

Regelungstechnik

Zusammenfassung

Joel von Rotz

2023-03-28

 [Quelldateien](#)

Inhaltsverzeichnis

1	Einführung	3
1.1	Rückkopplung	3
1.2	Mit- und Gegenkopplung	3
1.3	Steuerung (Feedforward Control)	3
1.4	Regelung (Feedback Control)	3
1.4.1	Eigenschaften	3
2	Modellierung	3
2.1	Zustandsraum	3
2.2	Störverhalten	3
2.3	Führungsverhalten	4
2.4	Vorsteuerung	4
2.5	Stationären Zustand (steady state)	4
3	Dynamik	4
3.1	Stabilität	4
4	Linearität & Zeitinvarianzen	4
4.1	Adjunkte $\text{adj}(A)$	4
4.2	LTI-Systeme	4
4.2.1	Überlagerungsprinzip	4
4.2.2	Verstärkungsprinzip	4
4.3	Linearisierung	5
5	Grundelemente	5
5.1	Elementare Glieder	5
5.1.1	Elementare Funktionen	5
6	MATLAB	5
6.1	Vektoren	5
6.2	Plotting	5
6.2.1	XY-Graph	6
6.2.2	XYZ-Graph	6
6.3	Transferfunktion $\text{tf}(\dots)$	6
6.3.1	Bode-Diagramm <code>bode</code>	6
6.3.2	Nyquist-Diagramm <code>nyquist</code>	7
6.3.3	Sprungantwort <code>step</code>	7
6.3.4	Impulsantwort <code>impulse</code>	7
6.3.5	Pol-Nullstellen-Diagramm <code>pzmap</code>	7
6.4	Zustandsraumdarstellung <code>ss()</code>	7
7	Simulink	8

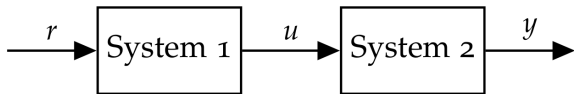
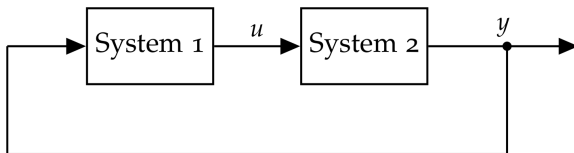
8 Glossar

8

1. Einführung

1.1 Rückkopplung

Rückkopplung beschreibt eine Anordnung, bei welcher zwei oder mehr dynamische 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.

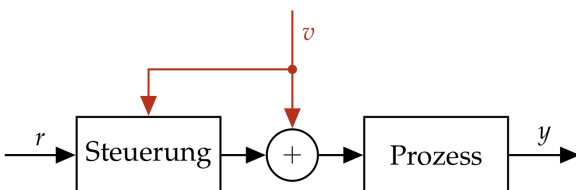
Vorsicht

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.

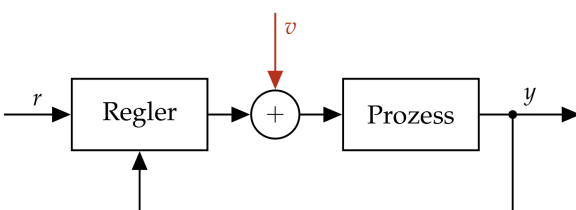
1.2 Mit- und Gegenkopplung

Beide beschriebenen Systeme arbeiten nach dem Prinzip der *Gegenkopplung*

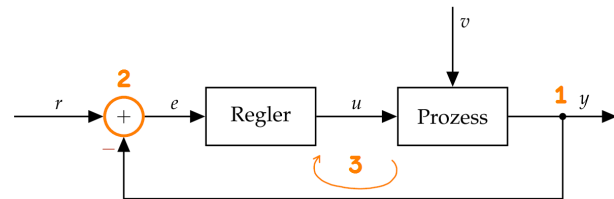
1.3 Steuerung (Feedforward Control)



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

i 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

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

$$y + R \cdot P \cdot y = R \cdot P \cdot r \Rightarrow \frac{y}{r} = \frac{R \cdot P}{1 + R \cdot P} \stackrel{!}{=} 1$$

1.4.1 Eigenschaften

- **Robustheit** –
- **Dynamik** –
- **Modularität** –
- **Genauigkeit** –
- **Herausforderungen** –

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 letztendliche 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 Sauna 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 Störgrösse 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 ($t \rightarrow \infty$) 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 $\text{adj}(A)$

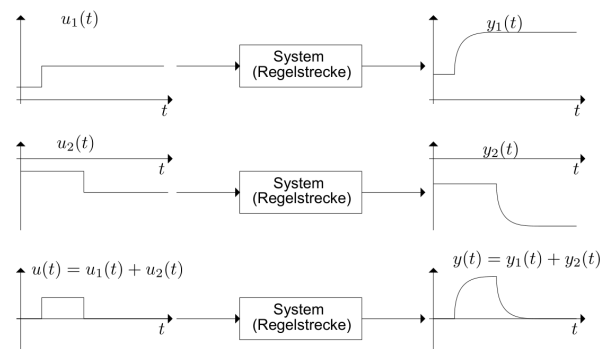
$$\text{adj}(A) =$$

4.2 LTI-Systeme

4.2.1 Überlagerungsprinzip

! Definition

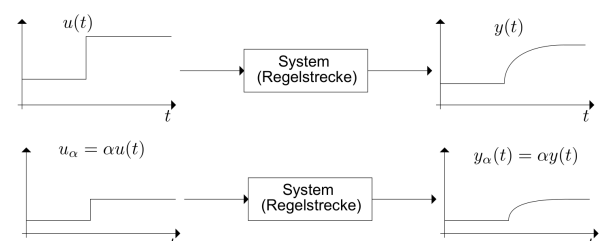
Wenn $y_1(t)$ die Antwort auf $u_1(t)$ ist und $y_2(t)$ die Antwort auf $u_2(t)$ ist, so ist $y_1(t) + y_2(t)$ die Antwort auf $u_1(t) + u_2(t)$.



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 $f(\dots) = 0$

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

(@) In Linearisierungsgleichung einsetzen

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

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

i Grösse size

Mit `size` kann die Grösse einer Variable ermittelt werden. `size` gibt als Resultat ein 1x2 Vektor zurück ([Rows Columns])

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

5. Grundelemente

5.1 Elementare Glieder

$[1]$ oder einfach 1

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

Die `size`-Funktion gibt auch bei einzelnen Werte eine Grösse aus, nämlich $[1 \ 1]$

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

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

m : Nullstellen $z_{1\dots m}$

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

$[1 \ 2 \ 3]$

5.1.1 Elementare Funktionen

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

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

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

Die zugehörigen Nullstellen

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

💡 Slicing

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

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

6. MATLAB

6.1 Vektoren

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

6.2 Plotting

i Figure-Separierung

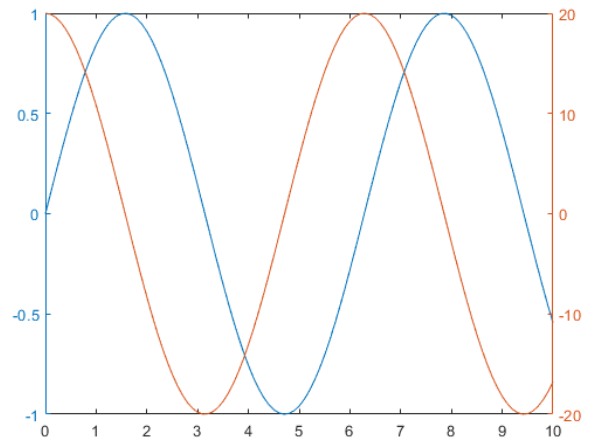
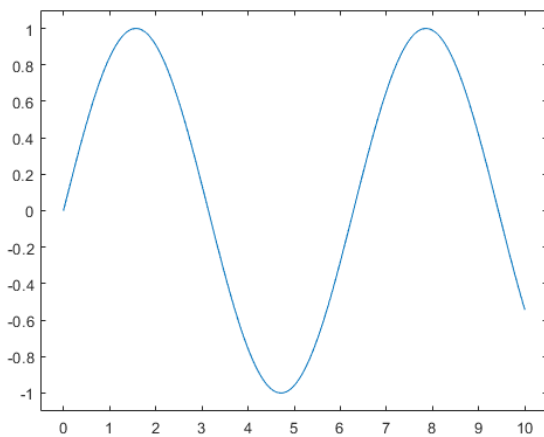
Mit `figure(n)` können mehrere Plot-Befehle in eigene Figuren geladen werden.

6.2.1 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]);
```



6.3 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.

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

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

6.2.2 XYY-Graph

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

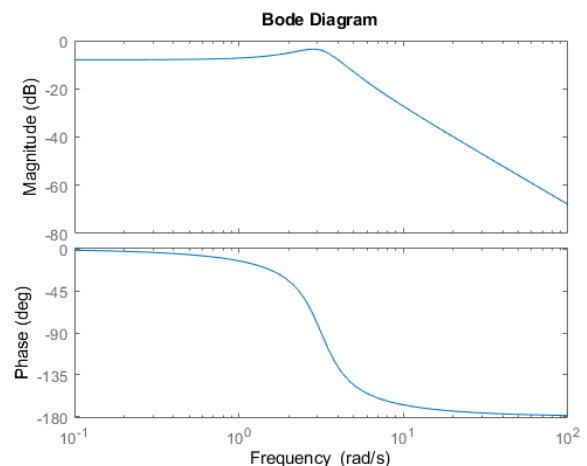
```
figure(1);
t = 0:0.5:10;

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]);
```

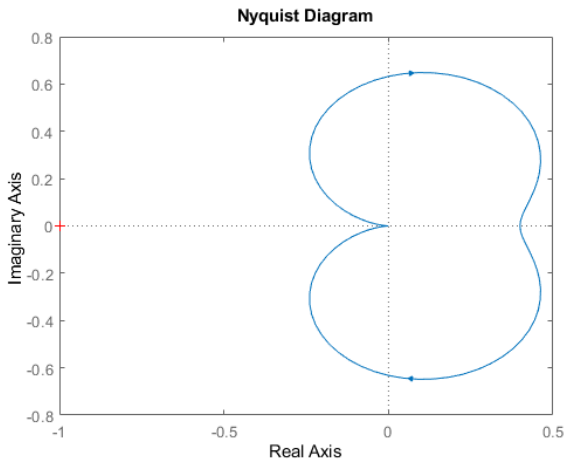
6.3.1 Bode-Diagramm bode

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



6.3.2 Nyquist-Diagramm `nyquist`

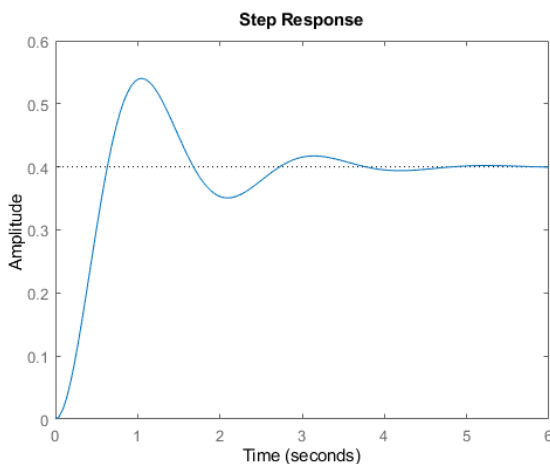
```
nyquist(sys)
```



6.3.3 Sprungantwort `step`

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

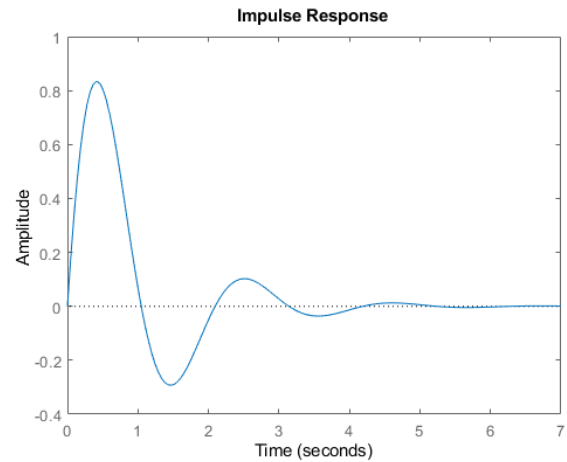
```
step(sys);
```



6.3.4 Impulsantwort `impulse`

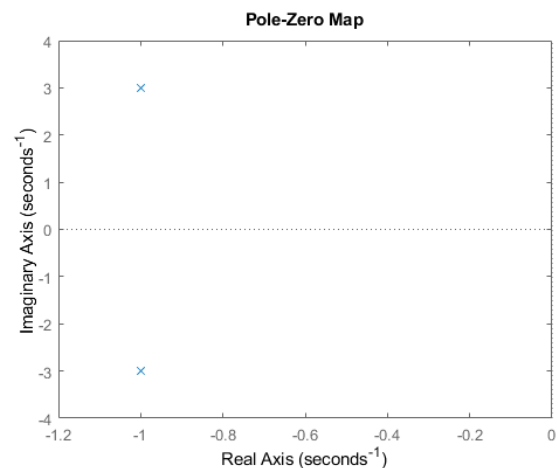
Mit `impulse(...)` kann die Impulsantwort der Transferfunktion ausgegeben werden.

```
impulse(sys);
```



6.3.5 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.

6.4 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.

7. Simulink

8. Glossar

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