Regelungstechnik

Zusammenfassung

Joel von Rotz

Invalid Date

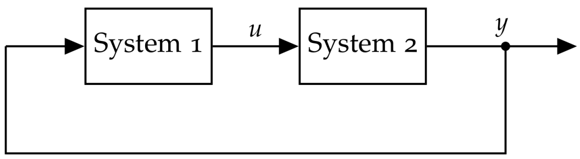
Quelldateien

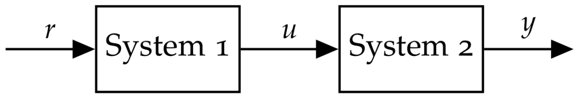
Inhaltsverzeichnis

1. Einführung

1.1 Rückkopplung

Rückkopplung beschreibt eine Anordnung, bei welcher zwei oder mehr dynamische Systeme Systeme untereinander so verbunden sind, dass sie sich gegenseitig beeinflussen.





Wird das Rückkopplungssignals des geschlossenen Kreises vom Eingangssignal entfernt, also die Leitung wird aufgebrochen, wird aus dem Kreis ein *offener* Kreis.

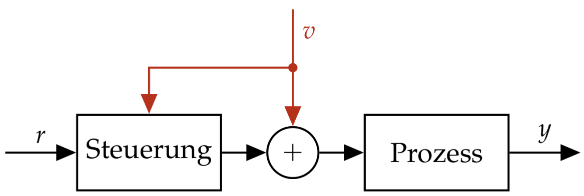
Vorsicht

Geschlossene Kreise sind schwieriger zum Berechnen und zum Untersuchen, da diese ein rückgekoppeltes Signal (mit dem Eingangssignal kombinierend) Teil des Eingangssignals zum System besitzen. **Offene** Kreise haben kein rückgekoppeltes Signal.

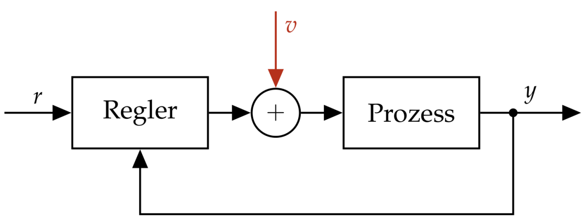
1.2 Mit- und Gegenkopplung

Beide beschriebenden Systeme arbeiten nach dem Prinzip der *Gegenkopplung*

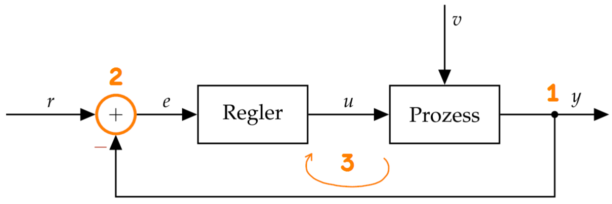
1.3 Steuerung (Feedforward Control)



1.4 Regelung (Feedback Control)



Ziel eines Reglers ist die Angleichung einer Regelgrösse y an eine Führungsgrösse r. Hauptmerkmal des Reglers ist die Rückkopplung und die Fehlergrösse e. Das System versucht die Fehlergrösse möglichst auf 0 zu behalten, was $r = y$ bedeutet, also die Ist-Grösse entspricht der Soll-Grösse.



r : Führungsgrösse (Soll-Wert)
e : Regelfehler
u : Stell-/Steuergrösse
y : Regelgrösse (Ist-Wert)
v : Störgrösse

Merkmale einer Regelung

Folgende Merkmale **muss** eine Regelung aufweisen. Liegt eines nicht vor, so handelt es sich nicht um eine Regelung.

1. Erfassung (Messen) der Regelgrösse
2. Vergleich von Regel- und Führungsgrösse
3. Geschlossener Wirkungskreis

$$y = R \cdot P \cdot e = R \cdot P\,(r - y) = R \cdot P \cdot r - R \cdot P \cdot y$$

$$y + R \cdot P \cdot y = R \cdot P \cdot r \quad \Rightarrow \quad \frac{y}{r} = \frac{R \cdot P}{1 + R \cdot P} \overset{!}{=} \underline{1}$$

1.4.1 Eigenschaften

- **Robustheit** –
- **Dynamik** –
- **Modularität** –
- **Genauigkeit** –
- **Herauserforderungen** –

2. Modellierung

2.1 Zustandsraum

2.2 Störverhalten

Das Störverhalten beschreibt den Einfluss der Störgrössen v auf die Regelgrösse y bei einer konstanten Führungsgrösse r. Ein gutes Störverhalten minimiert diese Einflüsse, wobei das letzendliche Verhalten des Systems abhängig auf das Zielsystem ist.

Beispiel

- Eine Beigemaschine darf keine Überschwingungen in der Regelgrösse haben, da dies zu einer Überbiegung führt, was ein *no-go* ist.
- Eine Saune kann sich dies eher noch erlauben, da eine Überschwingung nur einen kleinen Einfluss auf die Systemqualität hat.

i Merkmale

Die Störgrösse verfügt über vier Merkmale, welche für jedes System betrachtet soll:
- **Stabilität** –
- **Statischer Fehler / stationäre Genauigkeit** –
- **Überschwingen** –
- **Schnelles Erreichen des stationären Wertes** –

2.3 Führungsverhalten

Das Führungsverhalten beschreibt den Einfluss der Führungs-/Sollgrösse r auf die Regel-/Istgrösse y bei (idealerweise) einer konstanten Störungsgrösse e. Ein gutes Führungsverhalten minimiert die Ausschwingungen und Trägheit der Regelgrösse zur Führungsgrösse.

2.4 Vorsteuerung

2.5 Stationären Zustand (steady state)

3. Dynamik

3.1 Stabilität

Die Stabilität eines Systems wird in drei Zustände unterschieden:

stabil, falls alle Zustände (unterschiedliche Anfangspositionen) in der Nähe der Gleichgewichtslage x_e zu Lösungen führen. **asymptotisch stabil**, falls alle Zustände in der Nähe von x_e nach langer Zeit ($t \to \infty$) in x_e enden. **instabil**, falls der Zustand nie eine Gleichgewichtslage erreicht.

i Beispiel

Das Pendel, welches die gesamte Rotationsachse (360, rundherum) ausnützen kann, hat zwei Gleichgewichtslagen. Eine Lage ist wenn die Pendelmasse nach oben gerichtet ist und eine andere wenn die Masse nach unten gerichtet ist.
Wird das Pendel in diese beiden Lage gelegt, ist das System **stabil**. Wird es nach links oder nach rechts gerichtet losgelassen, dauert es eine Weile bis es die eine Gleichgewichtslage erreicht, dies nennt man **asymptotisch stabil**. Würde es einen Zustand geben, wo das Pendel nie "still steht", nennt man dies **instabil**.

4. Linearität & Zeitinvarianzen

4.1 Adjunkte adj(A)

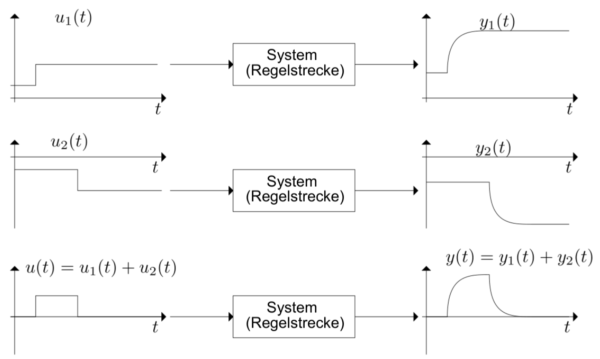
$$\text{adj}(A) =$$

4.2 LTI-Systeme

4.2.1 Überlagerungsprinzip

! Definition

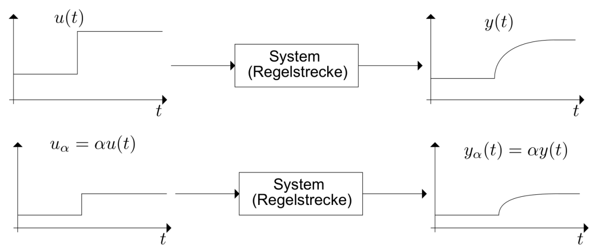
Wenn $y_1(t)$ die Antwort auf $u_1(t)$ ist und $y_2(t)$ die Antwort auf $u_2(t)$ ist, so ist $y_1(t)+y_2(t)$ die Antwort auf $u_1(t) + u_2(t)$.



4.2.2 Verstärkungsprinzip

! Definition

Wenn $y(t)$ die Antwort auf $u(t)$ ist, $\alpha \cdot y(t)$ ist die Antwort auf $\alpha \cdot u(t)$.



4.3 Linearisierung

Vorgehen Linearisieren

(@) Differentialgleichung gleich 0 setzen $f(\cdots) = 0$

$$M \cdot \frac{d^2h}{dt^2} + \alpha \frac{dh}{dt} + k \cdot h^3 - M \cdot g = 0 \quad \rightarrow \quad f(\ddot{h}, \dot{h}, h) = 0$$

(@) In Linearisierungsgleichung einsetzten

$$\left. \frac{\delta f}{\delta \ddot{h}} \right|_{h=\bar{h}}$$

5. Grundelemente

5.1 Elementare Glieder

$$G(s) = \frac{b_0 \cdot s^m + b_1 \cdot s^{m-1} + \cdots + b_m}{s^n + a_1 \cdot s^{n-1} + \cdots + a_n}$$

$$= b_0 \cdot (s + z_1) \cdot (s + z_2) \cdots (s + z_m) \cdot \frac{1}{s + p_1} \cdot \frac{1}{s + p_2} \cdots \frac{1}{s + p_n}$$

m : Nullstellen $z_{1 \dots m}$
n : Polstellen $p_{1 \dots m}$

5.1.1 Elementare Funktionen

Mit Parametern $k, a, \zeta, \omega_0, \tau \in \mathbb{R}$

$G(s) = k$: konstanter Faktor
$G(s) = s + a$: einfache reelle Nullstelle
$G(s) = s^2 + 2\zeta\omega_0 s + \omega_o^2$: konj. komplexe Nullstellen ($\zeta \leq 1$)
$G(s) = \frac{1}{s+a}$: einfacher reller Pol
$G(s) = \frac{1}{s^2+2\zeta\omega_0 s+\omega_o^2}$: konj. komplexe Pole ($\zeta \leq 1$)
$G(s) = e^{-s\tau}$: Totzeitglied $\tau > 0$

Die zugehörigen Nullstellen

$$\lambda = \begin{cases} -a & \text{einfach reell} \\ -\zeta\omega_0 \pm \omega_0 \sqrt{\zeta^2 - 1} & konj.komplex \end{cases}$$

6. PID-Regler

Die Aufgabe eines Reglers besteht darin, ausgehen von einem Regelfehler e zum Zeitpunkt t eine Stellgrösse u so zu bestimmen, dass der Fehler in absehbarer Zeit reduziert wird.

Verhalten Regler

Grössere Fehler sollten zu grösseren Stellgrössen führen und kleinere Fehler zu kleineren Stellgrössen.

6.1 Proportional k_p

P-Anteil verstärkt den Regelfehler e um die *Proportionalverstärkung* k_p.

$$u = k_p(r - y) = k_p \cdot e$$

P-Regler

$e = 0$ ist mit einem P-Regler nicht möglich. Unter Annahme eines stabilen Regelkreises:

$$G_{er} = \frac{1}{1 + P \cdot C} = \frac{1}{1 + P \cdot k_p}$$

entsteht ein bleibender Fehler von:

$$G_{er}(0) = \frac{1}{1 + P(0) \cdot C(0)} = \frac{1}{1 + P(0) \cdot k_p}$$

Proportionalband

$$u = \begin{cases} u_{max} & \text{falls } e \geq e_{max} \\ k_p \cdot e & \text{falls } e_{min} < e < e_{max} \\ u_{min} & \text{falls } e \geq e_{min} \end{cases}$$

mit

$$e_{min} = \frac{u_{min}}{k_p} \qquad e_{max} = \frac{u_{max}}{k_p}$$

Permanentes Stellsignal u

Wird ein permanentes Stellsignal u benötigt, so gilt für den P-Regler einen bestimmten Fehler $e \neq 0$.

6.2 Integral k_i/T_i

Mit dem I-Anteil werden *vergangene* Fehler mitberechnet → stationäre Fehler des P-Anteils wird korrigiert.

$$u = k_i \cdot \int_0^t e(\tau) \, d\tau$$

Die Stellgrösse wird dadurch solange geregelt, bis der Regelfehler $e = 0$ wird.

6.3 Proportional k_d/T_d

Der D-Anteil reagiert auf *zukünftige* Fehler, indem die Steigung mit einem Verstärkungsfaktor k_d verstärkt wird.

$$e(t+T_d) \approx e(t) + T_d \frac{\mathrm{de}}{\mathrm{d}t} \qquad u = k_d \frac{\mathrm{de}}{\mathrm{d}t}$$

! Limitierung der D-Verstärkung

Grund: Für träge Prozess führt eine sprungartige Veränderung (z.B. Sprungantwort oder Dirac-Impuls) zu einem sprungartigen Regelfehler $e(t) \approx \sigma$.

6.4 Übertragungsfunktion

$$C(s) = k_p \left(1 + \frac{k_i}{s} + k_d \cdot s\right) = k_p \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s\right)$$

! Wichtig

Diese Beschreibung ist nur eine idealisierte Repräsentation, welche für das Verständnis des System hilfreich ist. Im praktischen Einsatz sind Modifikationen notwendig.

6.5 Auslegung

6.5.1 Anhand Bodediagramm

Diese Auslegung fokussiert anhand des **offenen** Kreises ($L = C \cdot P$) des Regelkreises.

$$C(s) = k_i \frac{(1+s\,T_1)(1+s\,T_2)}{s} = k_p \frac{(1+s\,T_i)(1+s\,T_d)}{s \cdot T_i}$$

Zielgrössen: Durchtrittsfrequenz ω_{gc}, die Phasenreserve φ_m und allenfalls Amplitudenreserve g_m.

Vorgehen

Prozess: $P(s) = \frac{10}{(1+s)^2}$ mit Ziel $\omega_{gc} \geq 10\frac{rad}{s}$, $\varphi_m \geq 50$.

1. P-Regler für Erreichung von ω_{gc}. Mit $|k_p \cdot P(j\omega_{gc})| = 1$ (Nyquist-Kriterium) folgt:

$$k_p = \frac{1}{\left|\frac{10}{1+10j}\right|} = \frac{(\sqrt{1^2+10^2})^2}{10} = 10.1$$

$$C(s) = k_p = 10.1$$

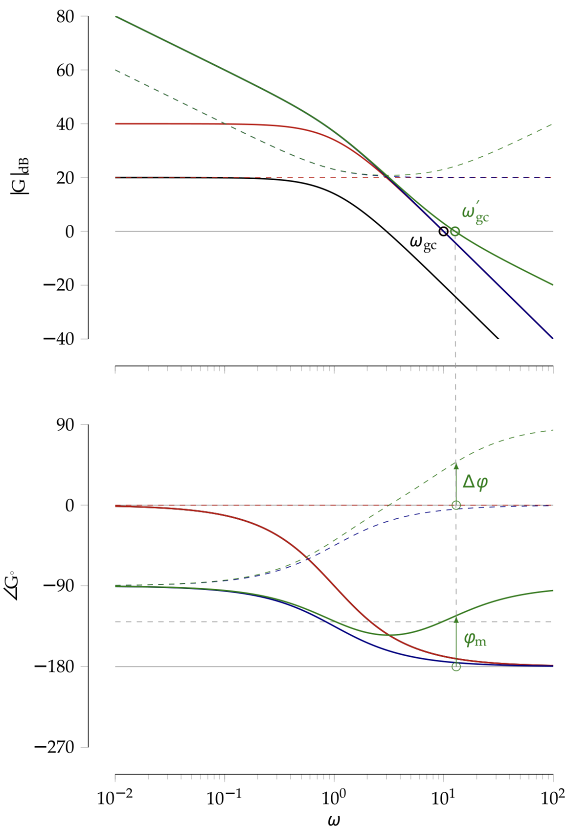
2. PI-Regler für Reduktion der zusätzlichen Phasensenkung im Bereich von ω_{gc}

$$C(s) = k_i \cdot \frac{(1+s\cdot T_1)}{s} = \frac{10\cdot(1+s)}{s}$$

1. PID-Regler für genügend Phasenabhebung im Bereich von ω_{gc}

$$C(s) = k_i \cdot \frac{(1+s\cdot T_1)(1+s\cdot T_2)}{s} = 10 \cdot \frac{(1+s)(1+0.1s)}{s}$$

4. Kontrolle von resultiernden Durchtrittsfrequenz ω'_{gc} und damit ergebenden Phasenreserve φ_m.



6.5.2 Anhand von Einstellregeln

6.6 Stellgrössen-Sättigung

Sättigungseffekt

6.6.1 Windup & Anti-Windup

Windup entsteht durch

7. MATLAB

7.1 Vektoren

Vektoren werden mit `[...]` deklariert. Elemente werden Spaltenweise mit einem Leerschlag `' '` oder Komma `,` eingeteilt und mit einem Semikolon `;` Reihenweise geteilt.

```
data = [1,2,3;4,5,6;7,8,9]; % same as [1 2 3;4 5 6;7 8
↪  9];
```

Grösse `size`

Mit `size` kann die Grösse einer Variable ermittelt werden. `size` gibt als Resultat ein 1x2 Vektor zurück ($\begin{bmatrix}\text{Rows} & \text{Columns}\end{bmatrix}$)

```
>> a = 1
>> size(a)
     1     1  % rows, columns
```

```
a = 1
```

$$\begin{bmatrix}1\end{bmatrix} \qquad \text{oder einfach} \qquad 1$$

Die `size`-Funktion gibt auch bei einzelnen Werte eine Grösse aus, nämlich $\begin{bmatrix}1 & 1\end{bmatrix}$

```
b = [1 2 3] % Linienvektor
```

$$\begin{bmatrix}1 & 2 & 3\end{bmatrix}$$

```
c = [2;3;4] % Spaltenvektor
```

$$\begin{bmatrix}2\\3\\4\end{bmatrix}$$

Slicing

Mit *Slicing* kann ein Teil einer Matrix **kopiert** werden und einer anderen Variable zugewiesen werden.

```
<matrix>(<rowStart>:<rowEnd>,<colStart>:<colEnd>)
```

7.2 Plotting

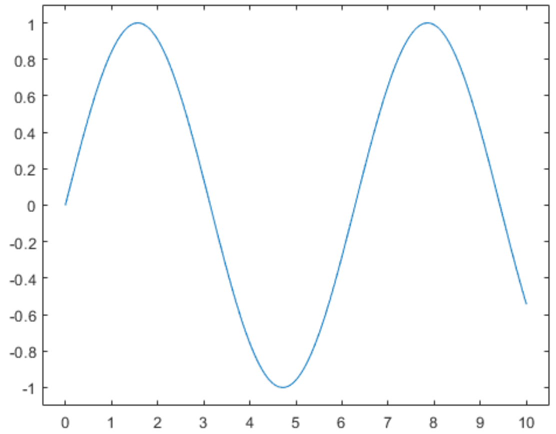
Figure-Separierung

Mit `figure(n)` können mehrere Plot-Befehle in eigene Figuren geladen werden.

7.2.1 XY-Graph

```
figure(1);
t = 0:0.5:10;
y = sin(t);

plot(t,y);
xlim([-0.5 10.5]);
ylim([-1.1 1.1]);
```



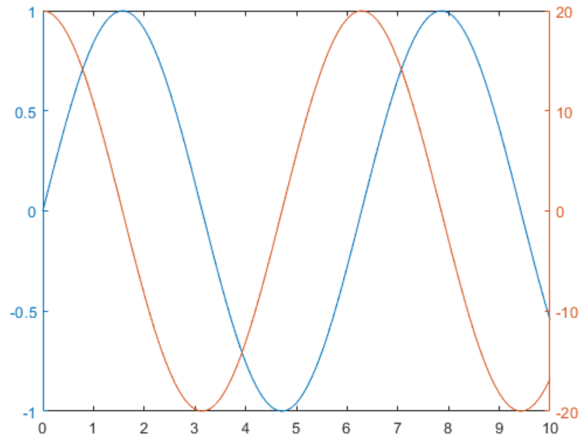
7.2.2 XYY-Graph

Mit `yyaxis` kann die Y-Achse beim selben Plot mit `left` & `right` gewechselt werden.

```
figure(1);
t = 0:0.5:10;

yyaxis left;
plot(t, sin(t));
xlim([-0.5 10.5]);
ylim([-1.1 1.1]);

yyaxis right;
plot(t, 20*cos(t));
xlim([-0.5 10.5]);
ylim([-20.5 20.5]);
```



7.3 Transferfunktion `tf(...)`

Mit dem Befehl `tf(...)` kann eine Transferfunktion deklariert werden mit Zähler- und Nenner-Zeilenvektoren.

```
sys = tf(numerator,denominator);
```

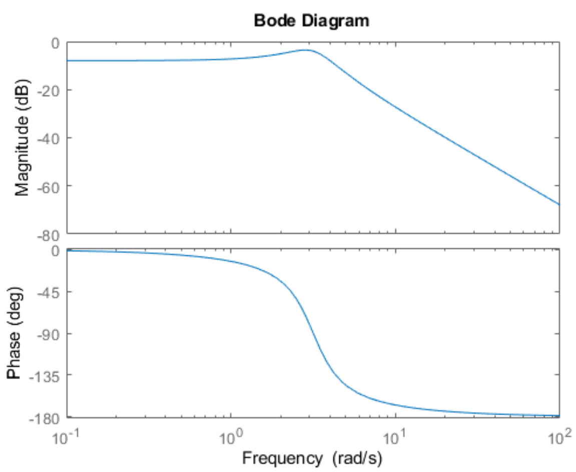
Die Transferfunktion kann in anderen Funktion wiederverwendet werden, wie zum Beispiel `step` oder `bode`. Folgende Beispiele sind mit der `sys`-Transferfunktion (folgende Gleichung) gemacht.

$$G_{sys}(s) = \frac{4}{s^2 + s + 10}$$

```
sys = tf(4,[1 2 10]);
```

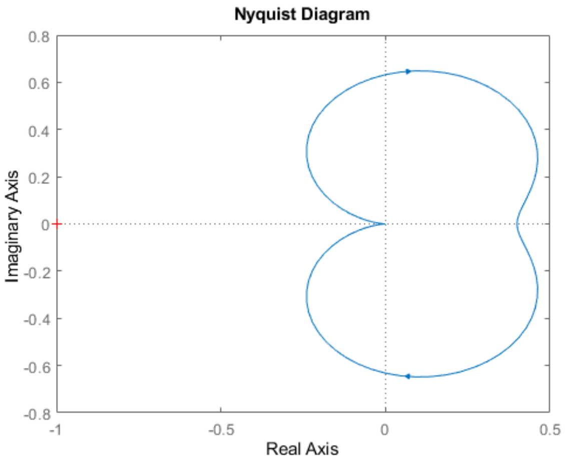
7.3.1 Bode-Diagramm `bode`

```
bode(sys,{0.1,100}); % or bode(sys);
% grid on; to enable Grid in Plot
```



7.3.2 Nyquist-Diagramm `nyquist`

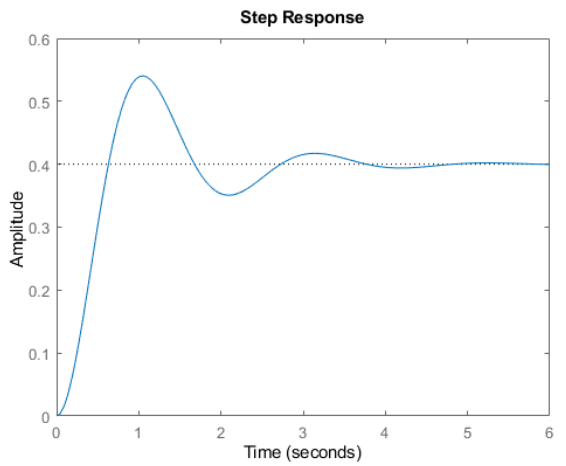
```
nyquist(sys)
```



7.3.3 Sprungantwort `step`

Mit `step(...)` kann eine Transferfunktion mit der Sprungfunktion σ verwendet werden. Damit

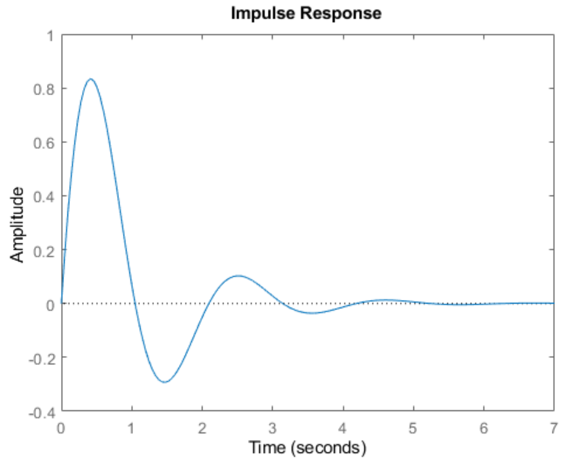
```
step(sys);
```



7.3.4 Impulsantwort `impulse`

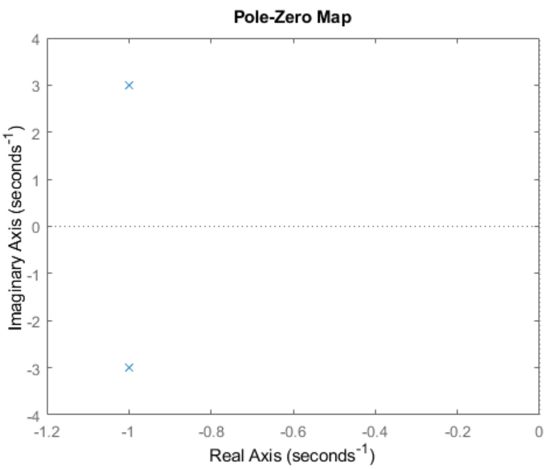
Mit `impulse(...)` kann die Impulsantwort der Transferfunktion ausgegeben werden.

```
impulse(sys);
```



7.3.5 Pol-Nullstellen-Diagramm `pzmap`

```
pzmap(sys);
ylim([-4 4]); xlim([-1.2 0]);
```

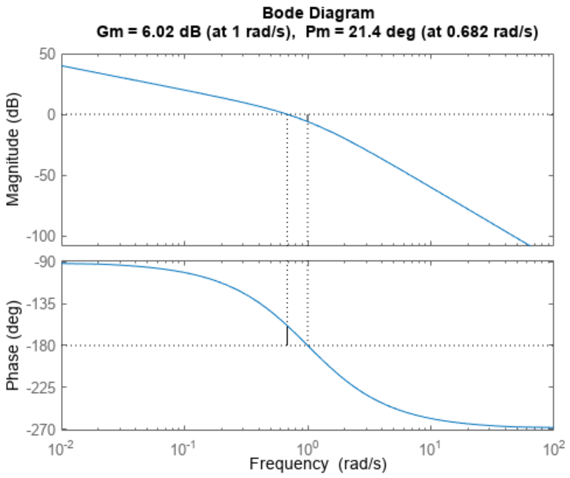


> **MATLAB Zauber**
>
> Damit die Pol- und Nullstellen erkennbar sind, muss eventuell mit den Darstellungsgrenzen gespielt werden.

7.4 Margin `margin(tf)`

Mit dem Befehl `margin(tf)` kann das Bode-Diagramm

```
margin(tf(1,[1 2 1 0]))
```



7.5 Zustandsraumdarstellung `ss()`

Mit `ss(...)` können vier Matrizen A, B, C, D zu einer Zustandsraumdarstellung zusammengeführt werden.

```
A = [0 1;-5 -2];
B = [0;3];
C = [0 1];
D = 0;
Ts = 0.25;
sys = ss(A,B,C,D,Ts);
```

Es kann ebenfalls `bode`, `nyquist`, `step`, etc. angewendet werden, da die ZRD eine andere Darstellung der Übertragungsfunktion ist.

8. Simulink

Warum Simulink?

In MATLAB können Übertragungsfunktionen berechnet werden und Regelkreise simuliert werden. Warum trotzdem Simulink verwenden?

9. Glossar

- *SISO* – **S**ingle **I**nput **S**ingle **O**utput
- *MIMO* – **M**ultiple **I**nput **M**ultiple **O**utput