2024

EDS-Laborversuch 1

PRAKTIKUM EREIGNISDISKRETE SYSTEME

FABIAN BRZESINA

Aufgabe 1:

Thema: MathLab Grundlagen

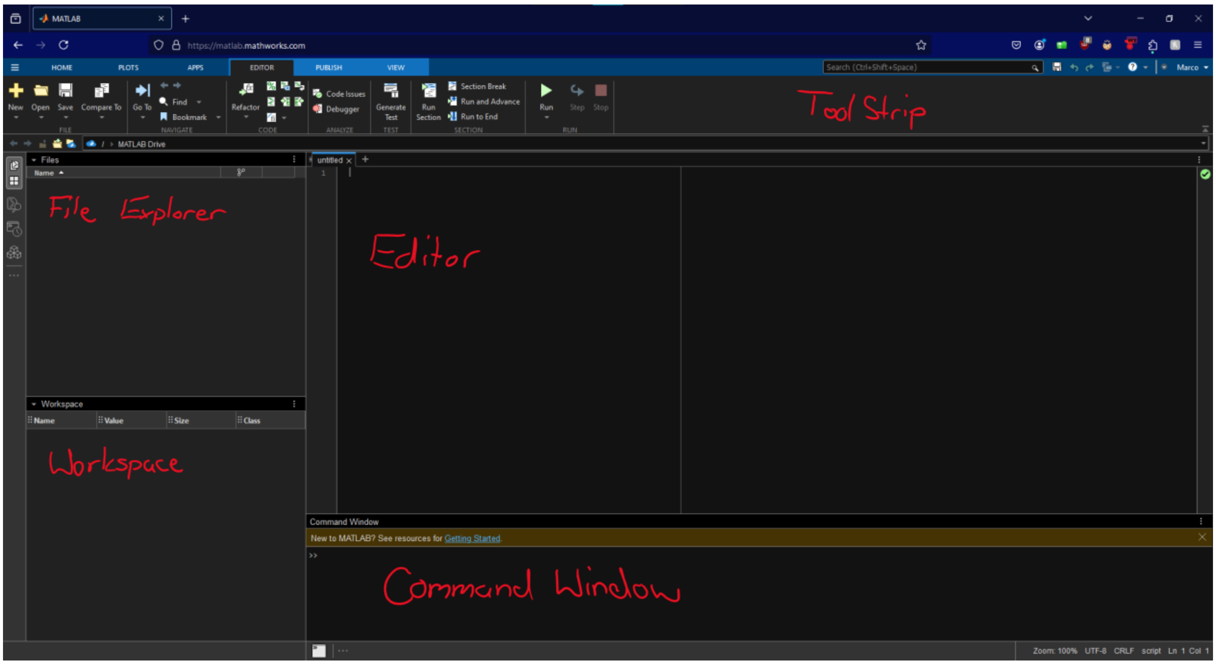
a) Was ist MatLab?

MATLAB ist eine Software und Programmiersprache, die hauptsächlich für numerische Berechnungen, Datenanalysen, Visualisierungen und die Entwicklung von Algorithmen verwendet wird. MATLAB wird häufig in Ingenieurwissenschaften, Physik, Mathematik und anderen wissenschaftlichen Bereichen eingesetzt, da es eine Vielzahl von Werkzeugen und Funktionen für die Bearbeitung von Matrizen, mathematischen Funktionen und Visualisierungen bietet.

b) Nennen Sie anhand eines Screenshots die wesentlichen Komponenten der Oberfläche von MATLAB.

Die wesentlichen Komponenten der MATLAB-Oberfläche umfassen:

Command Window – Bereich, in dem Befehle und Berechnungen eingegeben werden.
Workspace – Zeigt alle aktuellen Variablen und ihre Werte an.
Current Folder – Listet Dateien im aktuellen Arbeitsverzeichnis.
Editor – Zum Schreiben und Bearbeiten von Skripten.
Tool-Strip – Menüleiste mit verschiedenen Funktionen und Werkzeugen.



c) Wozu wird der Current Folder Browser benötigt und was ist bezüglich des Current Folder (aktuelles Arbeitsverzeichnis) zu beachten?

Der Current Folder Browser zeigt alle Dateien und Skripte im aktuellen Arbeitsverzeichnis an. Dieses Verzeichnis ist der Speicherort, von dem MATLAB standardmäßig Dateien liest und in dem es Ausgaben speichert. Es ist wichtig sicherzustellen, dass das aktuelle Arbeitsverzeichnis korrekt eingestellt ist, damit MATLAB Dateien und Daten findet, die für die Berechnung oder das Projekt benötigt werden.

d) Was verbirgt sich hinter dem Begriff Command Window?

Das Command Window ist der zentrale Bereich der MATLAB-Oberfläche, in dem der Benutzer Befehle eingeben und ausführen kann. Ergebnisse und Ausgaben von Berechnungen werden hier ebenfalls angezeigt. Es ist das primäre Interface zur Kommunikation mit MATLAB. Hier können eigens entwickelte Skripte direkt ausgeführt werden.

e) Was verbirgt sich hinter dem Tool-Strip?

Der Tool-Strip ist die Menüleiste am oberen Rand der MATLAB-Oberfläche. Er enthält eine Vielzahl von Symbolen und Menüs für häufig verwendete Funktionen, wie das Öffnen von Dateien, das Erstellen von Skripten, das Speichern und die Navigation zwischen verschiedenen Modulen und Werkzeugen. Außerdem kann auch hier auf Simulink zugegriffen werden.

f) Welchen Zweck hat der Workspace? Nennen Sie vergleichbare Eigenschaften aus Ihnen bekannten Gebieten.

Der Workspace zeigt alle im aktuellen MATLAB-Skript oder -Projekt definierten Variablen, ihre Werte und Datentypen. Er ermöglicht das Verwalten und Überprüfen von Daten, die in Berechnungen verwendet werden. Vergleichbar ist der Workspace mit Variablen-Inspektoren in anderen Programmiersprachen oder Entwicklungsumgebungen, wie z. B. dem Variablen-Viewer in Python-IDE's wie Spyder.

g) Nennen Sie zwei Möglichkeiten um Informationen aus der MATLAB-Hilfe abzurufen.

- Über das Hilfe-Menü im Tool-Strip, das eine allgemeine Übersicht über die MATLAB-Dokumentation bietet.
- Durch Eingabe von help <Befehl> im Command Window, was eine kurze Beschreibung zu einem spezifischen Befehl anzeigt.

h) Dokumentieren Sie die notwendigen Schritte um Simulink zu starten.

1. Öffnen Sie MATLAB.
2. Klicken Sie im Tool-Strip auf Simulink oder geben Sie simulink im Command Window ein.
3. Simulink wird gestartet und öffnet die Simulink Library Browser, in dem verschiedene Modelle und Bibliotheken zur Auswahl stehen.

i) Was ist die Control System Toolbox und wo findet man diese?

Die Control System Toolbox ist eine MATLAB-Toolbox, die Werkzeuge zur Analyse und zum Entwurf von Regelungssystemen bereitstellt. Sie enthält Funktionen für die Modellierung, Simulation und Analyse von dynamischen Systemen. Diese Toolbox kann im Add-Ons-Menü des Tool-Strips gefunden und installiert werden oder durch Eingabe von controlSystemDesigner im Command Window aufgerufen werden, sofern sie installiert ist.

j) Dokumentieren Sie wie Stateflow gestartet wird.

1. Öffnen Sie MATLAB und starten Sie Simulink, wie oben beschrieben.
2. Erstellen Sie ein neues Simulink-Modell oder öffnen Sie ein vorhandenes Modell.
3. Im Simulink Library Browser finden Sie Stateflow. Klicken Sie darauf, um Stateflow-Bibliotheken zu durchsuchen.
4. Ziehen Sie Stateflow-Blöcke in das Modell, um Zustandsautomaten zu modellieren.

Aufgabe 2:

Thema: Bodediagramm

a) Wie lautet der normierte Frequenzgang G(j!) eines PT1-Glieds?

Durch Substituierung kann von Ausgangsformel auf die normierte Formel geschlossen werden:

$$s = j\omega \qquad \omega = \omega_N \cdot \omega_G \quad | \; \omega_G = \frac{1}{T}$$
$$= \frac{\omega_N}{T}$$

$$G(s) = \frac{k_P}{1 + Ts} \quad \Leftarrow s \text{ substituieren mit } j\omega$$
$$= \frac{k_P}{1 + Tj\omega} \quad \Leftarrow \omega \text{ substituieren mit } \frac{\omega_N}{T}$$
$$= \frac{k_P}{1 + \cancel{T} j \frac{\omega_N}{\cancel{T}}} \quad \Leftarrow T \text{ wird gekürzt}$$

$$\underline{\underline{G(s) = \frac{k_P}{1 + j\omega_N}}}$$

b) Wie lautet der normierte Frequenzgang G(j!) eines PT2-Glieds?

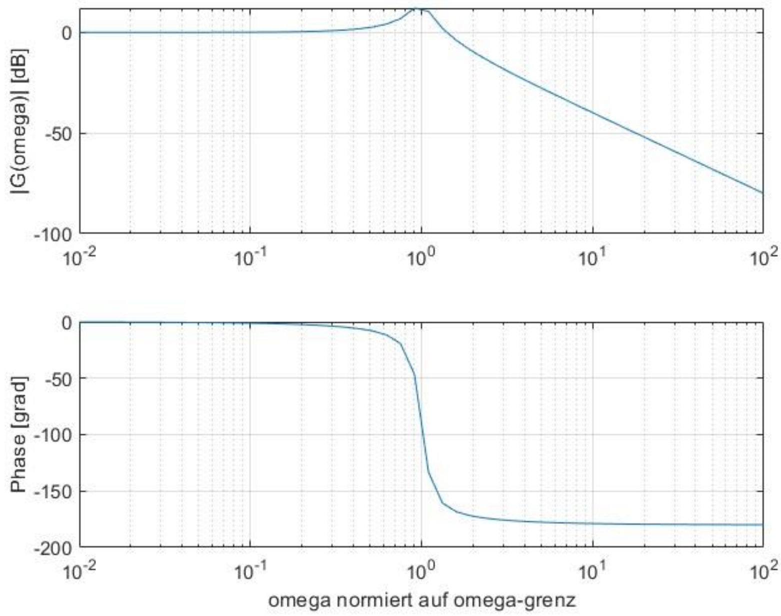
Durch Substituierung kann die Ausgangsformel auf die normierte Formel geschlossen werden:

$$G(s) = \frac{k_P}{T^2 s^2 + T_1 s + 1} \quad \Leftarrow s \text{ substituieren mit } j\omega$$
$$= \frac{k_P}{T^2 (j\omega)^2 + T j\omega + 1} \quad \Leftarrow \omega \text{ substituieren mit } \frac{\omega_N}{T}$$
$$= \frac{k_P}{\cancel{T^2} \left(\frac{j\omega_N}{\cancel{T}}\right)^2 + \cancel{T} j \frac{\omega_N}{\cancel{T}} + 1} \quad \Leftarrow T \text{ kürzen } // \; j^2 =: -1$$

$$\underline{G(s) = \frac{k_P}{-\omega_N^2 + j\omega_N + 1}}$$

c) **Erstellen Sie das MATLAB-Programm bodePT2.m für (b). Es soll das Bode-Diagramm für ein PT2-Glied mit D=0.1 (und KP=1) generieren.**

```
%establishes the variable omega as the logspace from 10^-2 to 10^2
omega=logspace(-2,2);
%divides an array of 50 ones by the term in brackets
G=ones(size(omega))./(1+j*2*0.1*omega-omega.^2);
subplot(2,1,1),semilogx(omega,20*log10(abs(G)))
grid,ylabel('|G(omega)| [dB]')
subplot(2,1,2),semilogx(omega,180*angle(G)/pi)
grid,xlabel('omega normiert auf omega-grenz')
ylabel('Phase [grad]')
```



Aufgabe 3

Thema: Ortskurven

Gegeben seien folgende Übertragungsfunktionen $G(s)$:

1) $G(s) = e^{-Ts}$ mit $T = 0.03$

2) $G(s) = \dfrac{Ke^{-Ts}}{(1+T_1 s)}$ mit $T = 0.03; T_1 = 0.03; K = 2$

3) $G(s) = \dfrac{Ke^{-Ts}}{s(1+T_1 s)}$ mit $T = 6; T_1 = 1; K = 2$

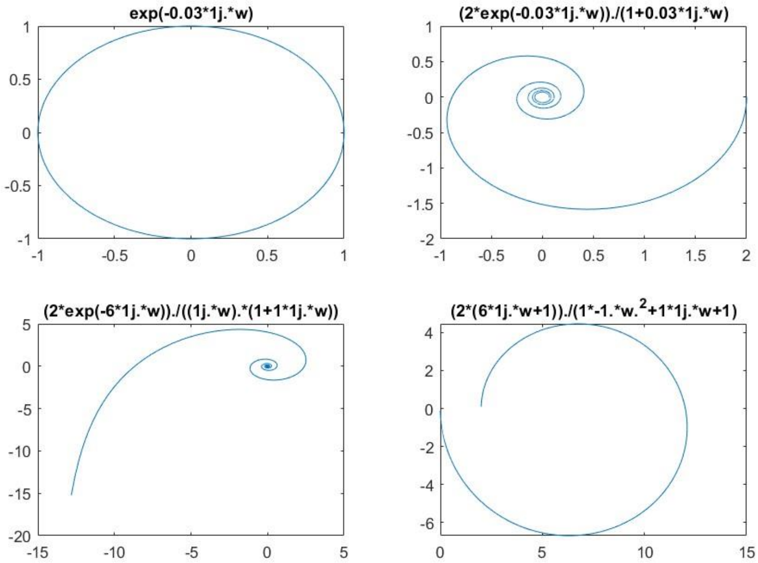
4) $G(s) = \dfrac{K(T_v s + 1)}{T_2 s^2 + T_1 s + 1}$ mit $T_v = 6; T_1 = T_2 = 1; K = 2$

a. **Erstellen Sie mit MATLAB die zugehörigen Ortskurven G(jw) des Frequenzganges der gegebenen Übertragungsfunktionen G(s) von 1) bis 4) für folgende w:**

1) und 2): $\omega \in [0; 1000]$, Schrittweite 0.1

3): $\omega = 0.1$ und $\omega = 100$, Schrittweite 0.01

4): $\omega = 0.01$ und $\omega = 100$, Schrittweite 0.01

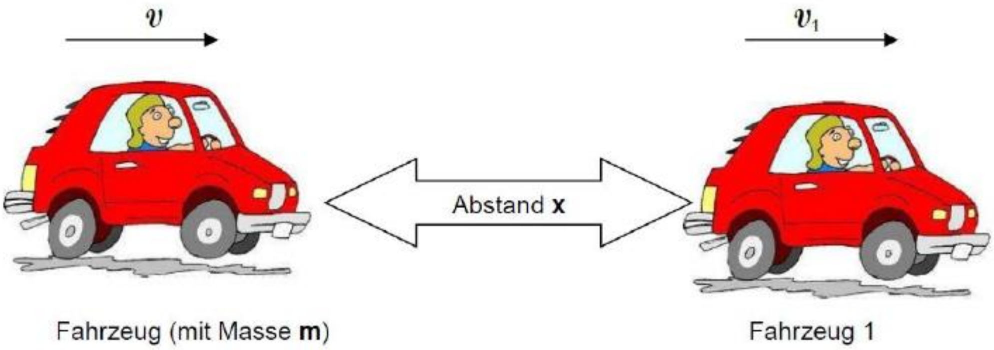


b. **Welches Grundverhalten haben die angegebenen Regelkreisglieder nach 1) bis 4)?**
Alle Regelkreise schwingen um einen bestimmten Punkt und bilden eine Spirale.
1) Totzeit-Glied
2) P-T1-Glied verkettet mit Totzeit-Glied und Verstärkung K
3) Wie 2) erweitert um ein I-Glied (s im Nenner)
4) Nenner: P-T2-Glied, Zähler: P-T1-Glied mit Verstärkung K

Aufgabe 4

Thema: MatLab Control System Toolbox

Die Geschwindigkeit eines Fahrzeuges soll so geregelt werden, dass es einem vorausfahrenden Fahrzeug 1 mit dem Abstand **x** folgt. **Skizze:**



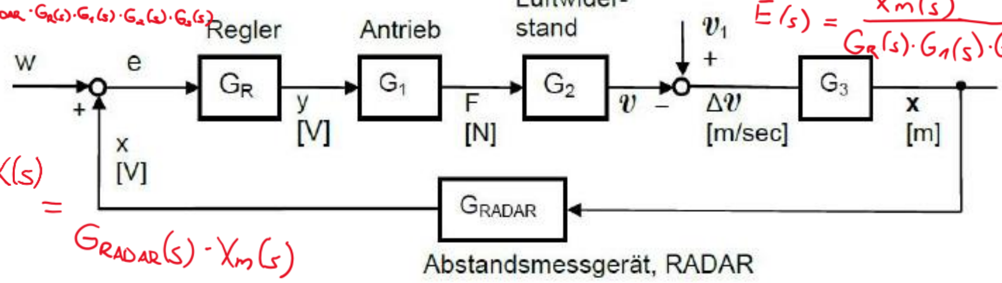
Regelkreis-Wirkungsplan:

$$G_0(s) = \frac{X(s)}{E(s)} = \frac{G_{RADAR} \cdot \cancel{X_m(s)}}{\frac{\cancel{X_m(s)}}{G_R(s) \cdot G_1(s) \cdot G_2(s) \cdot G_3(s)}}$$

$$G_0(s) = G_{RADAR} \cdot G_R(s) \cdot G_1(s) \cdot G_2(s) \cdot G_3(s)$$

$$X_m(s) = G_R(s) \cdot G_1(s) \cdot G_2(s) \cdot G_3(s) \cdot E(s)$$

$$E(s) = \frac{X_m(s)}{G_R(s) \cdot G_1(s) \cdot G_2(s) \cdot G_3(s)}$$



$$X(s) = G_{RADAR}(s) \cdot X_m(s)$$

Die Übertragungsfunktionen und Parameter der Regelkreisglieder des Regelkreis-Wirkungsplans seien dabei

$$G_R = K_R, \qquad G_1 = \frac{F_0}{T_0^2 s^2 + 2DT_0 s + 1}, \qquad G_2 = \frac{1}{K_L(1 + T_1 s)},$$

$$G_3 = \frac{1}{s}, \qquad G_{\text{RADAR}} = K_{\text{RADAR}}$$

mit Konstanten

$$F_0 = 1000\frac{\text{N}}{\text{V}}, \qquad T_0 = 1 \text{ sec}, \qquad D = 0.5, \qquad T_1 = 1 \text{ sec},$$

$$K_L = 1000\frac{\text{Nsec}}{\text{m}}, \qquad K_{\text{RADAR}} = 1\frac{\text{V}}{\text{m}}.$$

Nutzen Sie zur Analyse des Regelkreises die Befehle der Matlab Control System Toolbox:

a. Welches Grund-Typ-Übertragungsverhalten haben die einzelnen Regelkreisglieder?

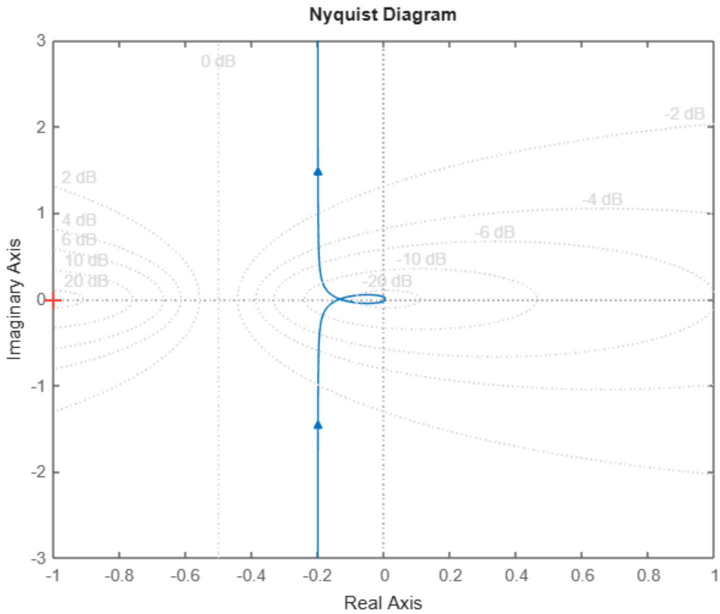
GR ist eine Konstante (P-Glied),
G1 ist ein PT2-Glied,
G2 ist ein PT1-Glied mit Verstärkung KL,
G3 ist ein i-Glied und
GRADAR ist eine Konstante (P-Glied)

b. Bestimmen Sie die Übertragungsfunktion des offenen Regelkreises GO(s) = ZO(s) / NO(s) für KR = 0.1 (die Dimensionen für GO heben sich auf). Nutzen Sie den MATLAB Control System Befehl [series] oder alternativ [conv] zur Erstellung der Übertragungsfunktion GO(s).

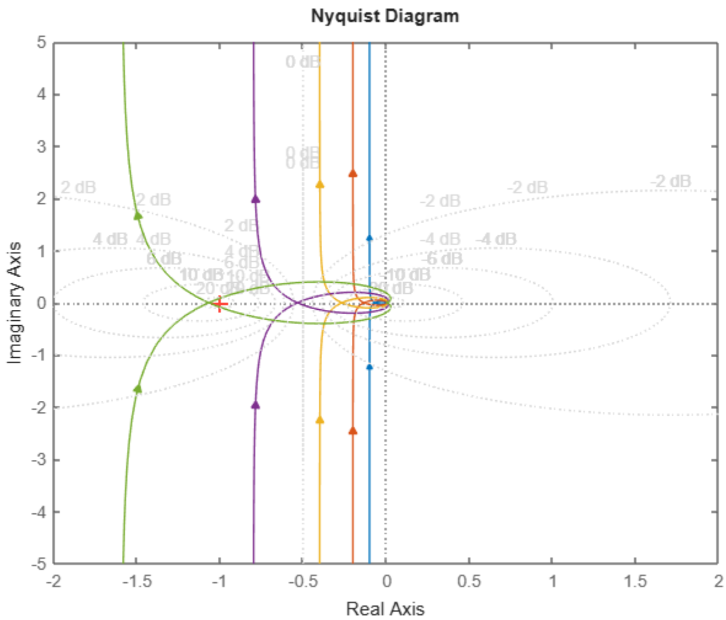
G0(s) = X(s) / E(s)

```
GR = tf([0.1],[1]);
G1 = tf([1000],[1 1 1]);
G2 = tf([1],[1000 1000]);
G3 = tf([1], [1 0]);
GRADAR = tf([1], [1]);
sys=series(series(series(series(GRADAR,GR),G1),G2),G3);
```

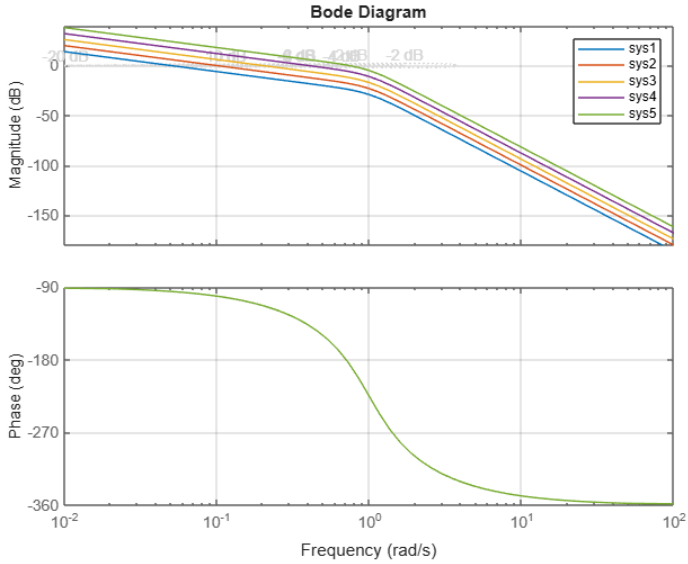
c. Erstellen Sie die Ortskurve GO(jw) für (b) mit dem Control System Befehl [nyquist]



d. Definieren Sie mehrere Übertragungsfunktionen des offenen Regelkreises GO(s) für KR1 = 0.05, KR2 = 0.1, KR3 = 0.2, KR4 = 0.4, KR5 = 0.8 in GO1, GO2, GO3, GO4, GO5 und erstellen Sie die Ortskurven mit dem Befehl [nyquist] dazu in einem Plot.

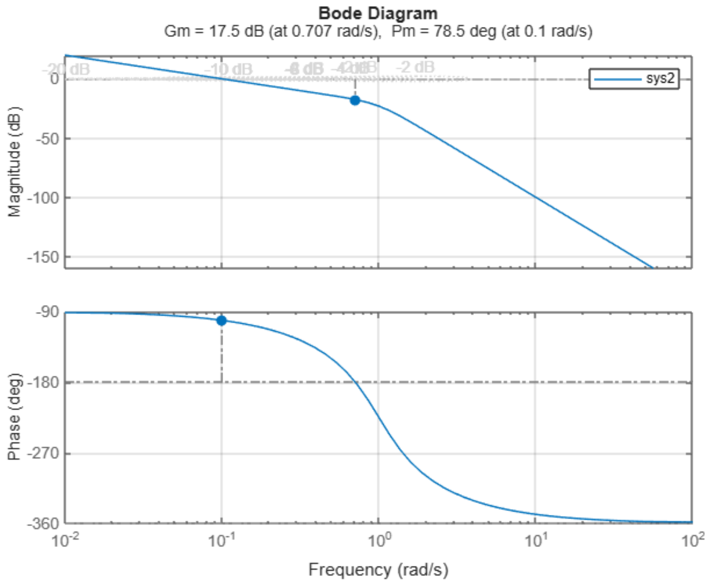


e. Erstellen Sie die Bode-Diagramme für GO1, GO2, GO3, GO4, GO5 in einem Plot mit dem Befehl [bode].



sys1 – sys5 stehen hier für GO1 – GO5

f. Generieren Sie das Bode-Diagramm für die Übertragungsfunktion des offenen Regelkreises GO(s) mit KR = 0.1 (bzw. für GO2) und ermitteln Sie die Phasen- und Amplituden-Reserve über den Befehl [margin].

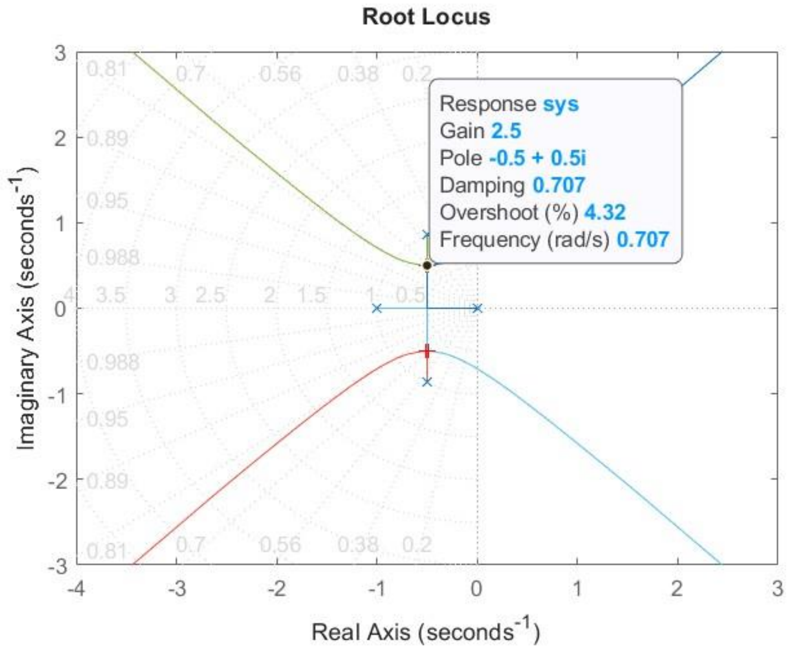


Auf welchen Wert darf KR maximal eingestellt werden?

```
G1 = tf([1000],[1 1 1]);
G2 = tf([1],[1000 1000]);
G3 = tf([1], [1 0]);
GRADAR = tf([1], [1]);

GR = 0.001:0.001:1;
for i =1:1:1000
    GT = GR(i);
    G0 = series(series(series(GT ,G1),G2),G3);
    [Gm ,Pm ,Wcg ,Wcp] = margin(G0);
    if Gm <=1 || Pm <= 0 || Wcg <= Wcp
        breakpoint = GR(i-1);
        break
    end
end

disp(breakpoint)
```

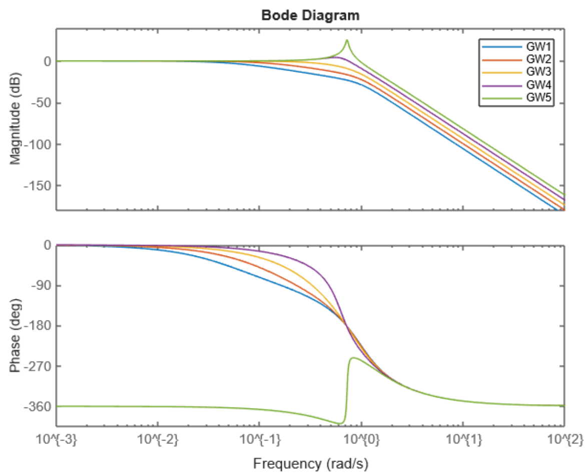
Der Breakpoint der nicht überschritten werden darf liegt nach ausführen des Skripts bei 0.7490

g. **Ermitteln Sie zu GO1 die Wurzelortskurven für den über eine Verstärkung K negativ rückgekoppelten Regelkreis mit dem Control System Befehl [rlocus] und bestimmen Sie den optimalen K-Wert bei der Dämpfung E = 0.707 (45°) über den Control System Befehl [rlocfind].**




```
GR = tf(0.1,1);
G1 = tf(1000,[1 1 1]);
G2 = tf(1,[1000 1000]);
G3 = tf(1, [1 0]);
GRADAR = tf(1, 1);

sys=series(series(series(series(GRADAR,GR),G1),G2),G3);
rlocus ( sys ) ;
grid on ;

[K , pole ]= rlocfind ( sys ) ;
```

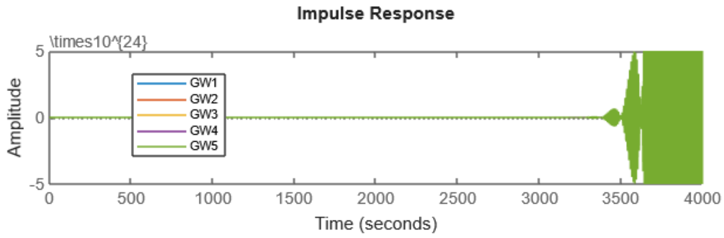
Erstellen Sie die Übertragungsfunktion des geschlossenen Regelkreises GW1 bis GW5 aus GO1 bis GO5 mit dem Befehl [feedback]. Beachten Sie hierbei GRADAR = 1.

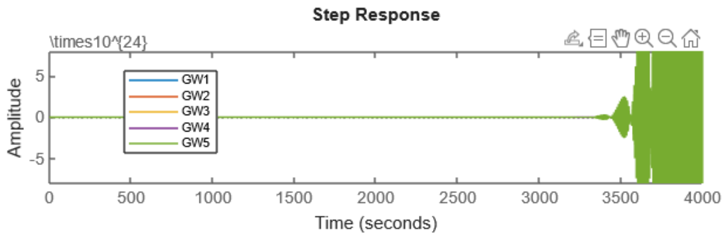



```
GR1 = tf([0.05],[1]);
GR2 = tf([0.1],[1]);
```

```
GR3 = tf([0.2],[1]);
GR4 = tf([0.4],[1]);
            GR5 = tf([0.8],[1]);
G1 = tf([1000],[1 1 1]);
G2 = tf([1],[1000 1000]);
G3 = tf([1], [1 0]);
GRADAR = tf([1], [1]);

        sys1=series(series(series(series(GRADAR,GR1),G1),G2),G3);
        sys2=series(series(series(series(GRADAR,GR2),G1),G2),G3);
        sys3=series(series(series(series(GRADAR,GR3),G1),G2),G3);
        sys4=series(series(series(series(GRADAR,GR4),G1),G2),G3);
        sys5=series(series(series(series(GRADAR,GR5),G1),G2),G3);

        GW1=feedback(sys1,GRADAR);
        GW2=feedback(sys2,GRADAR);
        GW3=feedback(sys3,GRADAR);
        GW4=feedback(sys4,GRADAR);
        GW5=feedback(sys5,GRADAR);

        bode (GW1,GW2,GW3,GW4,GW5);
        legend();
```

h. Ermitteln Sie die Impuls- und Sprungantwort für die geschlossenen Regelkreise GW1 bis GW5 in einem Plot mit den Befehlen [impulse] und [step].






```
GR1 = tf([0.05],[1]);
GR2 = tf([0.1],[1]);
GR3 = tf([0.2],[1]);
GR4 = tf([0.4],[1]);
GR5 = tf([0.8],[1]);
G1 = tf([1000],[1 1 1]);
G2 = tf([1],[1000 1000]);
G3 = tf([1], [1 0]);
GRADAR = tf([1], [1]);

sys1=series(series(series(series(GRADAR,GR1),G1),G2),G3);
sys2=series(series(series(series(GRADAR,GR2),G1),G2),G3);
sys3=series(series(series(series(GRADAR,GR3),G1),G2),G3);
sys4=series(series(series(series(GRADAR,GR4),G1),G2),G3);
sys5=series(series(series(series(GRADAR,GR5),G1),G2),G3);
GW1=feedback(sys1,1);
GW2=feedback(sys2,1);
GW3=feedback(sys3,1);
GW4=feedback(sys4,1);
GW5=feedback(sys5,1);

subplot(2,1,1)
impulse(GW1,GW2,GW3,GW4,GW5)
legend()
subplot(2,1,2)
step(GW1,GW2,GW3,GW4,GW5)
legend();
```