickt.

1.10 Stateflow

- MATLAB öffnen
- in der Kommandozeile ßtateflowëingeben und Ënter"drücken.

2 Aufgabe 2: Bodediagramme

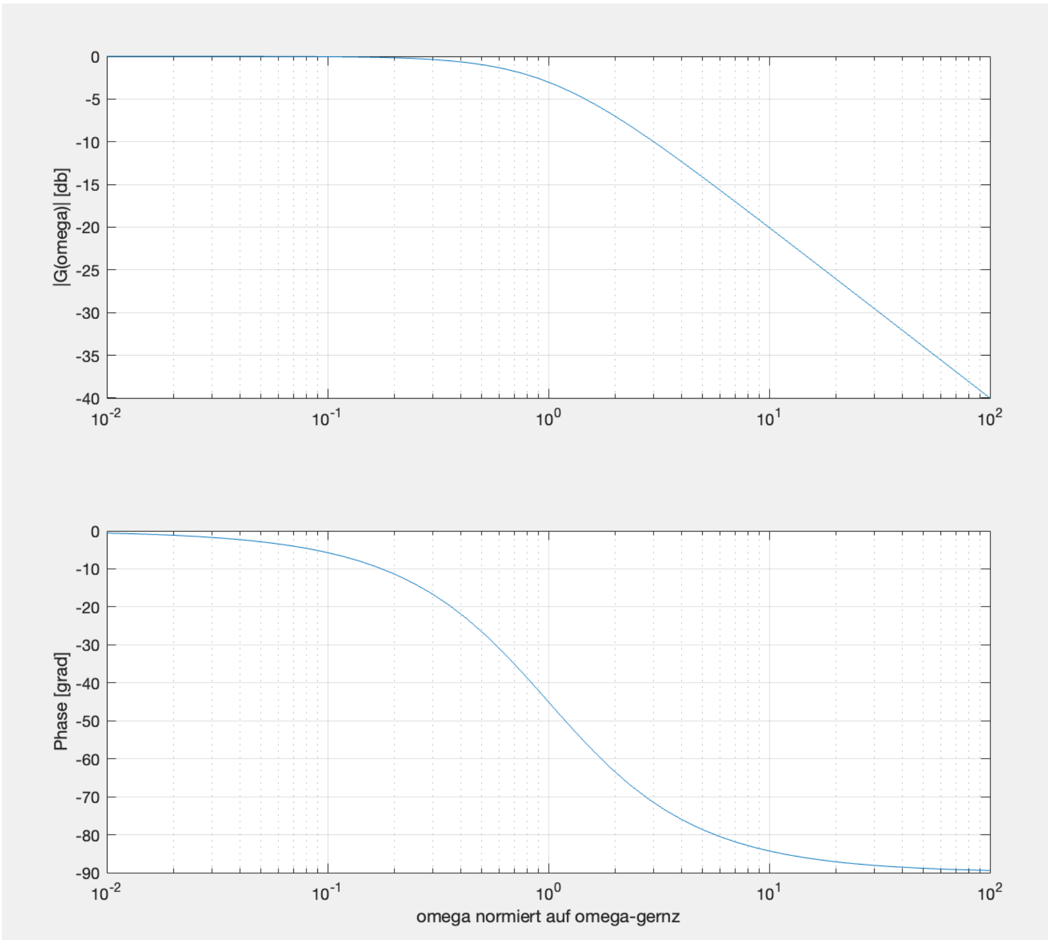


Abbildung 4: Bodediagramme entstanden aus bodePT1.m

```
w = logspace(-2,2);
G = ones(size (w))./(1+j*w);
subplot(2, 1, 1), semilogx(w, 20 * log10(abs(G)))
grid, ylabel('|G(w)| [db]')
subplot(2, 1, 2), semilogx(w, 180*angle(G)/pi)
grid, xlabel('w normiert auf w-gernz')
ylabel('Phase [grad]')
```

Der oben gegebene Code erzeugt für ein PT1-Glied das zughörige Bodediagramm (Abb.4).

2.1 Normierter Frequenzgang PT1-Glied

Um die normierte Form der Übertragsfunktion zu bekommen, nimmt statt der Variablen **s** im laplace-transformiereten Bereich, die im komplexen und Frequenzbereich $j\omega$.

$$G(j\omega) = \frac{K_p}{\left(1 + j\omega \cdot T\right)}$$

2.2 Normierter Frequenzgang PT2-Glied

$$G(j\omega) = \frac{K_p}{(j\omega)^2 + 2 \cdot d \cdot \omega_0 \cdot j\omega + \omega_0^2}$$

2.3 Bodediagramm PT2-Glied

Für die Darstellung eines PT2-Glied wird der, unter Punkt 2 gezeigte Code wie folgt angepasst.

```
1           Kp = 1;
2           D = 0.1;
3
4           w=logspace(-2,2);
5           s= 1j .* w;
6           G=ones(Kp)./(1 + 2*D*s - w.^2);
7           subplot(2, 1, 1), semilogx(w, 20 * log10(abs(G)))
8           grid, ylabel('|G(omega)| [db]')
9           subplot(2, 1, 2), semilogx(w, 180*angle(G)/pi)
10          grid, xlabel('omega normiert auf omega-gernz')
11          ylabel('Phase [grad]')
```

Die Zeile, auf die es zu Achten gilt ist Zeile 6. Hier musste die Formel von einem PT1-Glied angepasst werden für ein PT2-Glied.
Die Übertragsfunktion für ein PT2-Glied sieht folgender Maßen aus:

$$G(s) = \frac{Kp \cdot \omega_0^2}{s^2 + 2 \cdot D \cdot \omega_0^2 \cdot s + \omega_0^2}$$

Diese Formel wird für den Code angepasst. Die Resonanzfrequenz ω_0^2 beginnt hier bei 1. s ist definiert aus durch $s = 1j \cdot \omega$. Da in der Formel s^2 verwendet wird verändert sich der Nenner der Übertragungsfunktion

$$s^2 + 2 \cdot D \cdot s + 1$$

$$= (j\omega)^2 + 2 \cdot D \cdot s + 1$$

$$= j^2 \cdot \omega^2 + 2 \cdot D \cdot s + 1$$

$$= -\omega^2 + 2 \cdot D \cdot s + 1$$

$$= 1 + 2 \cdot D \cdot s - \omega^2$$

Diese Zeile in die Übertragungsfunktion eingesetzt, sieht dann wie folgt aus:

$$G(s) = \frac{Kp}{1 + 2 \cdot D \cdot s - \omega^2}$$

Diese Formel stimmt nun mit Zeile 6 aus dem Code überein und liefert das nachfolgende Bodediagramm.

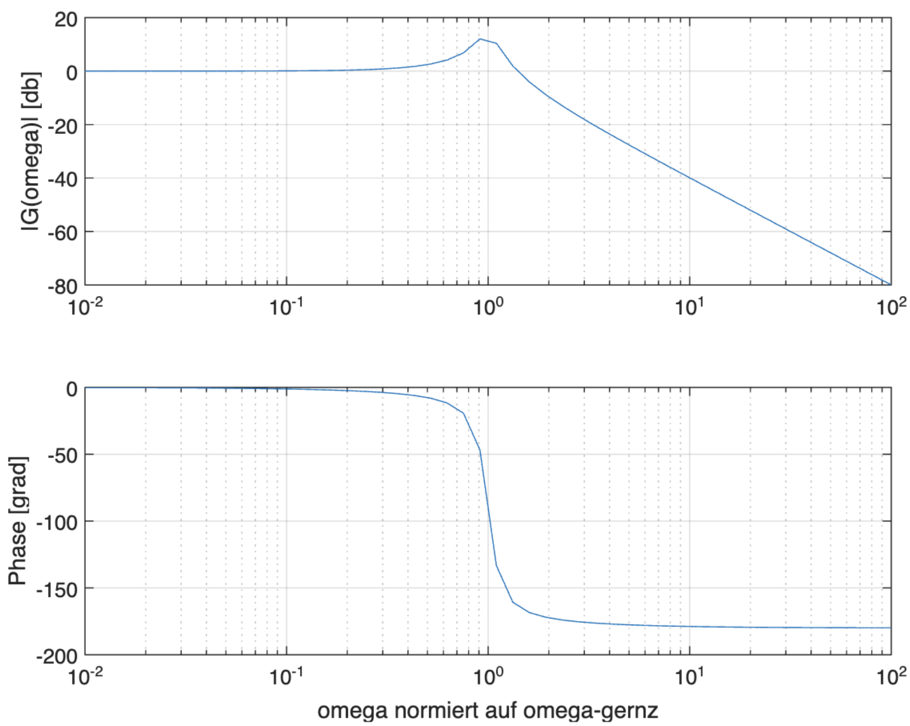


Abbildung 5: Bodediagramm entstanden mit den Code aus 2.3

3 Aufgabe 3: Ortskurve

Die Ortkurve ist eine graphische Darstellung des Frequenzgangs. Sie wird erstellt, indem man die den Frequenzbereich durchläuft, d.h. es werden einzelnen Werte (per Hand) oder alle Werte in die Übertragungsfunktion eingesetzt und etwaige Ergebnisse in den Koordinaten eingetragen.
Für die Ortskurve wird nicht die laplace transformierte Übertragfunktion mit s verwendet, sondern die normiert $j\omega$.

3.1 Ortskurven

3.1.1 a.1

$$G(s) = e^{-Ts}$$

$$G(j\omega) = e^{-Tj\omega}$$

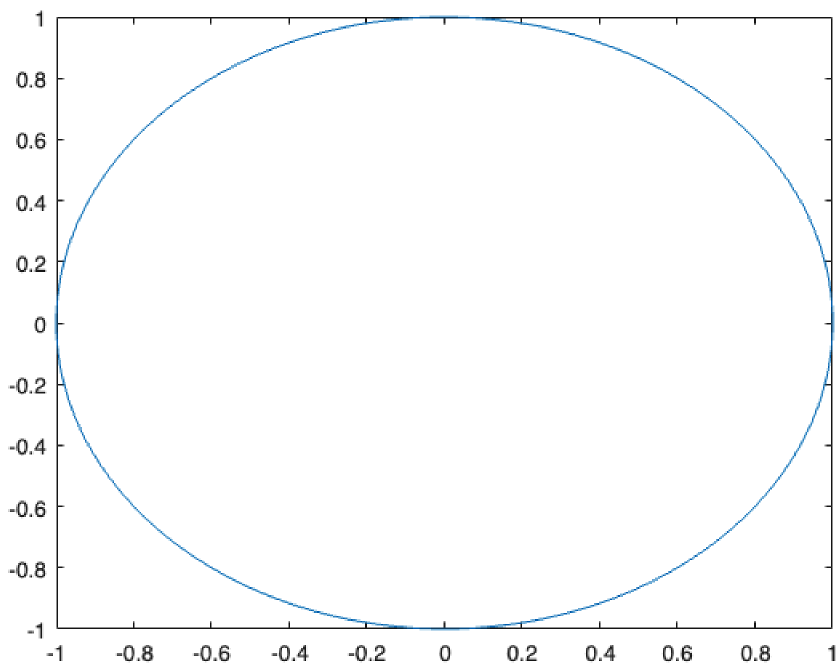


Abbildung 6: Ortskurve für unseres T-Glied

3.1.2 a.2

$$G(s) = \frac{K \cdot e^{-Ts}}{(1 + T_1 \cdot s)}$$

$$G(j\omega) = \frac{K \cdot e^{-Tj\omega}}{(1 + T_1 \cdot j\omega)}$$

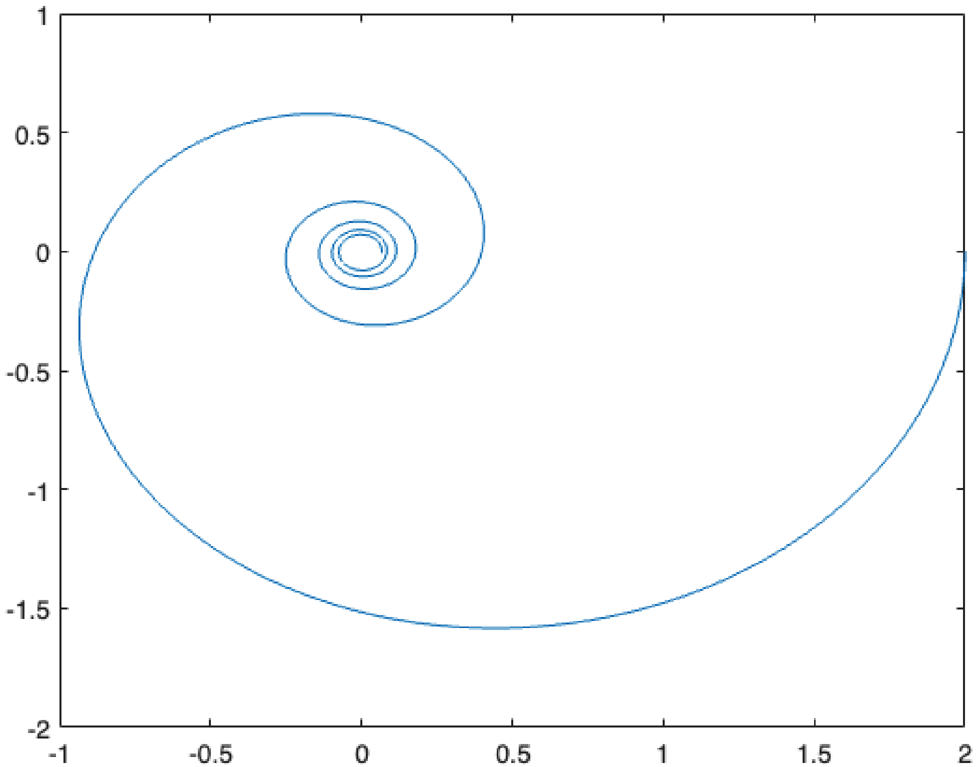


Abbildung 7: Ortskurve für unseres T-Glied

3.1.3 a.3

$$G(s) = \frac{K \cdot e^{-Ts}}{s(1 + T_1 \cdot s)}$$

$$G(j\omega) = \frac{K \cdot e^{-Tj\omega}}{j\omega(1 + T_1 \cdot j\omega)}$$

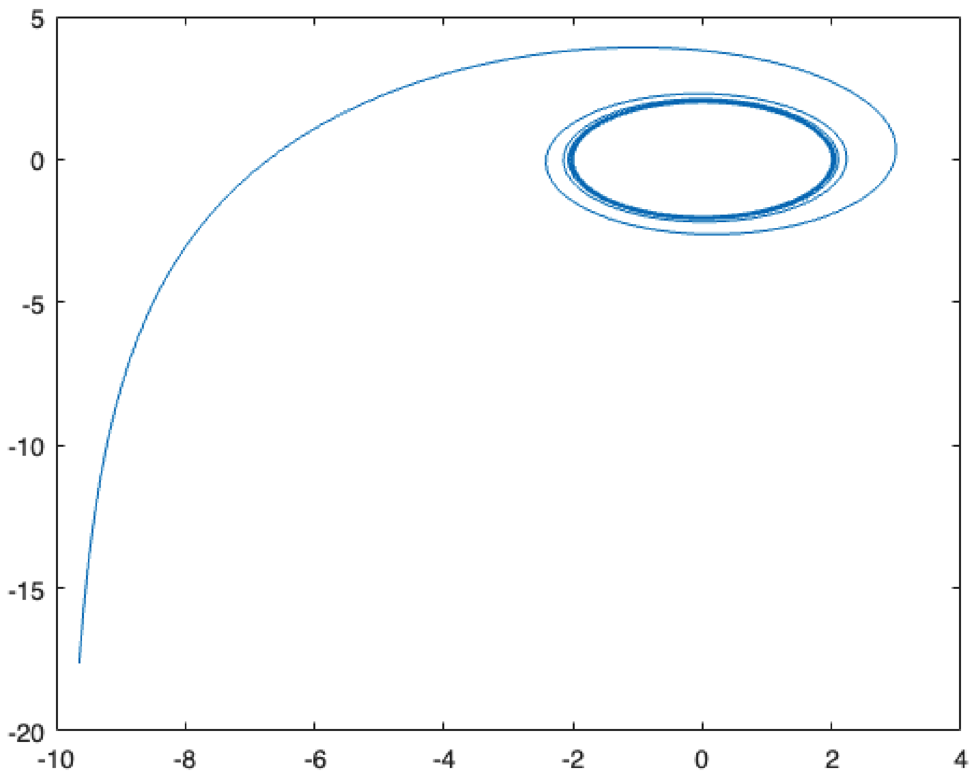


Abbildung 8: Ortskurve für unseres T-Glied

3.1.4 a.4

$$G(s) = \frac{K(T_v s + 1)}{T_2 s^2 + T_1 s + 1}$$

$$G(j\omega) = \frac{K(T_v j\omega + 1)}{T_2 (j\omega)^2 + T_1 j\omega + 1}$$

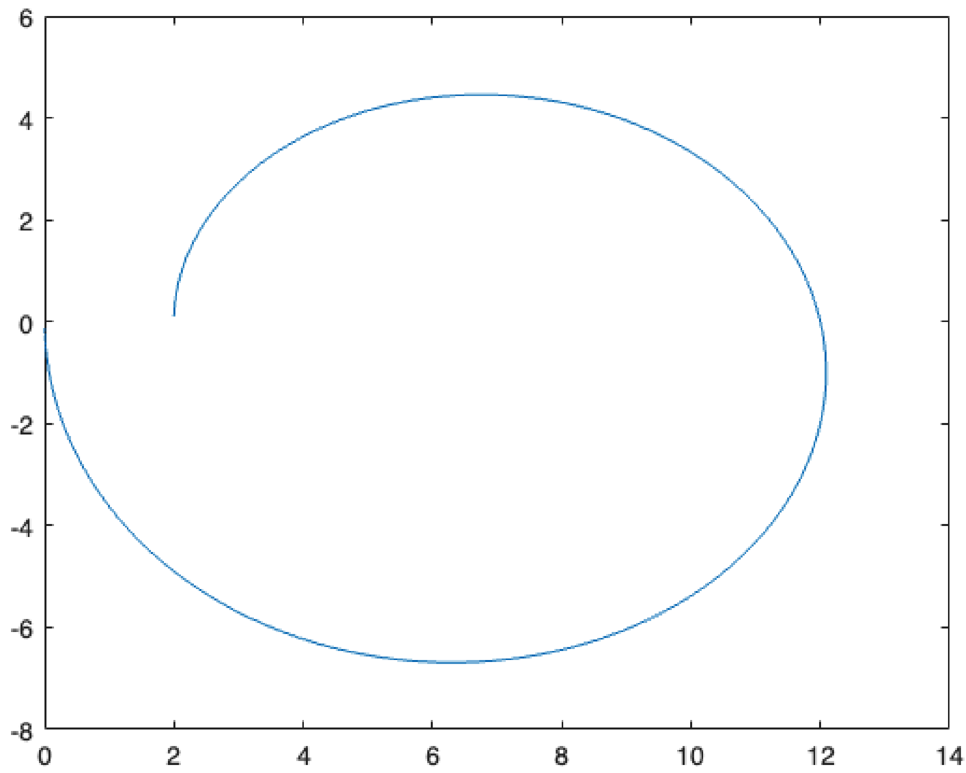


Abbildung 9: Ortskurve für unseres T-Glied

3.2 Grundverhalten der Regelglieder

3.2.1 b.1

Die erste Übertragungsfunktion besitzt das Grundverhalten eines T-Glied/Totzeitglied.

3.2.2 b.2

Die zweite Übertragungsfunktion besitzt das Grundverhalten eines PT_1-Glieds.

3.2.3 b.3

Die dritte Übertragungsfunktion besitzt das Grundverhalten eines IT_1-Glieds.

3.2.4 b.4

Die vierte Übertragungsfunktion besitzt das Grundverhalten eines PT_2-Glieds.

4 Aufgabe 4: MATLAB Control System Toolbox

4.1 Grund-Übertragungsverhalten

Die einzelnen Glieder verhalten sich wie folgt:

- $$G_R = K_R$$
  ist eine **Konstante**.
- $$G_1 = \frac{F_0}{T_0^2 s^2 + 2DT_0 s + 1}$$
  zeigt das Verhalten eines **PT-2 Glieds**.
- $$G_2 = \frac{1}{K_L(1 + T_s)}$$
  verhält sich wie ein **PT-1 Glied**.
- $$G_3 = \frac{1}{s}$$
  handelt nach einem **I-Glied**.
- $$G_{RADAR} = K_{RADAR}$$
  ist wieder **konstant**.

4.2 G_O Übertragungsfunktion des offenen Regelkreises

Für die Übertragungsfunktion G_O werden alle Teilglieder nacheinander multipliziert.

$$G_O = G_{RADAR} \cdot G_R \cdot G_1 \cdot G_2 \cdot G_3$$

4.3 Ortskurve G_O

Um mit der erstellt Formel für G_O die zugehörige Ortskurve zu erstellen, muss diese in MATLAB wie in folgendem Code angepasst werden.
Der Befehle `nyquist` in Zeile 18 plotet den Graphen.

```
1  // Konstanten
2  KRadar = 1;
3  F0 = 1000;
4  T0 = 1;
5  D = 0.5;
6  T1 = 1;
7  KL = 1000;
8  GR = 0.1;
9  // Uebertragungsfunktion
10 G1 = tf(F0 ,[T0^2 2*D*T0 1]);
11 G2 = tf(1, [KL KL*T1]);
12 G3 = tf(1, [1 0]);
13 GRadar = tf(KRadar, 1);
14 G0 = series(series(series(series(GRadar,GR),G1),G2),G3);
15 nyquist(G0);
```

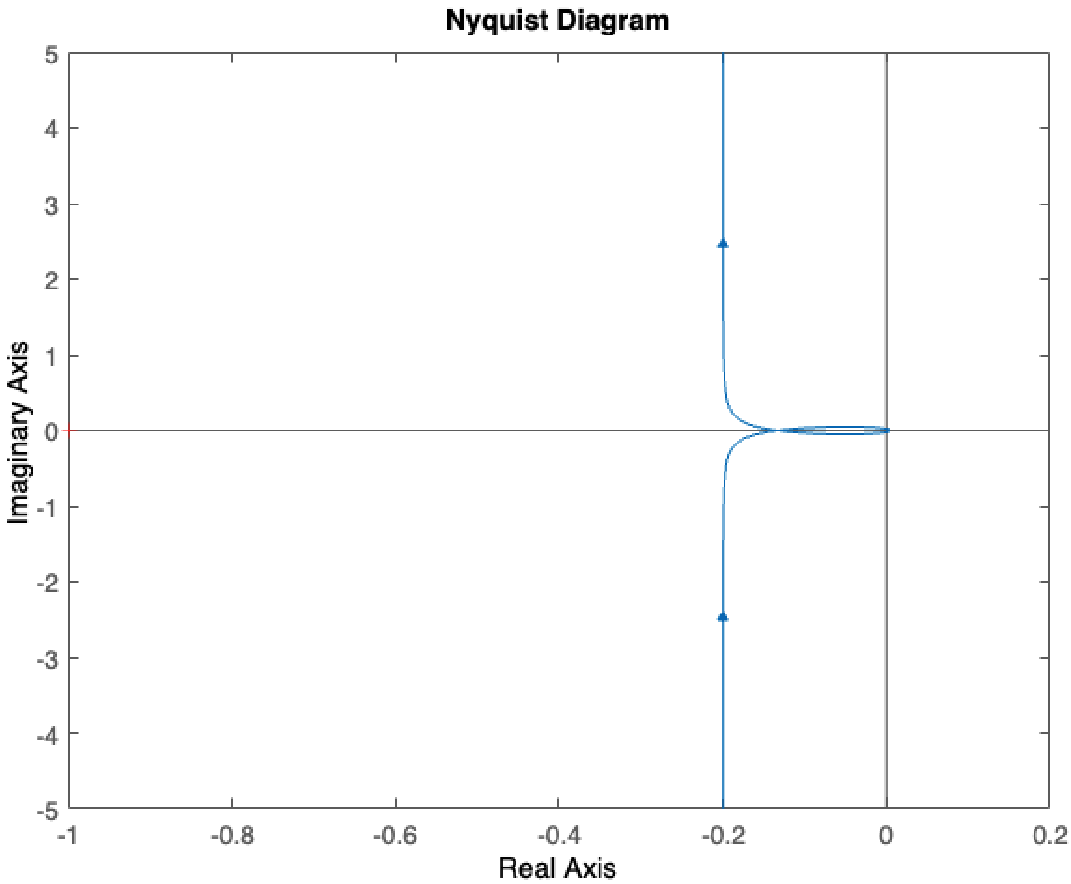


Abbildung 10: Ortskurve für die Übertragungsfunktion G_O des offenen Regelkreises.

4.4 Übertragungsfunktionen

Die Ortskurven für die gegebenen Werte K_{Ri} werden mit folgendem Code geplotet:

```
//Konstanten
KRadar = 1;
F0 = 1000;
T0 = 1;
D = 0.5;
T1 = 1;
KL = 1000;
GR = 0.1;

//Teilaufgabe d
KR1 = 0.05;
KR2 = 0.1;
KR3 = 0.2;
KR4 = 0.4;
KR5 = 0.8;

G01 = getG0(KR1, F0,T0,D,T1,KL,KRadar);
G02 = getG0(KR2, F0,T0,D,T1,KL,KRadar);
G03 = getG0(KR3, F0,T0,D,T1,KL,KRadar);
G04 = getG0(KR4, F0,T0,D,T1,KL,KRadar);
G05 = getG0(KR5, F0,T0,D,T1,KL,KRadar);

figure;
nyquist(G01, 'b', G02, 'g', G0, 'r', G04, 'c', G05, 'm');
axis([-1, 0.4, -3, 3])
grid;

function G0 = getG0(KR, F0, T0, D, T1, KL, KRadar)
    G1 = tf(F0 ,[T0^2 2*D*T0 1]);
   G2 = tf(1, [KL KL*T1]);
    G3 = tf(1, [1 0]);
    GRadar = tf(KRadar, 1);
    G0 = series(series(series(series(GRadar, KR), G1), G2),
        G3);
end
```

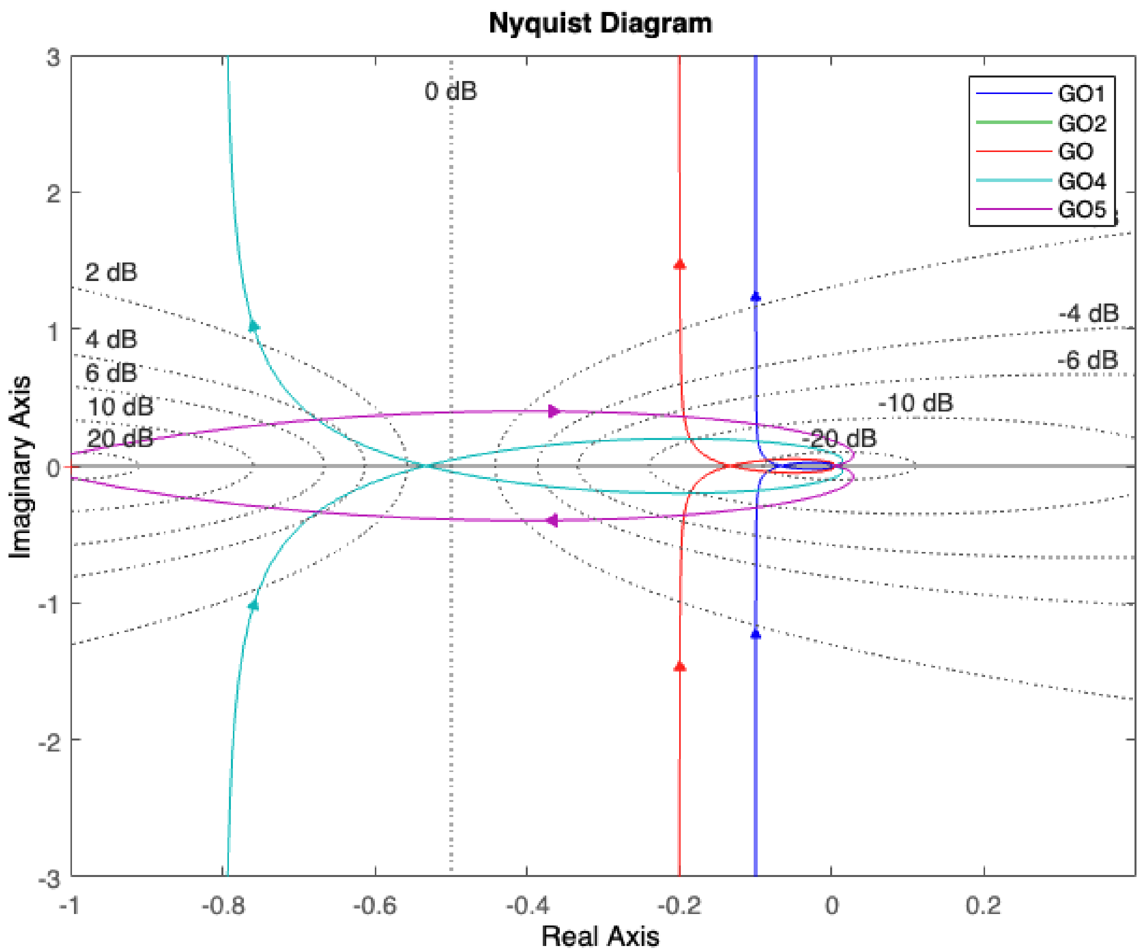


Abbildung 11: Ortskurven für die jeweiligen Werte für K_R.

4.5 Bodediagramme

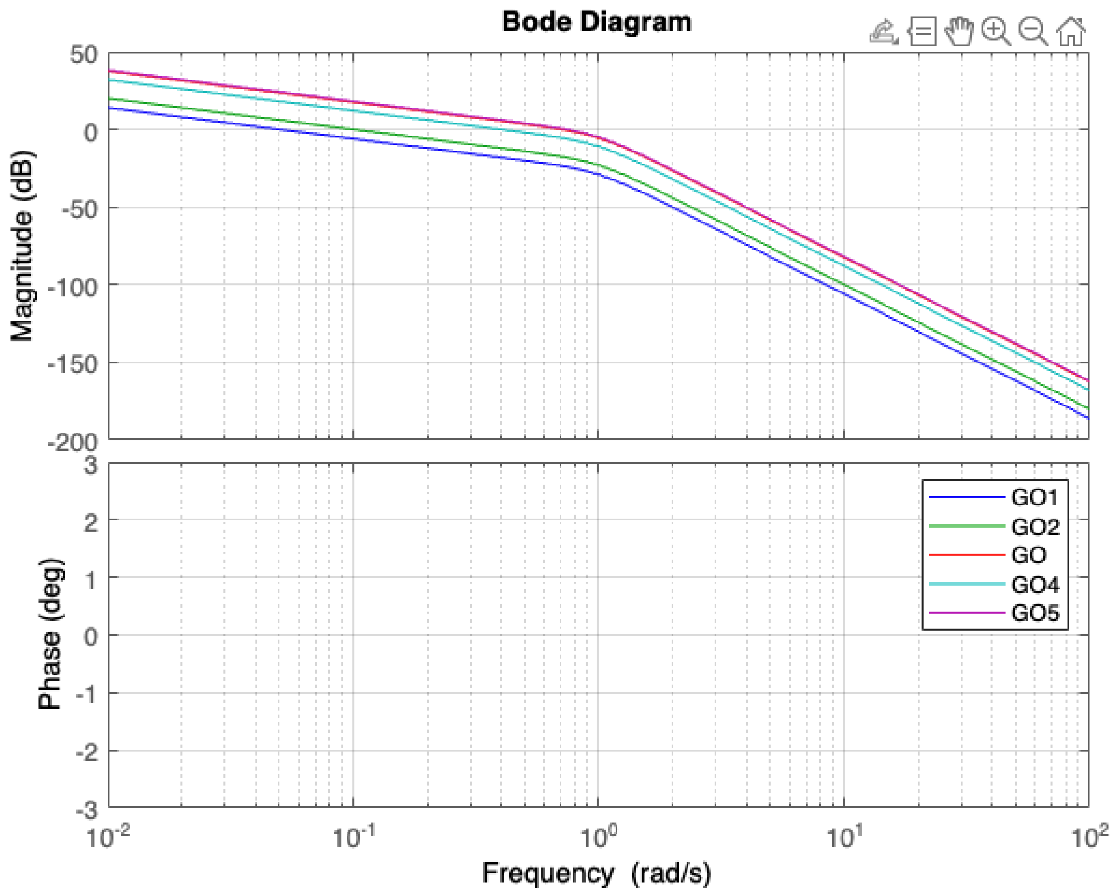


Abbildung 12: Ortskurven für die jeweiligen Übertragungsfunktionen G_{Oi}.

4.6 Bodediagramm des offenen Regelkreises mit `margin`

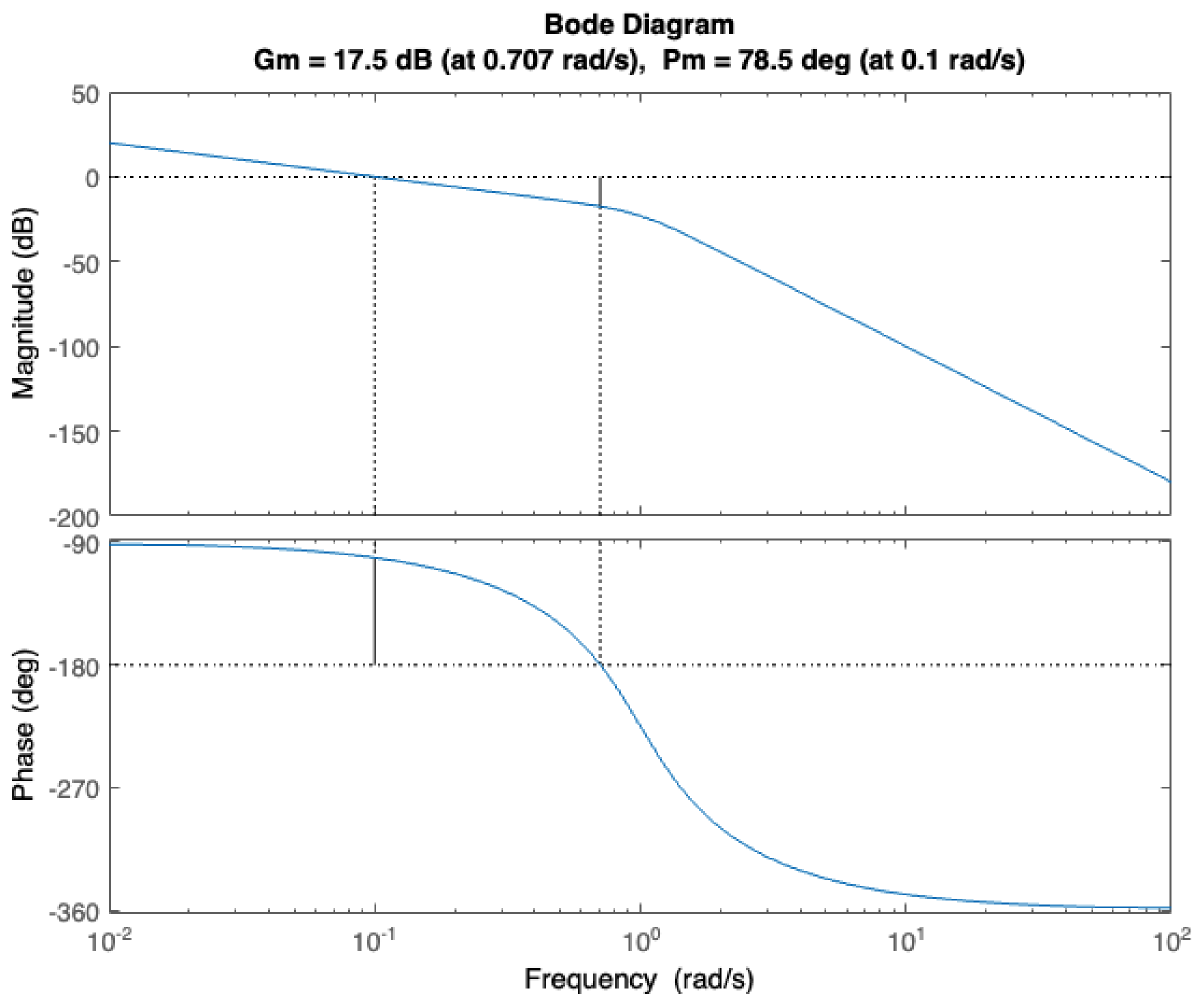


Abbildung 13: Bodediagramm für die Übertragungsfunktion des offenen Regelkreises $G_O(s)$.

Mit dem gegebenen Code lässt sich ausprobieren, welchen Wert K_R maximal annehmen darf.

```
'Ergebnis: groesstes KR mit Stabilitaet ist ...'
disp(KR_max);
KR_range= 0.1:0.01:2; //Vektor KR
for i = 1:length(KR_range),
    KR = KR_range(i);
    G0 = getG0(KR, F0,T0,D,T1,KL,KRadar);
    // GM = Amplitudenreserve
    // PM = Phasenreserve
    // WGM = Winkel der Frequenz, bei der 0 dB geschnitten wird.
    // WPM = Winkel der Frequenz, bei der 180 Grad geschnitten wird.
    [GM, PM, WGM, WPM] = margin(G0);
    if (GM<=1) || (PM<=0) || (WGM<=WPM)
        KR_max = KR_range(i-1);
        break;
    end;
end;
```

Die Schleife läuft über die Länge des Vektor KR, dessen Bereich von 0.1 bis 2 in 0.01 großen Schritten, reicht. Dadurch wird jeder Wert nacheinander innerhalb der Schleife abgefragt und mit der Übertragsfunktion berechnet. Der Befehl `margin` gibt damit die gesuchten Werte für Amplituden- und Phasenreserve wider; zusammen mit den dazugehörigen Winkelngrößen.
Mit der `if`-Abfrage bekommt man anschließend den gesuchten maximalen Wert zurück.

Der Komplette Code, der für diese Aufgabe verwendet wurde:

```
KRadar = 1;
F0 = 1000;
T0 = 1;
D = 0.5;
T1 = 1;
KL = 1000;
GR = 0.1;

G1 = tf(F0 ,[T0^2 2*D*T0 1]);
G2 = tf(1, [KL KL*T1]);
G3 = tf(1, [1 0]);
G0 = series(series(series(series(tf(KRadar,
    1),GR),G1),G2),G3);

// Bode-Diagramm plotten
figure;
bode(G0);
// Gain- und Phase Margin berechnen
margin(G0);
[GM, PM, WGM, WPM] = margin(G0);
disp(['Gain Margin (GM) in dB: ' num2str(GM)]);
disp(['Phase Margin (PM) in degrees: ' num2str(PM)]);
disp(['Gain Crossover Frequency (WGM) in rad/s: '
    num2str(WGM)]);
disp(['Phase Crossover Frequency (WPM) in rad/s: '
    num2str(WPM)]);

'Ergebnis: groesstes KR mit Stabilitaet ist ...'
disp(KR_max);
KR_range= 0.1:0.01:2; //Vektor KR
for i = 1:length(KR_range),
    KR = KR_range(i);
    G0 = getG0(KR, F0,T0,D,T1,KL,KRadar);
    [GM, PM, WGM, WPM] = margin(G0);
    if (GM<=1) || (PM<=0) || (WGM<=WPM)
        KR_max = KR_range(i-1);
        break;
    end;
end;
function G0 = getG0(KR, F0, T0, D, T1, KL, KRadar)
    G1 = tf(F0 ,[T0^2 2*D*T0 1]);
    G2 = tf(1, [KL KL*T1]);
    G3 = tf(1, [1 0]);
    GRadar = tf(KRadar, 1);
    G0 = series(series(series(series(GRadar, KR), G1), G2),
        G3);
end
```

4.7 Wurzelortskurve

```
KRadar = 1;
F0 = 1000;
T0 = 1;
D = 0.5;
T1 = 1;
KL = 1000;
GR = 0.1;
k = 0.707; // entspricht 45 Grad
G1 = tf(F0 ,[T0^2 2*D*T0 1]);
G2 = tf(1, [KL KL*T1]);
G3 = tf(1, [1 0]);
GRadar = tf(KRadar, 1);
G0 = series(series(series(GRadar,GR),G1),G2);
GCL1 = feedback(G0, G3);

rlocus(G0);
[K_opt, poles] = rlocfind(G0, k);
disp(['Der optimale Verstaerkungsfaktor K betraegt: ' num2str(K_opt)])
```

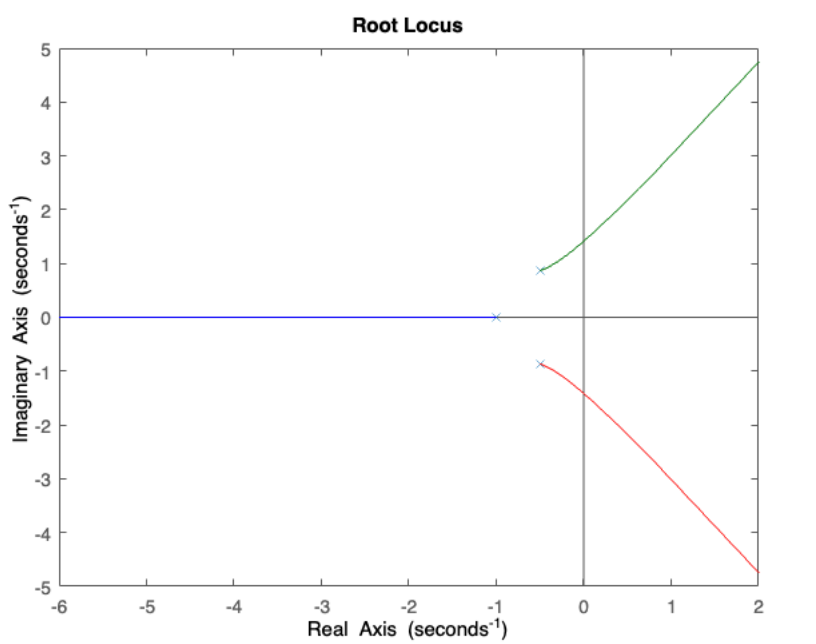


Abbildung 14: Wurzelortskurven

4.8 Übertragungsfunktionen der geschlossenen Regelkreise

```
// Konstanten
KRadar = 1;
F0 = 1000;
T0 = 1;
D = 0.5;
T1 = 1;
KL = 1000;
GR = 0.1;

// aus Teilaufgabe d
KR1 = 0.05;
KR2 = 0.1;
KR3 = 0.2;
KR4 = 0.4;
KR5 = 0.8;

GO1 = getGO(KR1, F0,T0,D,T1,KL,KRadar);
GO2 = getGO(KR2, F0,T0,D,T1,KL,KRadar);
GO3 = getGO(KR3, F0,T0,D,T1,KL,KRadar);
GO4 = getGO(KR4, F0,T0,D,T1,KL,KRadar);
GO5 = getGO(KR5, F0,T0,D,T1,KL,KRadar);

// geschlossener Regelkreis
GW1 = feedback(GO1, KRadar);
GW2 = feedback(GO2, KRadar);
GW3 = feedback(GO3, KRadar);
GW4 = feedback(GO4, KRadar);
GW5 = feedback(GO5, KRadar);

function GO = getGO(KR, F0, T0, D, T1, KL, KRadar)
    G1 = tf(F0 ,[T0^2 2*D*T0 1]);
    G2 = tf(1, [KL KL*T1]);
    G3 = tf(1, [1 0]);
    GRadar = tf(KRadar, 1);
    GO = series(series(series(series(GRadar, KR), G1),
        G2), G3);
end
```

Nach diesem Code, mit Hilfe des Befehls `feedback`, sehen die Übertragsfunktionen wie folgt aus:

4.8.1 $G_{O1} \rightarrow G_{W1}$

$$G_{W1} = \frac{50}{1000s^4 + 2000s^3 + 2000s^2 + 1000s + 50}$$

4.8.2 $G_{O2} \to G_{W2}$

$$G_{W2} = \frac{100}{1000s^4 + 2000s^3 + 2000s^2 + 1000s + 100}$$

4.8.3 $G_{O3} \to G_{W3}$

$$G_{W3} = \frac{200}{1000s^4 + 2000s^3 + 2000s^2 + 1000s + 200}$$

4.8.4 $G_{O4} \to G_{W4}$

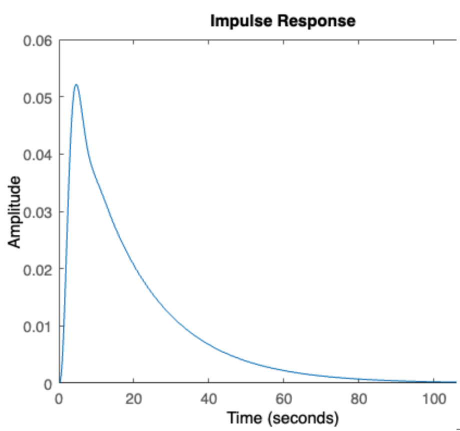
$$G_{W4} = \frac{400}{1000s^4 + 2000s^3 + 2000s^2 + 1000s + 400}$$

4.8.5 $G_{O5} \to G_{W5}$

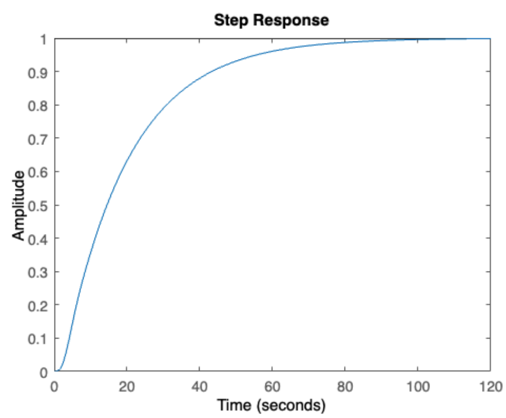
$$G_{W5} = \frac{800}{1000s^4 + 2000s^3 + 2000s^2 + 1000s + 800}$$

4.9 Impuls- und Sprungantworten

4.9.1 G_{W1}

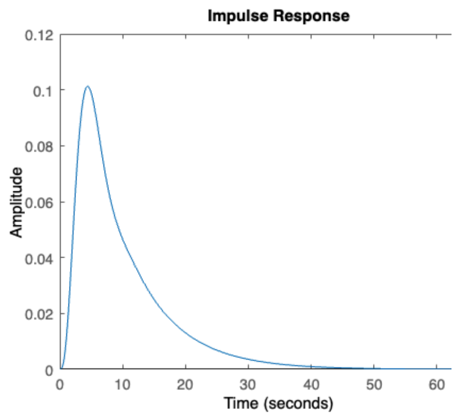


(a) Impulsantwort

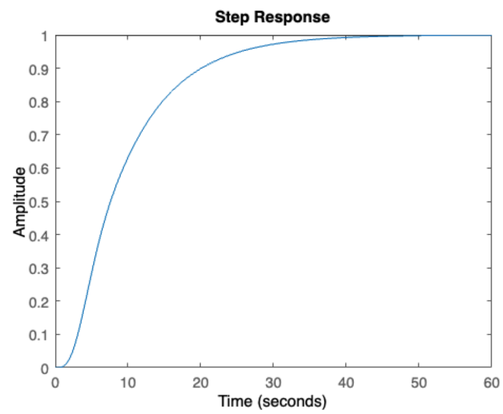


(b) Sprungantwort

4.9.2 G_{W2}

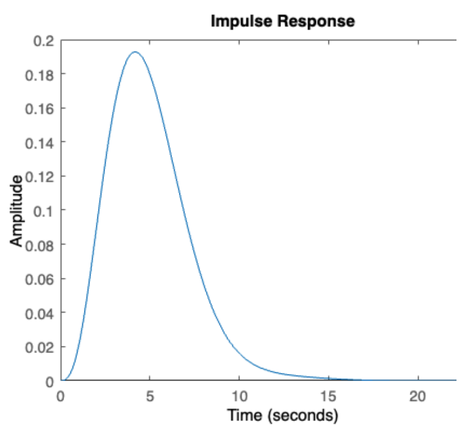


(a) Impulsantwort

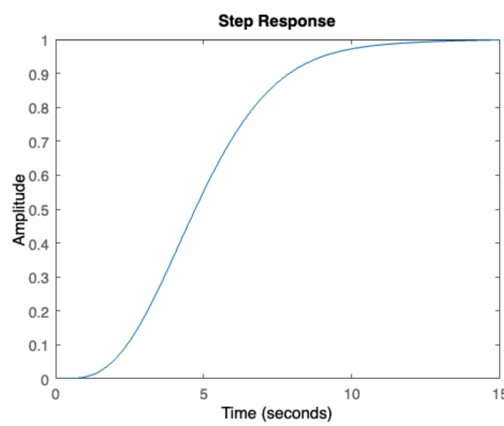


(b) Sprungantwort

4.9.3 G_{W3}

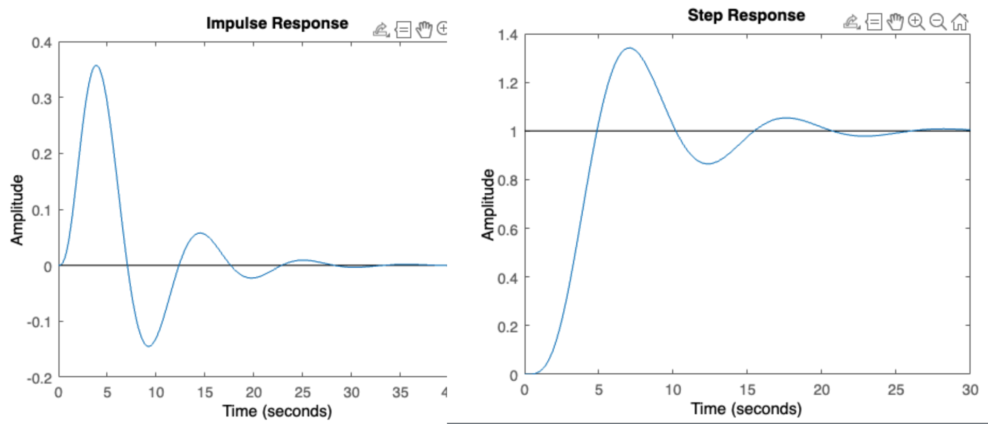


(a) Impulsantwort



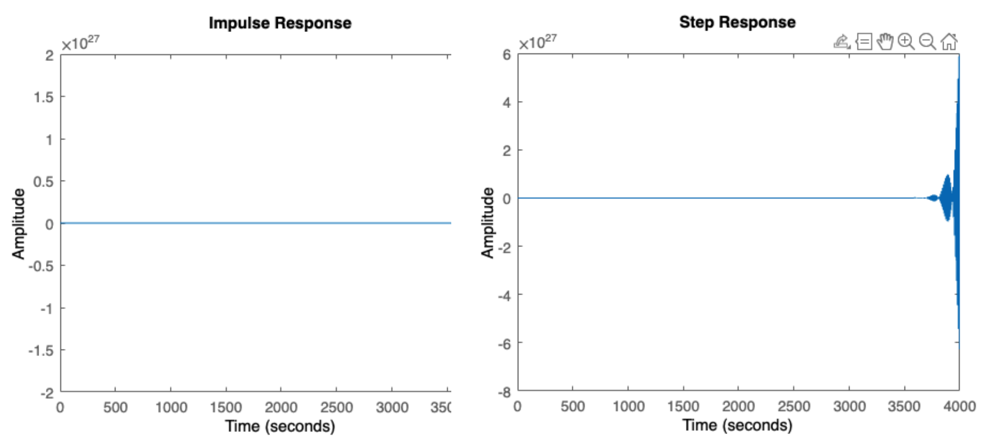
(b) Sprungantwort

4.9.4 G_{W4}



(a) Impulsantwort (b) Sprungantwort

4.9.5 G_{W5}



(a) Impulsantwort (b) Sprungantwort