



Regelungstechnik 4

FS 2025 – Prof. Dr. Markus Kottmann

Autoren: Simone Stitz

<https://gitlab.com/sstitz/regelungstechnik-4>

V1.0.20250701

Inhaltsverzeichnis

I Regelungstechnik 4

1 Zustandsraumdarstellung (ZRD)

1.1	Zustandsraumdarstellung von MIMO-Systemen (S. 153-154)	2
1.2	Zustandsraumdarstellung von SISO-Systemen (S. 154)	2
1.3	Verfahren mit ZRD – Übersicht	2
1.4	ZRD via physikalischer Modellbildung	2
1.5	Linearisierung einer ZRD (S. 168)	2
1.6	Zustandsraumdarstellung aus UTF oder DGL (S. 169-170)	2
1.7	ZRD im Bildbereich	3
1.8	Zustandsraumdarstellung aus Blockdiagramm (S. 171)	3
1.9	Blockdiagramm aus ZRD / Gleichung	3
1.10	Aggregation von Teilsystemen (S. 172-174)	3
1.11	Ähnlichkeitstransformation (S. 175)	3

2 Anwendungen der ZRD

2.1	Lösung der ZRD im Zeitbereich (S. 176-180)	3
2.2	Transitionsmatrix	3
2.3	Entkopplung mittels Diagonalform (S. 180-185)	4
2.4	Diagonalisierung (S. 182-185)	4
2.5	Stabilität	4
2.6	Steuerbarkeit (S. 185-187)	4
2.7	Beobachtbarkeit (S. 189)	4
2.8	Minimalität	5
2.9	Normalformen	5
2.10	Eigenschaften von Zustandsraummodellen (S. 190-193)	5

3 Zustandsrückführungen

3.1	Eigenschaften von Zustandsrückführungen	5
3.2	Reglerentwurf mittels Polvorgabe (S. 194-196)	5
3.3	LQR-Verfahren (Linear Quadratic Regulator) (S. 196-199)	5
3.4	Statische Erweiterungen der Zustandsrückführung	6

4 Beobachter

4.1	Dualität: Steuerbarkeit – Beobachtbarkeit (S. 205)	7
4.2	Entwurf eines Beobachters mittels Polvorgabe (S. 205-207)	7
4.3	LQG-Verfahren (Linear Quadratic Gaussian Estimation)	7
4.4	Varianten von Beobachtern (S. 209-211)	7
4.5	Implementierung eines Zustandsreglers mit Beobachter	7

5 Digitale Regelung

5.1	Regelkreis mit digitalem Regler (S. 214)	8
5.2	Signale im digitalen Regelkreis (S. 184)	8
5.3	Reglerentwurf – direkt vs. indirekt (S. 224-225)	8

6 Indirekter digitaler Reglerentwurf

6.1	Approximationen für Diskretisierung (S. 226)	8
6.2	Vorgehen: Diskretisierung eines Reglers (S. 226-228)	9
6.3	Empirische Einstellregeln für digitale PID-Regler	9

7 z-Transformation / Zeitdiskrete Übertragungsfunktion

7.1	Folgen (und deren z-Transformation) (S. 230)	9
7.2	z-Transformation – Definition (S. 232)	9
7.3	z-Transformierte einiger Funktionen (S. 232)	9
7.4	Eigenschaften der z-Transformation (S. 235)	9
7.5	Rücktransformation	9
7.6	Beispiele zur z-Transformation	9
7.7	Zeitdiskrete Übertragungsfunktion (S. 238-239)	10

7.8	Stabilität einer zeitdiskreten Übertragungsfunktion (S. 240-241)	10
7.9	Frequenzgang / Bodediagramm / Nyquistdiagramm (S. 241-243)	10

8 Zeitdiskrete Zustandsraumdarstellung

8.1	Zeitdiskrete ZRD (S. 243-244)	10
8.2	Stabilität (S. 245)	10
8.3	Steuerbarkeit (S. 245)	10
8.4	Beobachtbarkeit (S. 245)	10

9 Direkter digitaler Reglerentwurf

9.1	Direkter digitaler Reglerentwurf – Vorgehen	11
9.2	ZOH-Diskretisierung – Überblick	11
9.3	ZOH-Diskretisierung mit UTF (S. 248-249)	11
9.4	ZOH-Diskretisierung im Zustandsraum (S. 250-251)	11
9.5	Reglerentwurf – WOK (S. 254)	11
9.6	Reglerentwurf – Vorgabe des geschl. Regelkreises (S. 256-257)	12
9.7	Reglerentwurf – Deadbeat-Regler (S. 256-257)	12
9.8	Reglerentwurf – Zustandsraum (S. 257-258)	12

II Anhang

10 Mathematik-Hilfen

10.1	Trigonometrie	13
10.2	Lineare Algebra – Repetition	13
10.3	Dezibel dB	13
10.4	Spezielle Ansätze PBZ	13
10.5	Laplace-Transformation	14

11 Wurzelortskurven (WOK)

11.1	Grundsätzliches Aussehen der WOK	14
11.2	Zeichnen von Wurzelortskurven (S. 96-98)	14
11.3	WOK zu einigen geläufigen Pol- / NS-Verteilungen (S. 99)	14
11.4	Auslegen eines statischen P-Reglers mittels WOK (S. 101-102)	15

12 Matlab

12.1	Zustandsraumdarstellung (ZRD)	15
12.2	Bode- und Nyquist	15
12.3	z-Transformation	15
12.4	Implementation eines diskreten Reglers	15
12.5	Wurzelortskurven	15
12.6	Zustandsrückführungen / Beobachter	15

13 Statische Grundglieder (ohne Gedächtnis)

13.1	P-Glied (Proportional)	16
------	------------------------	----

14 Dynamische Grundglieder (mit Gedächtnis)

14.1	Totzeit-Glied	16
14.2	I-Glied (Integrierer)	16
14.3	PT1-Glied	16
14.4	PT2-Glied	16
14.5	D-Glied (Idealer Differenzierer)	16
14.6	DT1-Glied (Realisierbarer Differenzierer)	16
14.7	Lead-Glied / Lag-Glied / Lead-Lag-Glied	16

15 Grundglieder – Frequenzbereich

16 Analogrechner

I Regelungstechnik 4

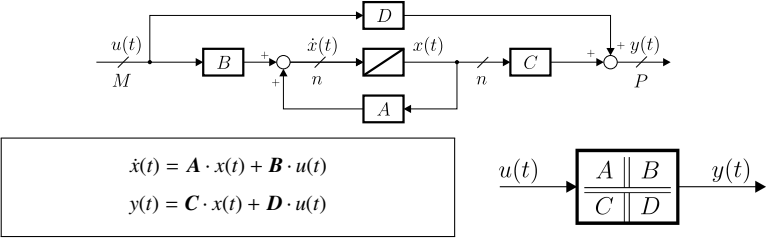
1 Zustandsraumdarstellung (ZRD)

1.1 Zustandsraumdarstellung von MIMO-Systemen (S. 153-154)

Der Vorteil der Zustandsraumdarstellung gegenüber der I/O-Beschreibung liegt darin, dass die internen Signale des Systems (Zustandsvariablen) mitmodelliert werden.

Die ZRD kann aus dem Blockdiagramm abgeleitet werden. Umgekehrt kann aus der ZRD das Blockdiagramm gezeichnet werden.

Die ZRD von MIMO-Systemen wird folgendermassen beschrieben:



$u(t) \in \mathbb{R}^M$	Eingangsvektor	fasst Eingangsgrössen in Vektor zusammen
$x(t) \in \mathbb{R}^n$	Zustandsvektor	fasst Zustände in Vektor zusammen
$y(t) \in \mathbb{R}^P$	Ausgangsvektor	fasst Ausgangsgrössen in Vektor zusammen
$A \in \mathbb{R}^{n \times n}$	Systemmatrix	beschreibt (Eigen-)Dynamik des Systems, beschreibt Stabilität des Systems
$B \in \mathbb{R}^{n \times M}$	Eingangsmatrix	beschreibt, wie der Eingang auf den Zustandsvektor einwirken kann
$C \in \mathbb{R}^{P \times n}$	Ausgangsmatrix	beschreibt, welche Zustandsvariablen (oder Linearkombinationen davon) messbar sind
$D \in \mathbb{R}^{P \times M}$	Durchgangsmatrix	beschreibt direkte Kopplung zwischen Ein- und Ausgang

Hinweis: Für Regelstrecken gilt aufgrund ihres Tiefpassverhaltens meist $D = 0$

1.1.1 ZRD für DGL der Ordnung n

Für eine DGL der Ordnung n werden die Zustände des Zustandsvektors x jeweils so gewählt, dass der erste Zustand der **unabgeleiteten** Variable entspricht und der letzte Zustand der $n - 1$ -ten Ableitung.
⇒ 'Kette' von Integratoren

$$x = \begin{bmatrix} x \\ \dot{x} \\ \vdots \\ x^{(n-1)} \end{bmatrix} \Rightarrow \dot{x} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \vdots \\ x^{(n)} \end{bmatrix}$$

1.2 Zustandsraumdarstellung von SISO-Systemen (S. 154)

$\dot{x}(t) = A \cdot x(t) + b \cdot u(t)$	Wegen $M = P = 1$ werden die Matrizen B und C zu Vektoren. D wird zu einem Skalar.
$y(t) = c \cdot x(t) + d \cdot u(t)$	

1.3 Verfahren mit ZRD – Übersicht

Zum Thema ZRD werden die folgenden Verfahren genauer betrachtet, bevor diese auf die Regelungstechnik angewendet werden.

- ZRD aus Modellierung nach physikalischen Prinzipien 1.4
- Linearisierung einer nichtlinearen ZRD 1.5
- ZRD aus einer UTF oder DGL bestimmen 1.6
- Herauslesen der ZRD aus einem Blockdiagramm 1.8
- Aggregation von Teilsystem (Serie-, Parallel-, Kreisschaltung) 1.10
- Ähnlichkeitstransformation durch andere Wahl des Zustandsvektors 1.11

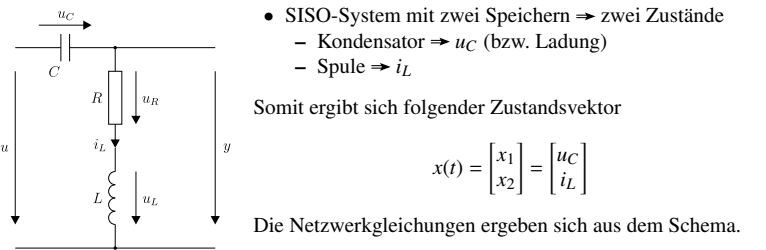
1.4 ZRD via physikalischer Modellbildung

Aus einem physikalischen System werden die Eingangs- und Ausgangssignale, sowie die Zustände bestimmt. Diese werden mathematisch / physikalisch beschrieben und als ZRD dargestellt.

1.4.1 Vorgehen – Herleitung eines ZRD-Modells

1. Anzahl Speicher (Integratoren) bestimmen
2. Wahl der Zustandsgrössen
3. Beschreibung der **Änderung** (Ableitungen) der Zustandsgrössen als Gleichungen
4. Substitution der 'nicht-Zustandsgrössen' mittels algebraischer Gleichungen
5. Bilden der Zustands- und Ausgangsgleichungen
6. Darstellung in Matrix-Notation

Beispiel: ZRD eines RLC-Systems (S. 155-156)



Knoten: $C \cdot \dot{u}_C = i_C = i_L \Rightarrow \dot{u}_C = \frac{i_L}{C}$

Eingangsmasche: $L \cdot \dot{i}_L = u - u_C - u_R = u - u_C - R \cdot i_L \Rightarrow \dot{i}_L = \frac{u}{L} - \frac{u_C}{L} - \frac{R}{L} \cdot i_L$

Äussere Masche: $u = u_C + y \Rightarrow y = u - u_C$

Die drei (linearen) Gleichungen rechts können direkt in die ZRD gebracht werden, indem sie in Matrizen geschrieben werden.

$$\dot{x}(t) = \begin{bmatrix} \dot{u}_C \\ \dot{i}_L \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{C} \\ -\frac{R}{L} & -\frac{1}{L} \end{bmatrix} \cdot \begin{bmatrix} u_C \\ i_L \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix} \cdot u(t)$$
$$y(t) = \begin{bmatrix} -1 & 0 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} u_C \\ i_L \end{bmatrix}}_{x(t)} + 1 \cdot u(t)$$

1.5 Linearisierung einer ZRD (S. 168)

Die Linearisierung eines Systems erfolgt üblicherweise in einem Arbeitspunkt, damit relativ einfach ein Regler für diesem Arbeitspunkt entwickelt werden kann.

- Wenn ein System in einem Arbeitspunkt linearisiert wird, hat das zur Folge,
1. dass das Verhalten des nichtlinearen Systems in einer 'kleineren' Umgebung um den Arbeitspunkt erhalten bleibt.
 2. dass das linearisierte System dieses Verhalten, entsprechend 'gestreckt', im ganzen Zustandsraum zeigt.

1.5.1 Linearisierte ZRD

Die linearisierte ZRD für ein nichtlineares System in einem Arbeitspunkt mit $f(x_0, u_0) = 0$ und $y_0 = g(x_0, u_0)$ kann gewonnen werden, indem die 'Deltas' gebildet werden. (Die Δ -Symbole werden in der Notation jedoch oft weggelassen.)

Zustandsgleichung(en) $\dot{x} = f(x, u) \Rightarrow \Delta \dot{x} = A \cdot \Delta x + B \cdot \Delta u$

Ausgangsgleichung(en) $y = g(x, u) \Rightarrow \Delta y = C \cdot \Delta x + D \cdot \Delta u$

Dazu werden die folgenden Jacobi-Matrizen benötigt, welche im Arbeitspunkt (x_0, u_0) ausgewertet werden.

$$A = \left. \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \right|_{x_0, u_0}$$
$$B = \left. \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \dots & \frac{\partial f_n}{\partial u_M} \end{bmatrix} \right|_{x_0, u_0}$$
$$C = \left. \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \dots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_p}{\partial x_1} & \dots & \frac{\partial g_p}{\partial x_n} \end{bmatrix} \right|_{x_0, u_0}$$
$$D = \left. \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \dots & \frac{\partial g_1}{\partial u_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_p}{\partial u_1} & \dots & \frac{\partial g_p}{\partial u_M} \end{bmatrix} \right|_{x_0, u_0}$$

1.5.2 Vorgehen – Linearisierung

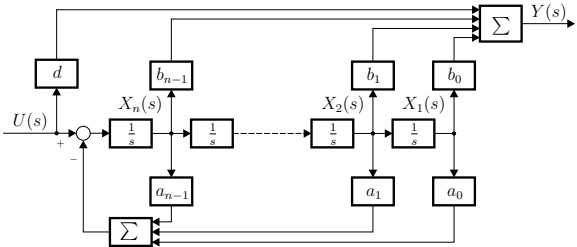
1. Statischen Arbeitspunkt (u_0, y_0) festlegen
Im AP sind alle Ableitungen 0 $\Rightarrow \dot{x} = f(x, u) \stackrel{!}{=} 0$
2. ZRD durch Berechnung der Jacobi-Matrizen linearisieren (Auswertung im AP)

1.6 Zustandsraumdarstellung aus UTF oder DGL (S. 169-170)

Für die folgenden Betrachtungen wird die UTF in einer Form betrachtet, in der ein allfälliger Proportionalteil durch den Term d repräsentiert wird.

$$G(s) = \frac{Y(s)}{U(s)} = d + \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \quad (m < n)$$

Ein entsprechendes Blockdiagramm in **Regelungsnormform** hat folgende Struktur:



In den Zeitbereich transformiert ergibt sich folgende Zustandsraumdarstellung:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \dots & 0 & 1 \\ -a_0 & -a_1 & \dots & -a_{n-2} & -a_{n-1} \end{bmatrix}}_A \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}}_B \cdot u$$
$$y = \underbrace{\begin{bmatrix} b_0 & b_1 & \dots & b_{n-2} & b_{n-1} \end{bmatrix}}_C \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + d \cdot u$$

1.6.1 Eindeutigkeit

- Aus einer UTF eine Zustandsraumdarstellung zu gewinnen ist **nicht eindeutig**. Auf triviale Art lassen sich alternative Formen produzieren (z.B. Zustandsvariablen anders nummerieren)
- Aus einer ZRD die UTF zu gewinnen ist **eindeutig**

1.7 ZRD im Bildbereich

Die ZRD existiert auch im Bildbereich und hat die Form:

$$\begin{aligned} sX(s) &= \mathbf{A} \cdot X(s) + \mathbf{B} \cdot U(s) + x(0) \\ Y(s) &= \mathbf{C} \cdot X(s) + \mathbf{D} \cdot U(s) \end{aligned}$$

1.7.1 UTF aus ZRD (Bildbereich)

Die UTF $G(s)$ kann durch den Zusammenhang rechts gewonnen werden.

$$\begin{aligned} Y(s) &= \mathbf{C} \cdot \overbrace{(\mathbf{sI} - \mathbf{A})^{-1} \cdot \mathbf{B} \cdot U(s)}^{X(s)} + \mathbf{D} \cdot U(s) \\ &= \underbrace{\left[\mathbf{C} \cdot (\mathbf{sI} - \mathbf{A})^{-1} \cdot \mathbf{B} + \mathbf{D} \right]}_{G(s)} \cdot U(s) \end{aligned}$$

Hinweis: In der UTF $G(s)$ kommen keine (statischen) Zustände x_0 mehr vor.

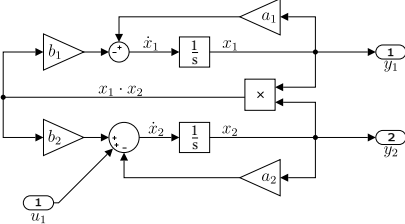
1.8 Zustandsraumdarstellung aus Blockdiagramm (S. 171)

Mittels folgendem Vorgehen kann die ZRD aus dem Blockdiagramm bestimmt werden:

1. Eingangs- und Ausgangssignale bestimmen
2. Zustandsvektor bilden, indem alle Speicher aufgelistet werden
⇒ Länge des Zustandsvektors entspricht Systemordnung
⇒ entspricht Summe der Ordnung der Teilsysteme
3. Gleichung $\dot{x} = f(x, u)$ bilden, indem alle Integratoreingänge (= $\dot{x}_i$) 'zurückverfolgt' werden, bis deren Abhängigkeit von Eingangssignalen & Zustandsvariablen definiert ist
4. Gleichung $y = g(x, u)$ bilden, indem Ausgangssignale (= y_i) 'zurückverfolgt' werden, bis deren Abhängigkeit von Eingangssignalen & Zustandsvariablen definiert ist
5. Bei linearen Systemen: Gleichungen f und g mit Matrizen $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ und $\mathbf{D}$ formulieren

Beim Ausgang anfangen und Integrator für Integrator in Richtung Eingang wandern. Für jeden 'Block' eine Gleichung aufstellen und diese in ZRD 'einfüllen'

Beispiel: ZRD aus Blockdiagramm



$$\begin{aligned} \dot{x}_1(t) &= f_1(x, u) &= a_1 \cdot x_1(t) &- b_1 \cdot x_1(t) \cdot x_2(t) \\ \dot{x}_2(t) &= f_2(x, u) &= -a_2 \cdot x_2(t) &+ b_2 \cdot x_1(t) \cdot x_2(t) + u_1(t) \\ y_1(t) &= g_1(x, u) &= x_1(t) \\ y_2(t) &= g_2(x, u) &= x_2(t) \end{aligned}$$

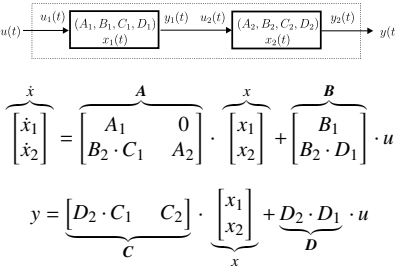
1.9 Blockdiagramm aus ZRD / Gleichung

- Auflösen nach höchster Ableitung
- Kette von Integratoren aufzeichnen
- Signale nach Integratoren wie benötigt abgreifen und Gleichung 'zusammenbauen'

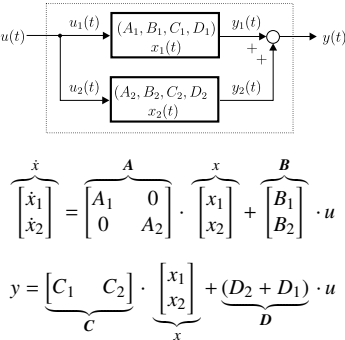
1.10 Aggregation von Teilsystemen (S. 172-174)

Hinweis: Grossbuchstaben und fett gedrucktes sind Matrizen

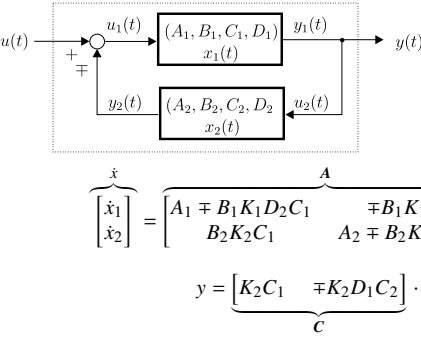
1.10.1 Serieschaltung



1.10.2 Parallelschaltung



1.10.3 Kreisschaltung



$$\begin{aligned} K_1 &= (\mathbf{I} \pm \mathbf{D}_2 \mathbf{D}_1)^{-1} \\ K_2 &= (\mathbf{I} \pm \mathbf{D}_1 \mathbf{D}_2)^{-1} \end{aligned}$$

SISO-Systeme: $K_1 = K_2$

1.11 Ähnlichkeitstransformation (S. 175)

Bei der Ähnlichkeitstransformation wird eine ZRD in eine andere ZRD transformiert, welche **dasselbe I/O-Verhalten**, aber einen **anderen internen Zustandsvektor** hat. Die Ähnlichkeitstransformation wird verwendet, um ein System in eine 'besser geeignete' Form zu bringen.

$$\begin{aligned} \dot{x}(t) &= \mathbf{A} \cdot x(t) + \mathbf{B} \cdot u(t) \\ y(t) &= \mathbf{C} \cdot x(t) + \mathbf{D} \cdot u(t) \end{aligned} \xrightarrow{\text{Ähnlichkeitstransformation}} \begin{aligned} \dot{\tilde{x}}(t) &= \tilde{\mathbf{A}} \cdot \tilde{x}(t) + \tilde{\mathbf{B}} \cdot u(t) \\ y(t) &= \tilde{\mathbf{C}} \cdot \tilde{x}(t) + \tilde{\mathbf{D}} \cdot u(t) \end{aligned}$$

Wenn der Zustand $x(t)$ mit einer **regulären konstanten Matrix $\mathbf{P}$** multipliziert wird, ergibt sich der neue Zustand $\tilde{x}(t)$. Die Umrechnungen (Ähnlichkeitstransformation) der ZRD erfolgt allgemein mittels.

$$\tilde{x} = \mathbf{P}x \quad \tilde{\mathbf{A}} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1} \quad \tilde{\mathbf{B}} = \mathbf{P}\mathbf{B} \quad \tilde{\mathbf{C}} = \mathbf{C}\mathbf{P}^{-1} \quad \tilde{\mathbf{D}} = \mathbf{D}$$

1.11.1 Eigenschaften der Ähnlichkeitstransformation

- Direkte I/O-Kopplung ($\tilde{\mathbf{D}} = \mathbf{D}$) bleibt wegen gleichem I/O-Verhalten unverändert
- Ändert Eigenwerte der Matrix **nicht**
- Ändert **nichts** an Steuerbarkeit, Beobachtbarkeit und Minimalität eines Systems
- Um z.B. in einer Simulation die korrekten Signalverläufe zu berechnen, muss eine Anfangsbedingung mittransformiert werden: $\tilde{x}(0) = \mathbf{P} \cdot x(0)$
- Wird ein System mit $\mathbf{P}_1 = \mathbf{P}$ transformiert und das resultierende System mit $\mathbf{P}_2 = \mathbf{P}^{-1}$ rücktransformiert, so ergibt sich wieder das ursprüngliche System

Beispiel: Zustandsvariablen in ZRD tauschen

Eine gegebene ZRD mit Zustandsvektor x soll mit Zustandsvektor $\tilde{x}$ ausgedrückt werden. Dafür wird die Transformationsmatrix $\mathbf{P}$ benötigt.

$$\Rightarrow \tilde{x} = \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{\mathbf{P}} \cdot \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_x$$

2 Anwendungen der ZRD

2.1 Lösung der ZRD im Zeitbereich (S. 176-180)

Durch eine Transformation der ZRD in den Frequenzbereich, algebraische Berechnungen und anschliessender Rücktransformation in den Zeitbereich ergibt sich die **Lösung der ZRD im Zeitbereich** als:

$$\begin{aligned} x(t) &= \underbrace{e^{\mathbf{A}t} \cdot x_0}_{x_F} + \underbrace{\int_0^t e^{\mathbf{A}(t-\tau)} \cdot \mathbf{B} \cdot u(\tau) \, d\tau}_{x_E} \\ y(t) &= \underbrace{\mathbf{C} \cdot e^{\mathbf{A}t} \cdot x_0}_{y_F} + \underbrace{\mathbf{C} \cdot \int_0^t e^{\mathbf{A}(t-\tau)} \cdot \mathbf{B} \cdot u(\tau) \, d\tau + \mathbf{D} \cdot u(t)}_{y_E} \end{aligned}$$

- x_F

Freie Antwort: System mit $u = 0$ aus Anfangszustand $x_0 = x(0)$ loslassen
- x_E

Erzwungene Antwort: System aus Zustand $x_0 = 0$ loslassen
⇒ nur abhängig von Eingangssignal u

Hinweis: $e^{\mathbf{A}t} = \Phi(t)$ heisst **Transitionsmatrix** ⇒ siehe Abschnitt 2.2

Eigenschaften – Freie Antwort:

Für die freie Antwort ergeben sich aus der Theorie der Eigenwerte / Eigenvektoren die folgenden Eigenschaften:

- Zeigt der Startwert $x_0 = v_i$ in die Richtung eines Eigenvektors, so bewegt sich der Zustand ausschliesslich in diese Richtung: $x(t) = e^{\lambda_i t} \cdot v_i = e^{\lambda_i t} \cdot x_0$
- Zeigt x_0 in eine beliebige Richtung, so kann $x(t)$ durch Superposition bestimmt werden
– x_0 wird in Richtung der Eigenvektoren v_i zerlegt und jede dieser Komponenten folgt bei der zeitlichen Entwicklung dem Faktor $e^{\lambda_i t}$

2.1.1 Impulsantwort $g(t)$

Ist die Transitionsmatrix $e^{\mathbf{A}t} = \Phi(t)$ bekannt, so ist die **Impulsantwort** $g(t)$

$$g(t) = \mathbf{C} \cdot e^{\mathbf{A}t} \cdot \mathbf{B} + \mathbf{D} \cdot \delta(t) = \mathbf{C} \cdot \Phi(t) \cdot \mathbf{B} + \mathbf{D} \cdot \delta(t)$$

2.2 Transitionsmatrix $e^{\mathbf{A}t} = \Phi$ (S. 177)

Die Transitionsmatrix (auch Fundamentalmatrix genannt) ist (wie oben bereits angewandt) definiert als:

$$e^{\mathbf{A}t} = \Phi(t)$$

Sie wird benötigt, um die Zustandsraumdarstellung im **Zeitbereich** zu lösen. Es gibt grundsätzlich mehrere Methoden, die quadratische ($n \times n$) Transitionsmatrix zu bestimmen.

Hier soll allerdings die analytische Berechnung mittels inverser Laplace-Transformation erwähnt sein:

$$\Phi(t) = e^{\mathbf{A}t} = \mathcal{L}^{-1}\{(\mathbf{sI} - \mathbf{A})^{-1}\}$$

Hinweis: Die Rücktransformation erfolgt erst im letzten Schritt.
⇒ Jedes Element der Matrix wird **einzel**n rücktransformiert

Eigenschaften der Transitionsmatrix $\Phi(t) = e^{\mathbf{A}t}$:

$$\Phi(0) = \mathbf{I} \quad | \quad \dot{\Phi}(t) = \mathbf{A} \cdot \Phi(t) (= \Phi(t) \cdot \mathbf{A}) \quad | \quad \Phi(t) \text{ ist stets invertierbar}$$

Transitionsmatrix einer Diagonalmatrix:

Die Transitionsmatrix $\Phi(t) = e^{At}$ kann einfach berechnet werden, wenn A eine Diagonal-matrix ist.

A_{Diag} = [lambda_1 0; 0 lambda_2] => e^{A_{Diag}t} = [e^{lambda_1t} 0; 0 e^{lambda_2t}]

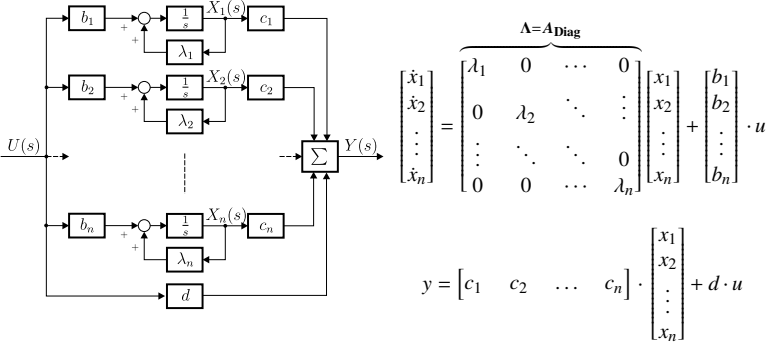
Hinweis: Diagonalisierung einer Matrix: siehe Abschnitt 2.4

2.3 Entkopplung mittels Diagonalform (s. 180-185)

Die Diagonalform entkoppelt die n Differentialgleichunge 1. Ordnung der ZRD, sodass man n voneinander unabhängige Differentialgleichungen erhält.

Die Lösung der Zustandsgleichungen (bzw. der ZRD) ist besonders einfach, wenn sich das System in Diagonalform befindet. Ausserdem ist die Stabilität des Systems sofort an der Diagonalmatrix Lambda = A_{Diag} ablesbar.

Für ein SISO-System ist die Diagonalform gegeben als:



Aus dem Diagramm ist ersichtlich, dass das System folgende UTF G(s) hat:

G(s) = Y(s)/U(s) = (b_1*c_1)/(s - lambda_1) + (b_2*c_2)/(s - lambda_2) + ... + (b_n*c_n)/(s - lambda_n)

Eigenschaften der Diagonalform:

- Wenn G(s) minimal ist, entsprechen die Diagonal-Elemente lambda_i der Systemmatrix Lambda = A_{diag} den Polen von G(s), also den Polen der n parallelen PT1-Glieder
 - 'Interne' Stabilität' kann bei nicht-minimalen Systemen von I/O-Stabilität der UTF G(s) abweichen
- G(s) ist minimal, bzw. hat Ordnung n, wenn
 - alle Produkte b_i*c_i != 0 sind und
 - alle Pole unterschiedlich sind, d.h. lambda_i != lambda_j für i != j
- Gut geeignet für Stabilitätsbetrachtung (siehe Abschnitt 2.5)

2.4 Diagonalisierung (s. 182-185)

2.4.1 Diagonalisierung mittels PBZ

Da die Diagonalmatrix Lambda = A_{Diag} die Pole beinhaltet, können diese aus der UTF mittels PBZ berechnet werden und anschliessend in eine Diagonalmatrix geschrieben werden.

Diese Methode ist nur für SISO-Systeme anwendbar!

- Wenn Pol-/Nullstellenkürzungen möglich sind resultiert ebenfalls eine Diagonalstruktur, jedoch mit reduzierter Ordnung
- Bei Mehrfachpolen resultiert die Jordanform
 - weicht geringfügig von Diagonalform ab

Beispiel: Diagonalisierung mittels PBZ

G(s) = (s+3)/((s+1)(s+2)) = 2/(s+1) + 1/(s+2) Pole in Matrix -> Lambda = A_{Diag} = [-1 0; 0 -2]

2.4.2 Diagonalisierung mittels Eigenwerten / Eigenvektoren

Eine andere Methode der Diagonalisierung benutzt die Theorie der Eigenwerte und Eigenvektoren, welche auch für MIMO-Systeme angewendet werden kann. Die Diagonalmatrix Lambda = berechnet sich als

Lambda = A_{Diag} = V^{-1} * A * V V = Matrix der Eigenvektoren

=> Entspricht Ähnlichkeitstransformation von A mit Transformationsmatrix P = V^{-1}

2.5 Stabilität

Ein LZI-System ist asymptotisch stabil, wenn alle Pole in der linken Halbebene liegen (bzw. einen negativen Realteil haben).

Unter Betrachtung der ZRD wird diese Bedingung interpretiert als: Wenn alle Eigenwerte der Systemmatrix A (welche den Eigenwerten von Lambda = A_{Diag} entsprechen) einen negativen Realteil besitzen, ist das System asymptotisch stabil.

det(lambda * I - A) != 0 -> for all lambda Re{lambda} < 0 -> System asymptotisch stabil

Achtung: Umgekehrt gilt diese Aussage nicht! Ein asymptotisch stabiles LTI-System bedeutet nicht, dass alle Eigenwerte der Systemmatrix A des Systems einen negativen Realteil besitzen. => Pol-/Nullstellenkürzungen

2.6 Steuerbarkeit (s. 185-187)

Die Steuerbarkeit beschäftigt sich mit der Frage, ob (bzw. wie) der Zustand x durch den Eingang u beeinflusst werden kann. Somit können z.B. instabile Pole weiter nach links geschoben werden, um ein instabiles System zu stabilisieren.

Definitionen von Steuerbarkeit:

Beide Definitionen von Steuerbarkeit sind äquivalent.

1. Ein System (A, B) ist steuerbar, wenn der Zustandsvektor x mit geeignetem u(t) in endlicher Zeit T_e von jedem Anfangszustand x(0) zu jedem Endzustand x(T_e) gebracht werden kann.
2. Ein System (A, B) ist steuerbar, wenn die Eigenwerte von A - BK durch geeignete Wahl von K beliebig plaziert werden können.

2.6.1 Steuerbarkeitsmatrix Q_S

Ein System ist vollständig steuerbar, wenn

- Der Rang der Steuerbarkeitsmatrix Q_S der Ordnung n des Systems entspricht
 - Die Matrix Q_S hat somit vollen Rang
- Bei Single-Input-System: (m = 1): Die Determinante von Q_S ungleich Null ist

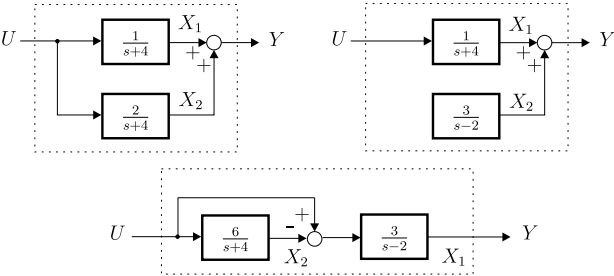
Q_S = [B AB A^2B ... A^{n-1}B] Dimension: n x (n * m)

A	Systemmatrix (n x n)	n	Zustände
B	Eingangsmatrix (n x m)	m	Eingänge

2.6.2 Nicht-steuerbare Systeme

- Gewisse Eigenwerte von A - BK sind nicht verschiebbar
- Anzahl der fixen Eigenwerte entspricht dem Defizit des Rangs von Q_S

Beispiel: Typische nicht-steuerbare Systeme



2.6.3 Regelungsnormform mittels Ähnlichkeitstransformation

Ein steuerbares SISO-System kann mit einer geeigneten Transformationsmatrix P in Regelungsnormform (siehe 2.9.1) gebracht werden.

Diese geeigneten Transformationsmatrix P wird mittels folgendem Vorgehen ermittelt:

1. Steuerbarkeitsmatrix Q_S invertieren
 - => Voraussetzung: muss invertierbar sein!
2. Von Q_S^{-1} wird die letzte Zeile als q bezeichnet
3. Transformationsmatrix P bilden als

P = [q; q * A; q * A^2; ...; q * A^{n-1}]

2.7 Beobachtbarkeit (s. 189)

Die Beobachtbarkeit stellt die Frage danach, ob (bzw. wie) der Zustand x durch die Messung des Ausgangs y bestimmt werden kann.

Definitionen von Beobachtbarkeit:

Beide Definitionen von Beobachtbarkeit sind äquivalent.

1. Ein System (A, C) ist beobachtbar, wenn der Anfangszustand x_0 innert endlicher Zeit T_e eindeutig bestimmt werden kann, sofern u und y für 0 <= t <= T_e bekannt sind.
2. Ein System (A, C) ist beobachtbar, wenn die Eigenwerte von A - HC durch geeignete Wahl von H beliebig plaziert werden können.

2.7.1 Beobachtbarkeitsmatrix Q_B

Ein System ist vollständig beobachtbar, wenn

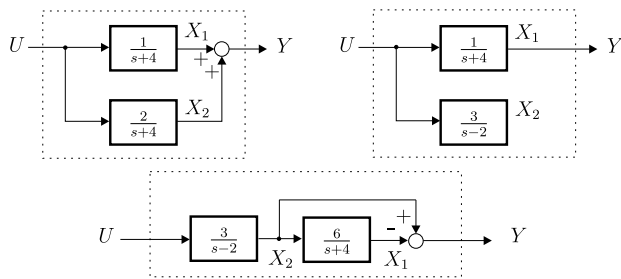
- Der Rang der Beobachtbarkeitsmatrix Q_B der Ordnung n des Systems entspricht
 - Die Matrix Q_B hat somit vollen Rang
- Bei Single-Output-System: Die Determinante von Q_B ungleich Null ist

Q_B = [C; CA; CA^2; ...; CA^{n-1}]	Dimension:	(k * n) x n
	A	Systemmatrix (n x n)
	C	Beobachtungsmatrix (k x m)
	n	Zustände
	m	Eingänge
	k	Ausgänge

2.7.2 Nicht-beobachtbare Systeme

- Gewisse Eigenwerte von A - HC sind nicht verschiebbar
- Anzahl der fixen Eigenwerte entspricht dem Defizit des Rangs von Q_B

Beispiel: Typische nicht-beobachtbare Systeme



2.8 Minimalität

Ein System ist minimal, wenn...

- Das System **steuerbar** und **beobachtbar** ist
- Keine Pol-/NS-Kürzungen in der UTF möglich sind

Zu einem nicht-minimalen System (A, B, C) kann ein System (A_M, B_M, C_M) tieferer Ordnung angegeben werden, das trotzdem **dieselbe UTF** (-Matrix) $G(s)$ hat.

2.8.1 Einfluss von Nichtminimalität

- **Bau eines neuen Systems**
 - Ersetzung durch minimale Version des Systems
⇒ Bau der **minimale Realisierung** des Systems
- **Reglerentwurf für bestehende nichtminimale Strecke**
 - mindestens ein Pol (bzw. Eigenwert) der Strecke nicht veränderbar
⇒ **gravierend**, wenn Strecke deshalb nicht stabilisiert werden kann
 - Algorithmen / Softwarepakete funktionieren oft nicht korrekt, wenn System nicht minimal

2.9 Normalformen

Kann ein SISO-System mit UTF $G(s)$ in eine der beschriebenen Normalformen gebracht werden, so hat das System gewisse Eigenschaften

$$G(s) = d + \frac{b_{n-1}s^{n-1} + \dots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0}$$

2.9.1 Regelungsnormalform (S. 188)

Kann ein System in Regelungsnormalform gebracht werden, so ist es **immer steuerbar**.

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ -a_0 & -a_1 & \dots & -a_{n-2} & -a_{n-1} \end{bmatrix} \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad c = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix}^T \quad d = d$$

2.9.2 Beobachtungsnormalform

Kann ein System in Beobachtungsnormalform gebracht werden, so ist es **immer beobachtbar**.

$$A = \begin{bmatrix} 0 & 0 & 0 & \dots & -a_0 \\ 1 & 0 & 0 & \dots & -a_1 \\ 0 & 1 & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 & -a_{n-1} \end{bmatrix} \quad b = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{bmatrix} \quad c = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}^T \quad d = d$$

Hinweis: Die Beobachtungsnormalform ist **dual** zur Regelungsnormalform.

2.10 Eigenschaften von Zustandsraummodellen (S. 190-193)

Ausgewählte wichtige Eigenschaften eines Systems (A, B, C) mit Zustandsvektor x sind hier noch einmal aufgelistet:

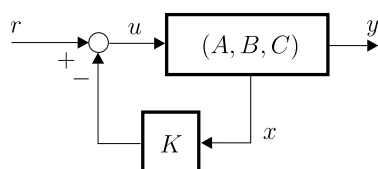
- Die Ordnung des Systems entspricht der Länge des Zustandsvektors x
- System ist asymptotisch stabil, wenn alle Eigenwerte von A in der LHE liegen bzw. alle $\lambda_i < 0$ sind
- System ist steuerbar, wenn Q_S vollen Rang hat
- System ist beobachtbar, wenn Q_B vollen Rang hat
 - Bei einem Beobachtbarkeits-Problem kann z.B. ein zusätzlicher Sensor die Strecke beobachtbar machen
- Ein System ist minimal, wenn es steuerbar **und** beobachtbar ist
- Wenn nur die UTF $G(s)$ eines Systems bekannt ist, sind **keine Aussagen über Steuerbarkeit / Beobachtbarkeit möglich**
⇒ Dafür muss innerer Aufbau des Systems (ZRD) bekannt sein!

3 Zustandsrückführungen

Bei einer Zustandsrückführung geht man davon aus, dass der Zustandsvektor x des Systems vollständig bekannt ist und zur Regelung benutzt werden kann.

Hinweis: Für die folgenden Betrachtungen des Systems gilt $D = 0$, sowie

$$K = \begin{bmatrix} k_1 & k_2 & \dots & k_n \end{bmatrix}$$



Hinweis: Die Zeichnung zeigt das Konzept. In einem Blockdiagramm hat jede Zustandsgrösse ihre eigene Rückführung (und somit ihre eigene Verstärkung k_i)

Die Systemgleichung des geschlossenen Regelkreises mit **neuer Systemmatrix** $\bar{A}$ ergibt sich zu:

$$\dot{x} = A \cdot x + B \cdot u = A \cdot x + B \cdot \underbrace{(r - Kx)}_u = \underbrace{(A - BK)}_{\bar{A}} \cdot x + B \cdot r$$

Die UTF des geschlossenen Regelkreises $G_f(s)$ berechnet sich dann als:

$$G_f(s) = \frac{Y(s)}{R(s)} = C \cdot (sI - A + BK)^{-1} \cdot B = C \cdot (sI - \bar{A})^{-1} \cdot B$$

Wenn vorausgesetzt wird, dass das System (A, B) **steuerbar** ist, können die **Pole des geschlossenen Regelkreises beliebig vorgegeben** werden.

Dies kann mittels Polvorgabe (3.2) oder LQR-Verfahren (3.3) erreicht werden.

3.1 Eigenschaften von Zustandsrückführungen

Reglenentwurf mittels Polvorgabe und LQR-Verfahren haben folgende Eigenschaften:

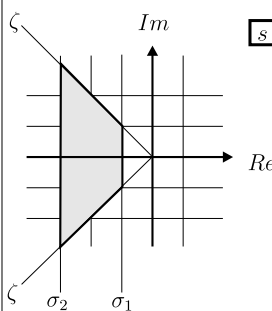
- Zustandsrückführung wirkt sich nur auf **Pole** aus, nicht auf Nullstellen
 - Nullstellen, welche durch b -Koeffizienten bestimmt sind, belieben erhalten
- Ordnung des Regelkreises bleibt erhalten (⇒ statischer Regler)
- Zustandsrückführung k entspricht grundsätzlich einem statischen (P-)Regler.
 - Bei konstanter Führungsgrösse r entsteht ein stationärer Fehler
⇒ kann durch Erweiterung mit Vorverstärkung oder I-Anteil in Regelkreis korrigiert werden (siehe Abschnitt 3.4)

3.2 Reglerentwurf mittels Polvorgabe (S. 194-196)

Bei der Zustandsrückführung können **alle n Pole beliebig vorgegeben** werden, sofern das System **steuerbar** ist.

Bei einer WOK hingegen hat man meist nur einen Freiheitsgrad (Verstärkungsfaktor des Reglers), um alle n Pole festzulegen, da die Pollagen miteinander gekoppelt sind.

3.2.1 Lage der Pole



Die Wahl der Pole des Reglers soll folgende Kriterien erfüllen:

- Pole müssen in LHE liegen ($\sigma < 0$), damit Regelkreis stabil ist
- Transienten sollen schneller abklingen als ein geeignetes $C \cdot e^{-\sigma t}$
- System soll genügend gedämpft sein, z.B. $\zeta > \frac{1}{\sqrt{2}}$
- Obere Grenze $\sigma_2 > \sigma$ verhindert zu grosse Stellsignale
- Pole, welche bereits akzeptabel liegen nicht unnötig verschieben

3.2.2 Vorgabe der Pollagen

Die Eigenwerte der Matrix $\bar{A} = (A - BK)$ sind gegeben durch die **Wurzeln** (Nullstellen) des Polynoms

$$\det(\lambda \cdot I - \bar{A}) = \det(\lambda \cdot I - A + BK) = (\lambda - \lambda_1) \cdot (\lambda - \lambda_2) \dots (\lambda - \lambda_n)$$

Das geregelte System bzw. der geschlossene Regelkreis besitzt die UTF $G_f(s)$, bei welcher der **Nenner** bzw. die Pole vorgegeben werden.

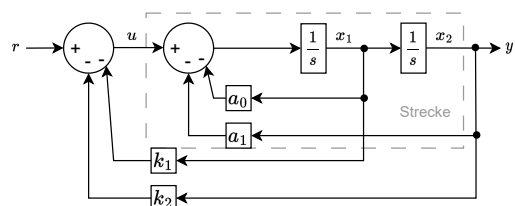
$$G_f(s) = \frac{Y(s)}{U(s)} = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \dots + b_1s + b_0}{s^n + (a_{n-1} + k_n)s^{n-1} + \dots + (a_1 + k_2)s + (a_0 + k_1)}$$

Um die Pole des geregelten Systems $G_f(s)$ an die vorgegeben Stellen $\lambda_1 \dots \lambda_n$ zu bringen, müssen folgende Polynome mittels **Koeffizientenvergleich** zum Übereinstimmen gebracht werden. (Das Polynom rechts wird sinnvollerweise ausmultipliziert.)

$$s^n + (a_{n-1} + k_n)s^{n-1} + \dots + (a_1 + k_2)s + (a_0 + k_1) \stackrel{!}{=} \det(s \cdot I - A + BK)$$

Hinweis: Die Parameter k_i sind **eindeutig** bestimmbar, sofern das System **steuerbar** ist.

Beispiel: Struktur eines Regelkreises mit Zustandsrückführung



3.3 LQR-Verfahren (Linear Quadratic Regulator) (S. 196-199)

Für die Zustandsrückführungen wird die Matrix K mit Einträgen k_i benutzt. Nun kann es sein, dass mehrere Matrizen K zur gleichen Polverteilung führen.

⇒ Das (Linear Quadratic Regulator) LQR-Verfahren ermittelt die **beste Matrix K**

Das LQR-Verfahren beruht auf der Lösung eines **Optimierungsproblems**. Aus folgenden beiden Anforderungen muss ein 'optimaler' Kompromiss gefunden werden:

- Zustände $x(t)$ sollten möglichst schnell gegen 0 konvergieren
- Stellgrösse $u(t)$ soll nicht zu gross werden (nicht zu viel Stellenergie brauchen)

4.1 Dualität: Steuerbarkeit – Beobachtbarkeit (S. 205)

Aufgrund der Dualität von Steuerbarkeit und Beobachtbarkeit sind viele Konzepte in den folgenden Abschnitten sehr ähnlich zu den Konzepten der Zustandsrückführung. Es gelten die folgenden Korrespondenzen:

Steuerbarkeit		Beobachtbarkeit
Q_s	$\leftrightarrow$	Q_B^T
B	$\leftrightarrow$	C^T
A	$\leftrightarrow$	A^T
K	$\leftrightarrow$	H^T

4.2 Entwurf eines Beobachters mittels Polvorgabe (S. 205-207)

Voraussetzung: Das System ist beobachtbar. $\Rightarrow$ Beobachtbarkeitsmatrix (2.7.1)

Die Ermittlung der Parameter h_i des Beobachters erfolgt analog (bzw. dual) zu Ermittlung der Parameter k_i der Zustandsrückführung. Typischerweise werden die **Pole des Beobachters aber etwas weiter links** platziert (bzw. vorgegeben), damit der Fehler schneller abklingt.

4.2.1 Vorgabe der Pollagen

Die Eigenwerte der Matrix $\bar{A} = (A - HC)$ sind gegeben durch die **Wurzeln** (Nullstellen) des Polynoms

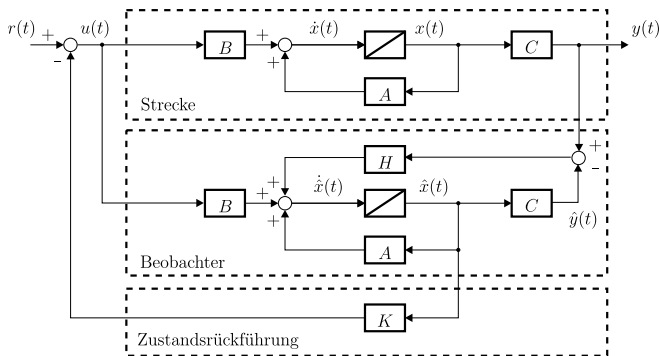
$$\det(\lambda \cdot I - \bar{A}) = \det(\lambda \cdot I - A + HC) = (\lambda - \lambda_1) \cdot (\lambda - \lambda_2) \dots (\lambda - \lambda_n)$$

Die Eigenwerte von $\bar{A} = (A - HC)$ müssen mit den **vorgegebenen Polen** übereinstimmen. Dies wird wieder durch einen **Koeffizientenvergleich** erreicht.

$$(s - p_1) \cdot (s - p_2) \cdot \dots \cdot (s - p_n) \stackrel{!}{=} \det(s \cdot I - A + HC)$$

4.2.2 Separationsprinzip

In der Regelungstechnik werden oft Systeme mit Zustandsrückführung **und** Beobachter eingesetzt.



Es kann gezeigt werden, dass beide Teilsysteme (Zustandsrückführung und Beobachter) **separat entworfen** werden können und ein **stabiles Gesamtsystem** resultiert. Die Bedingungen dafür sind:

- Zustandsrückführung stabil: $\det(\lambda_Z \cdot I - A + BK) \stackrel{!}{=} 0 \rightarrow \forall \lambda_Z \quad \text{Re}\{\lambda_Z\} < 0$
- Beobachter stabil: $\det(\lambda_B \cdot I - A + HC) \stackrel{!}{=} 0 \rightarrow \forall \lambda_B \quad \text{Re}\{\lambda_B\} < 0$

Die Eigenwerte (bzw. Pole) des Gesamtsystems entsprechen der **Vereinigung der Eigenwerte der Teilsysteme**. Die Eigenwerte (bzw. Pole) könnten auch durch Lösung einer einzigen (aufwendigen) Gleichung gelöst werden:

$$\det(\lambda \cdot I - A + BK) \cdot \det(\lambda \cdot I - A + HC) \stackrel{!}{=} 0$$

Hinweis: Gezeigt ist die 'Eigenwert-Sicht'. Wenn gezeigt werden soll, dass der Fokus auf den Polen liegt, dann wird in den Gleichungen jeweils λ durch s ersetzt.

4.3 LQG-Verfahren (Linear Quadratic Gaussian Estimation)

Die duale Methode zum LQR-Verfahren der Zustandsrückführung ist das LQG-Verfahren. Im Gegensatz zum LQR-Verfahren handelt es sich beim LQG-Verfahren um ein **stochastisches Optimierungsproblem**.

4.3.1 Optimale Beobacherverstärkung H

Die optimalen Beobacherverstärkungen (als Matrix H) ergeben sich aus der positiv definiten Lösung der Matrix S der entsprechenden Matrix-Ricatti-Gleichung:

$$H = (R^{-1}CS)^T = SC^T R^{-1}$$

Hinweis: Meist wird die Lösung H direkt mit Matlab ermittelt.

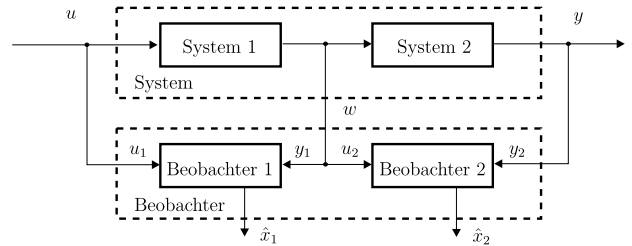
4.3.2 Eigenschaften des LQG-Verfahrens

- Die Gewichtungsmatrizen R und Q werden **positiv definit** gewählt
 - Die Gewichtungsmatrizen charakterisieren die Rauschleistung im System
- Ist R klein, so ist das Messrauschen gering bzw. die Messung zuverlässig
 - Die Verstärkung H ist dann grösser und Regelkreis somit schneller
- Die Robustheit des Gesamtsystems ist **nicht garantiert**, wenn Beobachter und Zustandsrückführung **unabhängig voneinander** entworfen werden
 - Separationsprinzip gilt nicht für Robustheit bei LQG!

4.4 Varianten von Beobachtern (S. 209-211)

4.4.1 Beobachter für Teilsysteme

Ein Beobachter kann auch den Ausgang eines Teilsystems (hier w) beobachten, statt den Ausgang des Gesamtsystems. Auch muss der Eingang eines Beobachters nicht der Stellgrösse u entsprechen, sondern kann das Ausgangssignal eines vorherigen Teilsystems sein (hier w).



Durch die Beobachtung von Teilsystemen kann die **Komplexität beim Entwurf reduziert** werden.

4.4.2 Redundante Messungen

Redundante Messungen sind sinnvoll, wenn Messrauschen vorhanden ist.

Zwei Sensoren, welche unabhängig voneinander dieselbe Grösse messen, liefern dasselbe Signal, überlagert von unterschiedlichem Rauschen. Das Signal kann mit zwei Sensoren besser geschätzt werden, wobei natürlich eine Kosten-Nutzen-Rechnung gemacht werden muss.

4.4.3 Reduzierter Beobachter

Wenn Zustandsgrössen rauschfrei gemessen werden können, ist es nicht nötig, diese zu schätzen. Es müssen dann nur $n - \mu$ Grössen geschätzt werden. Somit kann der Beobachter reduziert werden.

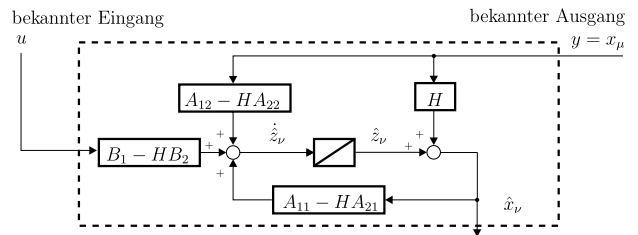
Bei folgender Darstellung der ZRD des Beobachters wird angenommen, dass die **Messgrössen einem Teil des Zustandsvektors entsprechen**. Diese müssen folglich **nicht** durch den Beobachter geschätzt werden.

$$\begin{bmatrix} \dot{x}_v \\ \dot{x}_\mu \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} x_v \\ x_\mu \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \cdot u$$

x_μ gemessene Zustandsgrössen
 x_v geschätzte Zustandsgrössen

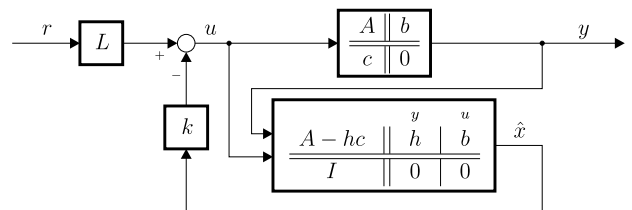
$$y = \begin{bmatrix} 0 & I \end{bmatrix} \cdot \begin{bmatrix} x_v \\ x_\mu \end{bmatrix}$$

Realisierung des reduzierten Beobachters:



4.5 Implementierung eines Zustandsreglers mit Beobachter

Für eine SISO-Strecke wurde ein Zustandsregler mit Beobachter mit Verstärkung k und eine Vorverstärkung L entworfen.



Das System kann auch so betrachtet werden, dass der Beobachter und die Vorverstärkung mit in den Regler 'gepackt' werden. Durch einsetzen von $u = Lr - k\hat{x}$ (gemäss Regelkreis oben) ergibt sich für das Gesamtsystem (unten) die folgende ZV-Darstellung:

$$\begin{aligned} \dot{\hat{x}} &= (A - hc - bk) \hat{x} + bLr + hy \\ u &= -k\hat{x} + bLr \end{aligned}$$

$$\begin{bmatrix} \dot{\hat{x}} \\ u \end{bmatrix} = \begin{bmatrix} A - hc - bk & bL & h \\ -k & L & 0 \end{bmatrix} \begin{bmatrix} \hat{x} \\ r \\ y \end{bmatrix}$$

4.5.1 Eigenschaften der Implementierung

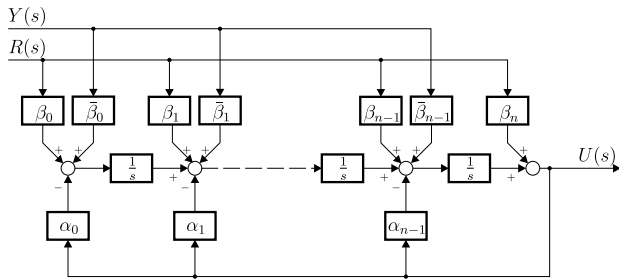
- Gesamtregler ist MIMO-System mit Eingängen r und y
- Die beiden UTF $\frac{U(s)}{R(s)}$ und $\frac{U(s)}{Y(s)}$ haben dieselbe Systemmatrix $(A - hc - bk)$
 - Beide UTF haben somit **identische Pole**
 - Nullstellen der beiden UTF sind unterschiedlich

$$\frac{U(s)}{R(s)} = \frac{\beta_n s^n \beta_{n-1} s^{n-1} + \dots + \beta_1 s + \beta_0}{s^n + \alpha_{n-1} s^{n-1} + \dots + \alpha_1 s + \alpha_0}$$

$$\frac{U(s)}{Y(s)} = \frac{\bar{\beta}_m s^m \bar{\beta}_{m-1} s^{m-1} + \dots + \bar{\beta}_1 s + \bar{\beta}_0}{s^n + \alpha_{n-1} s^{n-1} + \dots + \alpha_1 s + \alpha_0}$$

4.5.2 Reglerstruktur – Zustandsreglers mit Beobachter

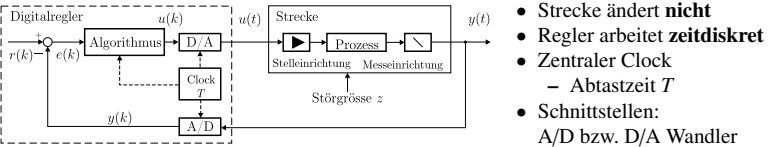
Die folgende Filterform (Direkte Form) implementiert die **minimale Realisierung** des Reglers mit Beobachter. Es sind allerdings auch andere Realisierungen möglich.



5 Digitale Regelung

5.1 Regelkreis mit digitalem Regler (S. 214)

Ein Regelkreis mit digitalem Regler hat folgende Struktur:



- Strecke ändert **nicht**
- Regler arbeitet **zeitdiskret**
- Zentraler Clock
 - Abtastzeit T
- Schnittstellen:
 - A/D bzw. D/A Wandler

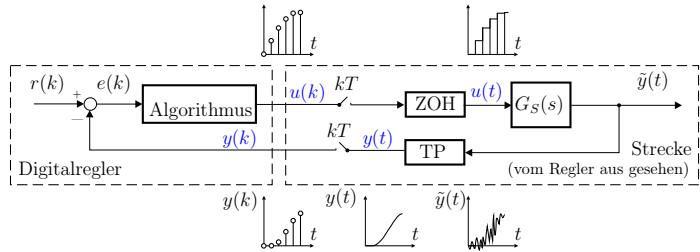
5.1.1 Vorteile / Nachteile digitale Regler (S. 215-217)

- + Struktur / Parameter des Reglers einfach veränderbar $\Rightarrow$ adaptive Regelung
- + Entwicklungsumgebungen (z.B. Simulink) erlauben Simulation des Reglers und anschließende Code-Generierung
- + Nichtlineare Regleralgorithmen sind einfach implementierbar
- + Digitale Regler können Teil eines hierarchischen Regelsystems sein
- + Einfacher, schneller Wechsel zwischen verschiedenen Reglern möglich
- + (Intelligente) Sensoren liefern Daten digital, nicht analog
 - Wertquantisierung der Signale $\Rightarrow$ Quantisierungsrauschen
 - Zeitquantisierung $\Rightarrow$ Totzeit von ca. 50 % der Abtastzeit T
 - Rundungsfehler aufgrund von rechnerischer Zahlendarstellung

5.2 Signale im digitalen Regelkreis (S. 184)

Die im Digitalregler verarbeiteten Signale $y(k)$, $r(k)$, $e(k)$ und $u(k)$ sind keine Funktionen der Zeit, sondern **Folgen von Werten!**

- Sensorseitig wird jeweils zum Zeitpunkt $t = k \cdot T$ das Signal $y(t)$ abgetastet
 - **TP**: Analoges Tiefpassfilter $\Rightarrow$ Anti-Aliasing
- Aktorseitig wird mit einem **ZOH**-Halteglied (Zero-Order-Hold) aus dem diskreten Signal $u(k)$ das Signal $u(t)$ erzeugt, welches während eines Abtastintervalls konstant ist



Damit das Zusammenspiel zwischen digitalem Regler und analoger Strecke korrekt funktioniert, muss die Schnittstelle 'gut funktionieren'. Da diese Schnittstelle aus A/D- bzw. D/A-Wandlern besteht, müssen die Effekte rund um die **Signal-Quantisierung** betrachtet werden.

5.2.1 Quantisierung der Signalwerte / Sättigung (S. 219-220)

Quantisierung und Sättigung sind **nichtlineare Effekte**. Es ist daher wichtig, die Auflösung der A/D- und D/A-Wandler geeignet zu wählen.

- Zu grobe Quantisierung der Signalwerte führt zu **Nichtlinearität**
- Zu feine Quantisierung der Signalwerte hat **keinen negativen Einfluss** auf Regler
 - Ist jedoch teuer

5.2.2 Wahl der Abtastzeit T (S. 220-221)

Im Prinzip gilt: **Je kleiner die Abtastzeit T eines Reglers ist, desto besser kann der Regler arbeiten.**

- Wird die Abtastzeit zu klein gewählt, können numerische Probleme entstehen
- Wird die Abtastzeit zu gross gewählt, kommt es zu Überschwingen oder bis zur Destabilisierung des Systems

Damit die Abtastzeit T des Reglers entsprechend gewählt werden kann, gibt es 'Faustregeln', bei welchen die Führungsgrösse, der offene Regelkreis (Regelstrecke) oder der geschlossene Regelkreis betrachtet wird.

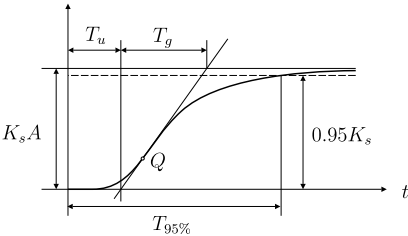
Faustregel: Abtastzeit mittels Führungsgrösse:

Das Spektrum der Führungsgrösse $r(t)$ ist begrenzt durch ω_r . Damit bei der Abtastung von $y(t)$ keine Informationen verloren gehen, wird das Nyquist-Theorem meist um **Faktor 10** übererfüllt.

$$\omega_s = \frac{2\pi}{T} \leq 10 \cdot 2 \cdot \omega_r$$

Faustregel: Abtastzeit mittels Regelstrecke:

Aus der **Schrittantwort der Regelstrecke** werden mittels **Wendetangente** die folgenden Parameter herausgelesen und damit die Abtastzeit bestimmt.



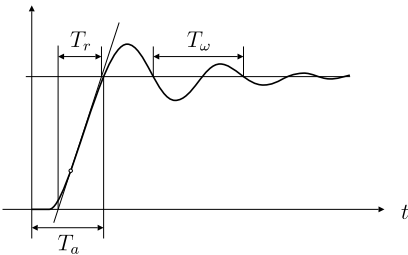
$\Rightarrow$ Wenn sich die verschiedenen **Empfehlungen widersprechen**, ist tendenziell der **kleinere Wert** als Abtastzeit T zu wählen!

Kennwert	Abtastzeit	Kommentar
T_u	$0.1 \cdot T_u \leq T \leq 0.25 \cdot T_u$	Prozess mit ausgeprägter Totzeit
T_g	$T \leq 0.1 \cdot T_g$	$T_g \gg T_u$
$T_{95\%}$	$0.05 \cdot T_{95\%} \leq T \leq 0.1 \cdot T_{95\%}$	

Faustregel: Abtastzeit mittels geschlossenem Regelkreis:

Voraussetzung: Es existiert bereits ein Regler in Analogtechnik oder in der Simulation.

Aus der **Schrittantwort des geschlossenen Regelkreises** werden die folgenden Parameter herausgelesen und damit die Abtastzeit bestimmt.



$\Rightarrow$ Wenn sich die verschiedenen **Empfehlungen widersprechen**, ist tendenziell der **kleinere Wert** als Abtastzeit T zu wählen!

Kennwert	Abtastzeit
T_a	$0.05 \cdot T_a \leq T \leq 0.1 \cdot T_a$
T_r	$0.05 \cdot T_r \leq T \leq 0.1 \cdot T_r$
T_ω	$T \leq 0.05 \cdot T_\omega$

5.2.3 Anti-Aliasing Filter (S. 222-223)

Ein Signal $y(t)$ wird periodisch mit Abtastzeit T , bzw. mit Abtastfrequenz ω_s abgetastet. Die Folge der abgetasteten Werte ist dann $y(k)$.

$$y(t) = A \cdot \cos(\omega_0 t) \xrightarrow{\text{sampling: } \omega_s} y(kT) = A \cdot \cos(\omega_0 \cdot kT) \quad \text{mit } \omega_s = 2\pi f_s = \frac{2\pi}{T}$$

Dieselbe Folge resultiert aber auch, wenn ein Signal $\tilde{y}(t) = A \cdot \cos([n \cdot \omega_s \pm \omega_0] \cdot t)$ mit ω_s abgetastet wird (n ganzzahlig).

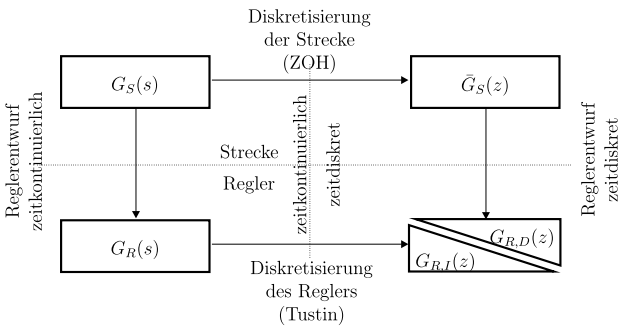
Hat solch ein Signal $\tilde{y}(t)$ Frequenzanteile, die **oberhalb der Nyquistfrequenz ω_N** liegen, so werden diese Anteile ebenfalls ins Frequenzband $\omega = [0, \omega_N]$ abgebildet.

Frequenzanteile, welche oberhalb der Nyquistfrequenz ω_N liegen, müssen deshalb mit einem **Anti-Aliasing-Filter (Tiefpass)** herausgefiltert werden. Somit ist das Nyquist-Kriterium erfüllt.

$$\omega_N = \frac{\omega_s}{2} = \frac{\pi}{T} \Leftrightarrow \omega_s > 2 \cdot \omega_{\text{Signal}}$$

5.3 Reglerentwurf – direkt vs. indirekt (S. 224-225)

- **Indirekter digitaler Reglerentwurf**
 - $G_S(s) \Rightarrow$ 'links herum' $\Rightarrow G_{R,I}(z)$
 - Zeitkontinuierlicher Regler wird zeitdiskret approximiert
- **Direkter digitaler Reglerentwurf**
 - $G_S(s) \Rightarrow$ 'rechts herum' $\Rightarrow G_{R,D}(z)$
 - 'Digitale Natur' des Reglers von Anfang an berücksichtigt



6 Indirekter digitaler Reglerentwurf

6.1 Approximationen für Diskretisierung (S. 226)

Ein kontinuierlicher Regler (hier I-Regler) weist folgendes Verhalten auf:

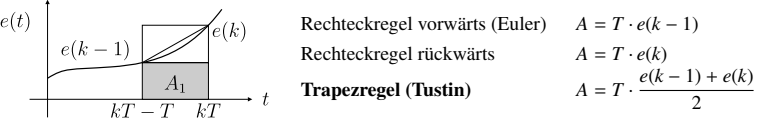
$$u(t) = K_R \cdot \int_0^t e(\tau) \, d\tau$$

Der entsprechende zeitdiskrete Regler kann **nicht exakt** gebildet werden, da $e(t)$ nur zu diskreten Zeitpunkten $e(kT)$ bekannt ist. Geht man davon aus, dass $e(t)$ **nicht start ändert** ($\Rightarrow$ geeignete Wahl der Abtastzeit T), dann kann $u(k)$ folgendermassen approximiert werden:

$$u(k) = K_R \cdot \int_0^{kT} e(\tau) \, d\tau = K_R \cdot \underbrace{\int_0^{kT-T} e(\tau) \, d\tau}_{u(k-1)} + K_R \cdot \underbrace{\int_{kT-T}^{kT} e(\tau) \, d\tau}_{\approx A}$$

6.1.1 Approximationen der Fläche A

Die Fläche A kann auf mehrere Arten approximiert werden:



Hinweis: Für die Diskretisierung von Reglern wird die Trapez-Approximation verwendet, da diese am genauesten ist.

6.2 Vorgehen: Diskretisierung eines Reglers (S. 226-228)

Die Ordnung des Reglers bleibt bei der Diskretisierung erhalten.

GR(s) = U(s)/E(s) = (bm s^m + bm-1 s^{m-1} + ... + b1 s + b0) / (sn + an-1 s^{n-1} + ... + a1 s + a0) (m ≤ n)

G_R(z) = U(z)/E(z) = (b_m z^m + b_{m-1} z^{m-1} + ... + b_1 z + b_0) / (z^n + a_{n-1} z^{n-1} + ... + a_1 z + a_0)

- 1. UTF des Reglers GR(s) bestimmen
- 2. Wahl der Abtastzeit T und Diskretisierungsmethode => typischerweise Tustin
- 3. Transformation zu GR(z) durch Substitution aller Variablen s in GR(s) mit gewünschter Approximation => siehe Abschnitt 6.2.1
- 4. Falls nötig: Doppelbrüche in GR(z) eliminieren und T numerisch einsetzen
- 5. GR(z) 'separieren' und ausmultiplizieren
- 6. Umformen in Differenzgleichung
- 7. Resultierende zeitinvariante Gleichung u(k) auflösen und bei Bedarf um n Zeitschritte verschieben

6.2.1 Substitutionsterme z-Transformation (S. 227)

Methode	Substitution von s	Substitution von z
Rechteckregel vorwärts (Euler)	$s = \frac{z-1}{T}$	$z = 1 + sT$
Rechteckregel rückwärts	$s = \frac{1}{T} \cdot \frac{z-1}{z}$	$z = \frac{1}{1-sT}$
Tustin	$s = \frac{2}{T} \cdot \frac{z-1}{z+1}$	$z = \frac{2+sT}{2-sT}$

Beispiel: Lead-Regler diskretisieren

Gegeben sei die Übertragungsfunktion GR(s) eines kontinuierlichen Reglers. Daraus soll die zu implementierende Differenzgleichung ermittelt werden.

- Diskretisierungsmethode: Tustin
 - Abtastzeit: T = 0.1 s
1. GR(s) = 4 * (s + 1) / (s + 5) => 2.
- GR(z) = 4 * (1 + z^-1) / (5 + z^-1) = 4 * (T(z + 1) + 2(z - 1)) / (5T(z + 1) + 2(z - 1)) = (3.36z - 3.04) / (z - 0.6) = U(z) / E(z)
5. z * U(z) - 0.6 * U(z) = 3.36z * E(z) - 3.04 * E(z)
6. u(k + 1) - 0.6 * u(k) = 3.36 * e(k + 1) - 3.04 * e(k) | -1 Zeitschritt
7. u(k) = 0.6 * u(k - 1) + 3.36 * e(k) - 3.04 * e(k - 1)

6.3 Empirische Einstellregeln für digitale PID-Regler

Für eine Strecke mit Totzeit GS(s) wird ein zeitkontinuierlicher PID-Regler mit UTF GR(s) entworfen. Dieser kann mit den empirischen Einstellregeln von Ziegler-Nichols parametrisiert werden.

GS(s) = KS / (sTs + 1) * e^-sTu

GR(s) = KR (1 + 1/sTN + sTV)

Wird der PID-Regler diskretisiert, so kann er nach den Regeln von Takahashi, Chan und Auslander empirisch eingestellt werden (siehe Abschnitt 6.3.1).

6.3.1 Einstellregeln nach Takahashi, Chan und Auslander (S. 230)

Voraussetzung: T ≤ 2Tu

Regler	KR	TN	TV
P-Regler	$\frac{T_g}{K_S(T_u+T)}$	-	-
PI-Regler	$0.9 \cdot \frac{T_g}{K_S(T_u+\frac{T}{2})}$	$3.33(T_u+\frac{T}{2})$	-
PID-Regler	$1.2 \cdot \frac{T_g}{K_S(T_u+T)}$	$(T_u+\frac{T}{2})$	$0.5(T_u+T)$

Hinweis: Für eine Abtastzeit von T = 0 entspricht obige Tabelle der zeitkontinuierlichen Ziegler-Nichols Approximation.

7 z-Transformation / Zeitdiskrete Übertragungsfunktion

7.1 Folgen (und deren z-Transformation) (S. 230)

Eine zeitliche Folge f(k) bricht typischerweise nicht ab und wird folgendermassen beschrieben:

f(k) = {1, a, a^2, a^3, ...} = a^k für k ≥ 0 k ganzzahlig

7.1.1 Spezielle Folgen: Abbrechende Folgen

Bei einer abbrechenden Folge liegen alle Pole bei Null. Man spricht in diesem Fall von einem FIR-Filter (Finite Response Filter)

Beispiel: Abbrechende Folge und deren z-Transformierte

f(k) = {f(0), f(1), f(2), f(3), 0, 0, 0, ...}

F(z) = f(0) + f(1)/z + f(2)/z^2 + f(3)/z^3 = (f(0)z^3 + f(1)z^2 + f(2)z + f(3))/z^3

7.2 z-Transformation – Definition (S. 232)

Die Überführung eine Folge in den Frequenzbereich erfolgt mittels z-Tranaformation und ist definiert als

F(z) = Z{f(k)} = sum_{k=0}^inf f(k) * z^-k mit der komplexen Variablen z

=> Häufig werden Tabellen und die Eigenschaften der z-Transformation verwendet => Siehe 7.3 und 7.4

7.3 z-Transformierte einiger Funktionen (S. 232)

Alle Folgen f(k) sind nur für k ≥ 0 und ganzzahlige k definiert.

f(k)		F(z)
δ(k)		1
ε(k)		z/(z-1)
k		z/(z-1)^2
k^2		z(z+1)/(z-1)^3
a^k		z/(z-a)

f(k)		F(z)
cos(bk)		z(z-cos(b))/z^2-2zcos(b)+1
sin(bk)		zsin(b)/z^2-2zcos(b)+1
k * a^k		az/(z-a)^2
k * cos(bk)		z(z-a cos(b))/(z^2-2azcos(b)+a^2)
k * sin(bk)		za sin(b)/(z^2-2azcos(b)+a^2)

Hinweis: ε(k) ist eine Folge von 1ern

Hinweis: Oft wird F(z)/z in der Tabelle gefunden

7.4 Eigenschaften der z-Transformation (S. 235)

Zeitbereich	Bildbereich	Operation im Zeitbereich
a1f1(k) + a2f2(k)	a1F1(z) + a2F2(z)	Superposition
f(k + 1)	zF(z) - z f(0)	Linksverschiebung...
f(k + 2)	z^2 F(z) - z^2 f(0) - z f(1)	...mehrfach angewendet
f(k - m) m ≥ 0	1/z^m F(z)	Rechtsverschiebung
sum_{v=0}^k f(v)	z/(z-1) F(z)	Partialsummen
sum_{v=0}^k f1(v) * f2(k - v)	F1(z) * F2(z)	Faltung

7.4.1 Grenzwertsätze der z-Transformation (S. 235)

Achtung: Grenzwertsätze dürfen nur angewendet werden, wenn Grenzwerte existieren! => In der Regelungstechnik eigentlich immer gegeben...

Zeitbereich	Bildbereich	Bezeichnung
f(0)	= lim_{z->inf} F(z)	Anfangswerttheorem
lim_{k->inf} f(k)	= lim_{z->1} [(z - 1) * F(z)]	Endwerttheorem

7.5 Rücktransformation

Die Rücktransformation kann auf mehrere Arten erreicht werden:

- 1. Direktes benutzen von Tabellen in Rückwärtsrichtung
- 2. Zerlegung von F(z) in einfachere, tabellierte Ausdrücke => Partialbruchzerlegung
- 3. Polynomdivision bzw. Reihenentwicklung von f(z) in negativen Potenzen von z
- 4. (Allgemeine Umkehroperation zur Definition der z-Transformation)

Bemerkung zu 2.: Statt F(z) zu zerlegen wird der Ausdruck F(z)/z = s1 + s2 + ... zerlegt. Anschliessend werden die Ausdrücke z * si rücktransformiert, deren Summe f(k) ergibt. Somit ist die Wahrscheinlichkeit höher, den benötigten Term in der Tabelle zu finden.

7.6 Beispiele zur z-Transformation

Beispiel: Geometrische Folge im Frequenzbereich

Die geometrische Folge ist das Pendant zur zeitkontinuierlichen Exponentialfunktion.

F(z) = sum_{k=0}^inf a^k * z^-k = sum_{k=0}^inf a^k * (a/z)^k = 1 + a/z + (a/z)^2 + (a/z)^3 + ...

Durch eine geeignete Umformung kann die Folge anders dargestellt werden und anschliessend nach F(z) aufgelöst werden.

F(z) - 1 = a/z * F(z) <=> 1 = F(z) (1 - a/z) <=> F(z) = z/(z - a)

Beispiel: Periodische Folge

f_5(k) = {0, 0, 0, 0, h(0), h(1), ..., h(m-1), h(0), h(1), ..., h(m-1), ...}

F_5(z) = z^{-m} \cdot F_5(z) + z^{-n} \cdot H(z) \Rightarrow F_5(z) = \frac{1}{z^n} \frac{z^m}{z^m - 1} H(z)

\Rightarrow H(z) entspricht der z-Transformierten von {h(0), h(1), ..., h(m-1), 0, 0, ...}

Beispiel: Ausgangssignal bestimmen

Der Ausgang entspricht der Faltung vom Eingangssignal und Impulsantwort:

y(k) = u(k) * g(k)

Bei jedem Schritt k wird die Impulsantwort mit dem aktuellen u(k) 'skaliert' um k Schritte nach rechts geschoben.

Für u(k) = {1, 0, 1, 0, 0, ...} und g(k) = {6, -2, -2, -2, 0, 0, ...} soll das Ausgangssignal y(k) berechnet werden.

u(0) \cdot g(k) = {6, -2, -2, -2, 0, 0, 0, ...}

u(1) \cdot g(k-1) = {0, 0, 0, 0, 0, 0, 0, ...}

u(2) \cdot g(k-2) = {0, 0, 6, -2, -2, -2, 0, ...}

y(k) = {6, -2, 4, -4, -2, -2, 0, ...}

Beispiel: Lösen einer linearen Differenzengleichung im Bildbereich

Wie eine Differentialgleichung mittels Laplace-Transformation wird auch eine Differenzgleichung mittels z-Transformation in drei Schritten gelöst:

- Überführung der Differenzengleichung (DG) in den Bildbereich (\Rightarrow z-Transformation)
- Lösen des resultierenden algebraischen Ausdrucks
- Rücktransformation von Y(z) in den Zeitbereich (\Rightarrow siehe Abschnitt 7.5)

Differenzengleichung: a_1 \cdot y(k+1) + a_0 \cdot y(k) = b_0 \cdot u(k)

1. a_1 \cdot z \cdot Y(z) - a_1 \cdot z \cdot y(0) + a_0 \cdot Y(z) = b_0 \cdot U(z)

2. Y(z) = \frac{b_0}{a_0 + a_1 \cdot z} \cdot U(z) + \frac{z}{a_0 + a_1 \cdot z} \cdot y(0)

3. Tabellen \rightarrow y(z)

7.7 Zeitdiskrete Übertragungsfunktion (S. 238-239)

Die zeitdiskrete Übertragungsfunktion G(z) beschreibt das Verhalten zwischen Ausgang und Eingang eines diskreten Systems und ergibt sich aus der Differenzengleichung im Zeitbereich.

G(z) = \frac{Y(z)}{U(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + ... + b_0}{z^n + a_{n-1} z^{n-1} + ... + a_0} = K \cdot \frac{(z-z_1)(z-z_2) \dots (z-z_m)}{(z-p_1)(z-p_2) \dots (z-p_n)} \quad (m \leq n)

Hinweis: Für m > n wäre G(z) akausal und somit nicht realisierbar

\Rightarrow Gerechnet werden kann mit G(z) gleich wie mit G(s)

7.7.1 DC-Gain

Der DC-Gain ist leicht anders definiert als bei der Laplace-Transformation

z-Transformation: Laplace-Transformation:

DC-Gain = G(z)|_{z=1} DC-Gain = G(s)|_{s=0}

7.8 Stabilität einer zeitdiskreten Übertragungsfunktion (S. 240-241)

Für die Untersuchung der Stabilität eines Systems wird der Nenner N(z) der Übertragungsfunktion G(z) betrachtet.

Für Grenzstabilität gilt eine UND-Verknüpfung der aufgeführten Punkte.

Für Stabilität und Instabilität gilt eine ODER-Verknüpfung der aufgeführten Punkte.

- Stabil:
 - Alle Polstellen innerhalb des Einheitskreises \Rightarrow |z| < 1
 - Keine Polstellen vorhanden
- Asymptotisch stabil:
 - Alle Polstellen innerhalb des Einheitskreises \Rightarrow |z| < 1
- Grenzstabil:
 - Keine Polstellen ausserhalb des Einheitskreises \Rightarrow |z| > 1
 - Mindestens eine einfache Polstelle auf dem Einheitskreis \Rightarrow |z| = 1
 - Keine doppelten Polstellen auf dem Einheitskreis \Rightarrow |z| = 1
- Instabil:
 - Mindestens eine Polstelle ausserhalb des Einheitskreises \Rightarrow |z| > 1
 - Mindestens eine mehrfache Polstelle auf dem Einheitskreis \Rightarrow |z| = 1

Hinweis: 'Stabil' ist ein Synonym für 'asymptotisch stabil'

Hinweis: Tustin erhält die Stabilitätseigenschaften eines Systems

7.8.1 Jury-Kriterium (S. 241)

Als Pendant zum Routh-Kriterium im Laplace-Bereich existiert in der z-Ebene das Jury-Kriterium. Somit kann vereinfacht festgestellt werden, ob sich alle Pole des Polynoms N(z) innerhalb des Einheitskreises befinden und das System somit stabil ist.

Systemordnung	Stabilitätsbedingungen für N(z) = z^n + a_{n-1} z^{n-1} + ... + a_0
n = 1	a_0 < 1
n = 2	a_0 + a_1 + 1 > 0 a_0 - a_1 + 1 > 0 a_0 < 1
n = 3	a_0 + a_1 + a_2 + 1 > 0 -a_0 + a_1 - a_2 + 1 > 0 a_0 < 1 a_0^2 - a_0 a_2 + a_1 < 1

7.9 Frequenzgang / Bodediagramm / Nyquistdiagramm (S. 241-243)

Wie bei zeitkontinuierlichen Systemen ergibt sich der Frequenzgang, wenn die UTF G(z) entlang der Stabilitätsgrenze ausgewertet wird. Die Stabilitätsgrenze ist der Einheitskreis, also alle Stellen z = e^{j\omega T}

G(z)|_{z=e^{j\omega T}} = K \cdot \frac{(z-z_1)(z-z_2) \dots (z-z_m)}{(z-p_1)(z-p_2) \dots (z-p_n)} \Big|_{z=e^{j\omega T}}

Bodediagramm und Nyquistdiagramm können dann ähnlich gezeichnet werden wie bei zeitkontinuierlichen Systemen. Es gelten allerdings folgende Besonderheiten:

- Darstellungsbereich der Diagramme eingeschränkt auf den Bereich \omega \in [0, \omega_N]
 - Entspricht Frequenzgang entlang oberer Hälfte des Einheitskreises
 - Untere Hälfte ist redundant
- Abtastzeit passend wählen
 - Vor allem zu grosse Abtastzeiten führen zu grossen Abweichungen
- Im Bodediagramm existieren keine Geradenapproximationen
- Die Beuteilung der Stabilität wird am offenen Regelkreis durchgeführt

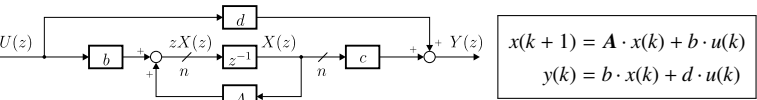
7.9.1 Darstellungsbereich

\omega_N = \frac{\omega_s}{2} \quad \omega_N \quad \text{Nyquistfrequenz} \quad | \quad \omega_s \quad \text{Samplingfrequenz}

8 Zeitdiskrete Zustandsraumdarstellung

8.1 Zeitdiskrete ZRD (S. 243-244)

Die Struktur der diskreten ZRD ist identisch wie die Struktur bei zeitkontinuierlichen Systemen. Auch im diskreten Bereich kann eine UTF nicht eindeutig in eine ZRD übersetzt werden (Diagonalform, Beobachtungsnormalform, etc.).



x(k+1) = A \cdot x(k) + b \cdot u(k)

y(k) = b \cdot x(k) + d \cdot u(k)

8.1.1 UTF aus diskreter ZRD

Die diskrete UTF G(z) kann jedoch eindeutig aus der ZRD bestimmt werden

G(z) = c \cdot (zI - A)^{-1} \cdot b + d

8.2 Stabilität (S. 245)

Das zeitdiskrete System ist asymptotisch stabil, wenn alle Pole innerhalb des Einheitskreises liegen (bzw. |z| < 1 ist).

Unter Betrachtung der ZRD wird diese Bedingung interpretiert als: Wenn alle Eigenwerte der Systemmatrix A (welche den Eigenwerten von \Lambda = A_{\text{Diag}} entsprechen) einen Betrag < 1 besitzen, ist das System asymptotisch stabil.

\det(\lambda \cdot I - A) \stackrel{!}{=} 0 \quad \rightarrow \quad \forall \lambda \quad |\lambda| < 1 \quad \Rightarrow \text{System asymptotisch stabil}

Achtung: Umgekehrt gilt diese Aussage nicht! Ein asymptotisch stabiles System bedeutet nicht, dass alle Eigenwerte der Systemmatrix A des Systems einen Betrag < 1 haben \Rightarrow Pol-/Nullstellenkürzungen

8.3 Steuerbarkeit (S. 245)

- Es gelten die (fast) äquivalenten Definitionen wie in zeitkontinuierlichen Systemen
- Die Steuerbarkeit des Systems wird mit der Steuerbarkeitsmatrix bestimmt

8.3.1 Steuerbarkeitsmatrix Q_S

Ein System ist vollständig steuerbar, wenn

- Der Rang der Steuerbarkeitsmatrix Q_S der Ordnung n des Systems entspricht
 - Die Matrix Q_S hat somit vollen Rang
- Bei Single-Input-System: (m = 1): Die Determinante von Q_S ungleich Null ist

Q_S = \begin{bmatrix} B & AB & A^2 B & \dots & A^{n-1} B \end{bmatrix} \quad \text{Dimension: } n \times (n \cdot m)

A	Systemmatrix (n \times n)	n	Zustände
B	Eingangsmatrix (n \times m)	m	Eingänge

8.4 Beobachtbarkeit (S. 245)

- Es gelten die (fast) äquivalenten Definitionen wie in zeitkontinuierlichen Systemen
- Die Beobachtbarkeit des Systems wird mit der Beobachtbarkeitsmatrix bestimmt

8.4.1 Beobachtbarkeitsmatrix Q_B

Ein System ist vollständig beobachtbar, wenn

- Der Rang der Beobachtbarkeitsmatrix Q_B der Ordnung n des Systems entspricht
 - Die Matrix Q_B hat somit vollen Rang
- Bei Single-Output-System: Die Determinante von Q_B ungleich Null ist

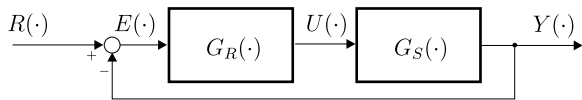
Q_B = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad \begin{array}{ll} \text{Dimension:} & (k \cdot n) \times n \\ A & \text{Systemmatrix } (n \times n) \\ C & \text{Beobachtungsmatrix } (k \times m) \\ n & \text{Zustände} \\ m & \text{Eingänge} \\ k & \text{Ausgänge} \end{array}

9 Direkter digitaler Reglerentwurf

Beim direkten Reglerentwurf wird die Strecke $G_S(s)$ in zeitdiskrete Form $\bar{G}_S(z)$ gebracht. Anschliessend wird für $\bar{G}_S(z)$ ein (zeitdiskreter) Regler $G_{R,D}(z)$ entworfen.

9.1 Direkter digitaler Reglerentwurf – Vorgehen

Das Vorgehen und die Berechnungen sind grundsätzlich gleich wie im zeitkontinuierlichen Bereich, weshalb in folgender Abbildung der Platzhalter $\cdot$ für s oder z verwendet ist. Es müssen jedoch einige Voraussetzungen erfüllt sein.



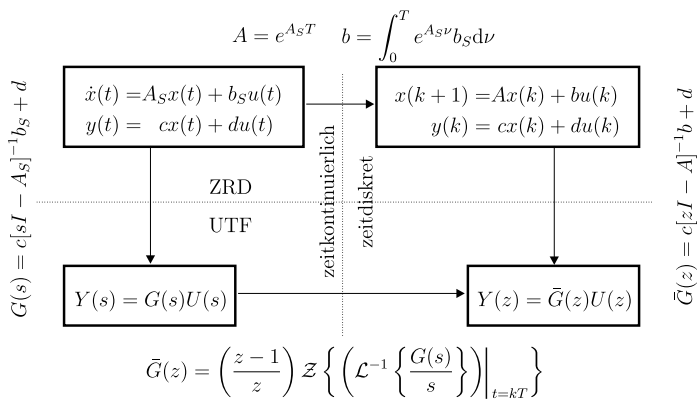
$$\text{Geschlossener Regelkreis} \quad G_f(\cdot) = \frac{Y(\cdot)}{R(\cdot)} = \frac{G_S(\cdot)G_R(\cdot)}{1 + G_S(\cdot)G_R(\cdot)}$$

9.1.1 Voraussetzungen

- Alle involvierten Teilsysteme müssen **'kompatibel'** sein
 - Entweder alles in z formuliert oder alles in s formuliert
- Zeitdiskrete Systeme müssen **dieselbe Abtastzeit** haben

9.2 ZOH-Diskretisierung – Überblick

Eine ZOH-Diskretisierung kann an einer UTF oder im Zustandsraum durchgeführt werden:



Beide Diskretisierungs-Arten werden in den Abschnitten 9.3 und 9.4 genauer beschrieben.

9.2.1 Steuerbarkeit / Beobachtbarkeit (s. 251-253)

Für Steuerbarkeit und Beobachtbarkeit bei der ZOH-Diskretisierung gilt:

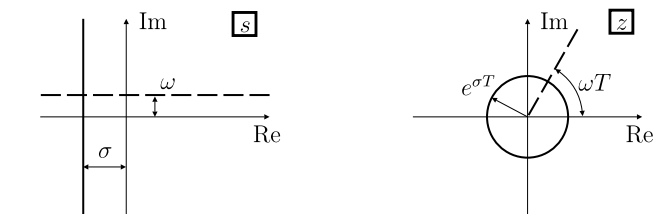
- Solange die Abtastzeit T **sinnvoll gewählt** wird, bleiben diese Eigenschaften **erhalten**
- Ein System kann durch Diskretisierung **nicht** steuerbar / beobachtbar **gemacht werden**

9.2.2 Stabilität

Für die Stabilität gilt bei der ZOH-Diskretisierung:

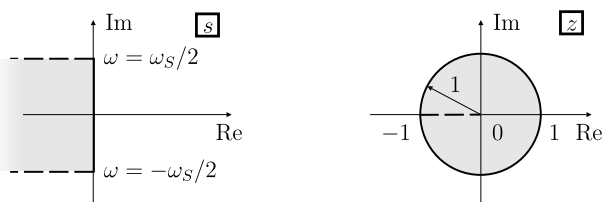
- Stabilität **bleibt** bei ZOH-Diskretisierung **erhalten**
 - Pole bzw. Eigenwerte gehorchen bei ZOH-Diskretisierung der Abbildung $z = e^{sT}$

Abbildung $z = e^{sT}$:



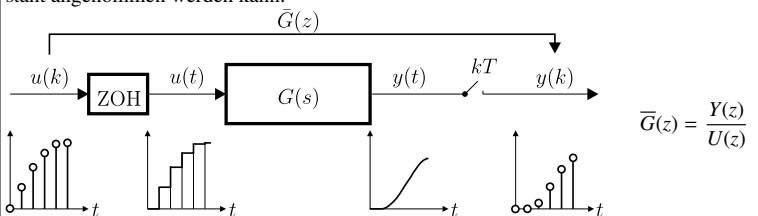
s-Ebene	↔	z-Ebene
Gerade mit konstantem Imaginärteil	↔	Halbgerade durch Nullpunkt
Geraden mit konstantem Realteil	↔	Kreise um den Nullpunkt
Imaginärachse	↔	Einheitskreis
Linke Halbebene	↔	Innerhalb des Einheitskreises
Nullpunkt (Ursprung)	↔	Punkt (1 0)

Beispiel: Abbildung eines Streifens



9.3 ZOH-Diskretisierung mit UTF (s. 248-249)

Eine zeitkontinuierliche Strecke $G_S(s) = G(s)$ wird mit einem ZOH-Glied (zero-order-hold) **exakt** diskretisiert, weil der Eingang zur zeitdiskreten Strecke als stückweise konstant angenommen werden kann.



Im Gegensatz zur ZOH-Diskretisierung ist die Approximation der Strecke mittels z.B. Tustin beim indirekten Reglerentwurf nicht exakt!

9.3.1 Bestimmung der diskreten Stecke $\bar{G}_S(z)$

Die Bestimmung von $\bar{G}_S(z)$ wird vereinfacht durch Betrachtung der **Impulsantwort**.

Mit $u(k) = \{1, 0, 0, 0, \dots\}$ ergibt sich am Ausgang die Impulsantwort $y(k) = g(k)$, deren z-Transformierte $G(z) (= \bar{G}_S(z))$ ist.

Als Formel kann die diskrete UTF $\bar{G}_S(z)$ ermittelt werden als:

$$\bar{G}_S(z) = \left(\frac{z-1}{z} \right) \mathcal{Z} \left\{ \left(\mathcal{L}^{-1} \left\{ \frac{G_S(s)}{s} \right\} \right) \right\}_{t=kT} = (1-z^{-1}) \mathcal{Z} \left\{ \left(\mathcal{L}^{-1} \left\{ \frac{G_S(s)}{s} \right\} \right) \right\}_{t=kT}$$

Beispiel: ZOH-Diskretisierung mit UTF

Eine gegebene Strecke $G_S(s) = \frac{1}{s^2}$ soll mittels ZOH diskretisiert werden. Es soll also $\bar{G}_S(z)$ bestimmt werden, indem obige Formel Schritt für Schritt angewendet wird.

$$\frac{1}{s^2} \xrightarrow{\div s} \frac{1}{s^3} \xrightarrow{\mathcal{L}^{-1}} \frac{t^2}{2} \xrightarrow{t=kT} \frac{k^2 T^2}{2} \xrightarrow{\mathcal{Z}} \frac{T^2}{2} \frac{z(z+1)}{(z-1)^3} \cdot \left(\frac{z-1}{z} \right) = \frac{T^2}{2} \frac{(z+1)}{(z-1)^2} = \bar{G}_S(z)$$

9.4 ZOH-Diskretisierung im Zustandsraum (s. 250-251)

Eine zeitkontinuierliche ZRD kann in eine zeitdiskrete ZRD umgewandelt werden (entspricht dem Weg 'rechts herum' in Abschnitt 9.2). Die Variablen des Zustandsvektors bleiben bei der Diskretisierung erhalten, werden jedoch nur noch zum Abtastzeitpunkt evaluiert.

Die diskrete ZRD kann dargestellt werden als:

$$x(k+1) = A \cdot x(k) + b \cdot u(k) \quad A = e^{A_S T} \quad b = \int_0^T e^{A_S \nu} d\nu \cdot b_S$$

$$y(k) = c \cdot x(k) + d \cdot u(k)$$

Hinweis: A_S und b_S entsprechen den zeitkontinuierlichen ZRD-Werten

9.4.1 Bestimmung der diskreten Stecke $\bar{G}_S(z)$

Die diskrete UTF der Strecke $\bar{G}_S(z)$ kann aus der diskreten ZRD ähnlich wie im zeitkontinuierlichen Bereich bestimmt werden:

$$\bar{G}_S(z) = c \cdot (zI - A)^{-1} \cdot b + d$$

9.5 Reglerentwurf – WOK (s. 254)

Basierend auf den Pole / Nullstellen des offenen Regelkreises werden die Pole / Nullstellen des geschlossenen Regelkreises (bzw. die Pole des Reglers) platziert. Da das Zeichnen der WOK (des **geschlossenen** Regelkreises) analog zum zeitkontinuierlichen Bereich erfolgt, wird dies hier nicht weiter erläutert ($\Rightarrow$ siehe Anhang Abschnitt 11).

Die (zeitdiskrete) UTF des geschlossenen Regelkreises mit einem P-Regler mit Verstärkung K ergibt sich als

$$G_f(z) = \frac{K \cdot G_S(z)}{1 + K \cdot G_S(z)} = \frac{K \cdot \frac{Z_S(z)}{N_S(z)}}{1 + K \cdot \frac{Z_S(z)}{N_S(z)}} = \frac{K \cdot Z_S(z)}{N_S(z) + K \cdot Z_S(z)} = \frac{Z_f(z)}{N_f(z)}$$

char. Polynom $N_f(z)$

Wird ein dynamischer Regler verwendet, so wird $G_0(z)$ in die **mathematisch verlangte Form** von aufgeteilt.

$$G_f(z) = \frac{G_R(z) \cdot G_S(z)}{1 + G_R(z) \cdot G_S(z)} = \frac{\frac{Z_R(z)}{N_R(z)} \cdot \frac{Z_S(z)}{N_S(z)}}{1 + \frac{Z_R(z)}{N_R(z)} \cdot \frac{Z_S(z)}{N_S(z)}} = \frac{Z_R(z) \cdot Z_S(z)}{N_R(z) \cdot N_S(z) + Z_R(z) \cdot Z_S(z)} = \frac{Z_f(z)}{N_f(z)}$$

char. Polynom $N_f(z)$

Hinweis: Die Methode zur **Wurzel-Bestimmung** funktioniert für **jedes Polynom** $N_f(s)$ (oder auch $N_f(z)$), wenn dieses in die verlangte Form gebracht werden kann (und auch nur dann)!

$$N_f(s) = P_N(s) + \text{Faktor} \cdot P_Z(s) = N(s) + K \cdot Z(s)$$

Hinweis: Wie eine WOK (in s) gezeichnet wird kann dem Anhang (Abschnitt 11.2) entnommen werden.

9.5.1 Platzierung der Pole

- Pole **innerhalb des Einheitskreises** platzieren ($|z| < 1$)
- Gebiet, welches Spezifikation erfüllt in z -Ebene transformieren und Pole innerhalb des transformierten Gebiets platzieren
- Es können Iteratoren (mit Polen bei $z = 1$) in den Regelkreis eingebaut werden
 - Somit wird der stationäre Fehler unterdrückt

9.6 Reglerentwurf – Vorgabe des geschl. Regelkreises (S. 256-257)

Bei bekannter **diskreten** Regelstrecke $G_S(z)$ kann die benötigte UTF des Reglers $G_R(z)$ ermittelt werden, wenn die UTF des geschlossenen Regelkreises $G_f(z)$ vorgegeben ist.

$$G_f(z) = \frac{G_0(z)}{1 + G_0(z)} \iff G_0(z) = \frac{G_f(z)}{1 - G_f(z)} \quad \text{und} \quad G_R(z) = \frac{G_0(z)}{G_S(z)}$$

Daraus folgt für die UTF $G_R(z)$ des diskreten Reglers:

$$G_R(z) = \frac{Z_f(z)}{N_f(z) - Z_f(z)} \cdot \frac{N_S(z)}{Z_S(z)} \quad \text{mit} \quad G_S(z) = \frac{Z_S(z)}{N_S(z)} \quad \text{und} \quad G_f(z) = \frac{Z_f(z)}{N_f(z)}$$

Bemerkungen:

- Die Strecke $G_S(z)$ darf **keine** Pole ausserhalb des Einheitskreises haben
- Die Strecke darf keine NS ausserhalb des Einheitskreises haben, ausser wenn diese in $G_f(z)$ übernommen werden
- Die Strecke darf **Totzeiten** (ganzzahlige Vielfache der Abtastzeit) haben: $T_f = N \cdot T$
 - Strecke erhält dadurch N zusätzliche Pole im Ursprung: $G_S(z) = \dots \cdot z^{-N}$
- Der relative Grad (**Polüberschuss**) der vorgegebenen UTF $G_f(z)$ muss mindestens so gross sein wie derjenige der Strecke $G_S(z)$
 - Regler ist sonst nicht realisierbar

$$\underbrace{\text{Ord}\{N_S(z)\} - \text{Ord}\{Z_S(z)\}}_{\text{Polüberschuss Strecke}} \leq \underbrace{\text{Ord}\{N_f(z)\} - \text{Ord}\{Z_f(z)\}}_{\text{Polüberschuss geschl. System}}$$

9.7 Reglerentwurf – Deadbeat-Regler (S. 256-257)

Ein Deadbeat-Regler wird bei einem Führungsschritt der **Fehler in endlich vielen Abtastschritten Exakt zu Null**. Der geschlossene Regelkreis $G_f(z)$ besetzt somit folgende Eigenschaften:

- Alle Pole bei $z = 0$
- Statische Verstärkung $G_f(1) = 1$

9.7.1 Einfachster Deadbeat-Regler

Damit der Fehler zu Null, wird ist integrierendes Verhalten nötig. Der einfachste mögliche Regler $G_R(z)$ und der geschlossene Regelkreis $G_f(z)$ besitzen also die folgenden UTFs:

$$G_R(z) = \frac{1}{z-1} \cdot \frac{1}{G_S(z)} \quad \Rightarrow \quad G_f(z) = \frac{1}{z}$$

Natürlich muss der Regler realisierbar sein (stabil, relativer Grad von G_S)

9.8 Reglerentwurf – Zustandsraum (S. 257-258)

Beim Reglerentwurf im Zustandsraum sind bei zeitdiskreten Systemen dieselben drei Komponenten von Interesse wie in zeitkontinuierlichen Systemen:

- Beobachter**, der den Zustandsvektor schätzt
- Zustandsrückführung**, die z.B. die Strecke stabilisiert
- Vorverstärkung**, für die statische Verstärkung $\Rightarrow G_f(1) = 1$

Da die Konzepte weitgehend vom zeitkontinuierlichen Bereich übernommen werden können, werden nur ausgewählte Aspekte erwähnt.

9.8.1 Polplatzierung

- Die Eigenwerte von $A - BK$ (und somit die Pole des Systems) ($\Rightarrow$ Zustandsrückführung) bzw. $A - HC$ ($\Rightarrow$ Beobachter) müssen nun innerhalb des Einheitskreises platziert werden
- Es können dieselben Matlab-Funktionen verwendet werden wie bei einem zeitkontinuierlichen System
- Separationsprinzip bleibt erhalten

9.8.2 LQR / LQG

- Es wird eine andere Zielfunktion J optimiert, was zu einer anderen Ricatti-Gleichung führt
 - Mathematischer Lösungsweg ist anders
- Matlab benötigt einen anderen Befehl: `dlqr(...)` statt `lqr(...)`

II Anhang

10 Mathematik-Hilfen

10.1 Trigonometrie

α	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$	$\frac{2\pi}{3}$	$\frac{3\pi}{4}$	$\frac{5\pi}{6}$	π	$\frac{7\pi}{6}$	$\frac{5\pi}{4}$	$\frac{4\pi}{3}$	$\frac{3\pi}{2}$	$\frac{5\pi}{3}$	$\frac{7\pi}{4}$	$\frac{11\pi}{6}$	2π
α°	0°	30°	45°	60°	90°	120°	135°	150°	180°	210°	225°	240°	270°	300°	315°	330°	360°
$\sin(\alpha)$	0	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	$-\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{3}}{2}$	-1	$-\frac{\sqrt{3}}{2}$	$-\frac{\sqrt{2}}{2}$	$-\frac{1}{2}$	0
$\cos(\alpha)$	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$	$-\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{3}}{2}$	-1	$-\frac{\sqrt{3}}{2}$	$-\frac{\sqrt{2}}{2}$	$-\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{3}}{2}$	1
$\tan(\alpha)$	0	$\frac{\sqrt{3}}{3}$	1	$\sqrt{3}$	$\pm\infty$	$-\sqrt{3}$	-1	$-\frac{\sqrt{3}}{3}$	0	$\frac{\sqrt{3}}{3}$	1	$\sqrt{3}$	$\pm\infty$	$-\sqrt{3}$	-1	$-\frac{\sqrt{3}}{3}$	0
$\cot(\alpha)$	$\pm\infty$	$\sqrt{3}$	1	$\frac{\sqrt{3}}{3}$	0	$-\frac{\sqrt{3}}{3}$	-1	$-\sqrt{3}$	$\pm\infty$	$\sqrt{3}$	1	$\frac{\sqrt{3}}{3}$	0	$-\frac{\sqrt{3}}{3}$	-1	$-\sqrt{3}$	$\pm\infty$

10.2 Lineare Algebra – Repetition

Der Anhang B des Skripts (S. 279 ff.) beschreibt die Matrizenrechnung. Hier werden nur einige Ausschnitte erwähnt.

10.2.1 Matrixmultiplikation

Die Spaltenzahl von A muss mit der Zeilenzahl von B übereinstimmen!

Dimensionen: $A \in \mathbb{C}^{m \times n}$, $B \in \mathbb{C}^{n \times p} \Rightarrow C \in \mathbb{C}^{m \times p}$

Beispiel: Matrixmultiplikation

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 0 & 4 & -1 \end{pmatrix}, \quad B = \begin{pmatrix} -2 & 4 \\ -4 & -5 \\ 2 & 4 \end{pmatrix}$$

$$C = A \cdot B = \begin{pmatrix} 1 \cdot (-2) + 2 \cdot (-4) + 4 \cdot 2 & 1 \cdot 4 + 2 \cdot (-5) + 4 \cdot 4 \\ 0 \cdot (-2) + 4 \cdot (-4) + (-1) \cdot 2 & 0 \cdot 4 + 4 \cdot (-5) + (-1) \cdot 4 \end{pmatrix} = \begin{pmatrix} -2 & 10 \\ -18 & -24 \end{pmatrix}$$

10.2.2 Übersicht zu (quadratischen) Matrizen

Die Aussagen innerhalb einer Spalte sind jeweils äquivalent.

$A \in \mathbb{C}^{n \times n}$ ist regulär	$A \in \mathbb{C}^{n \times n}$ ist singulär
$\det(A) \neq 0$	$\det(A) = 0$
Die Inverse A^{-1} existiert	Die Inverse A^{-1} existiert nicht
n Spaltenvektoren von A sind linear unabhängig	n Spaltenvektoren von A sind nicht linear unabhängig
n Zeilenvektoren von A sind linear unabhängig	n Zeilenvektoren von A sind nicht linear unabhängig
A hat vollen Rang: $\text{rang}(A) = n$	A hat nicht vollen Rang: $\text{rang}(A) < n$

⇒ Die oberen drei Zeilen gelten **nur für quadratische Matrizen**

10.2.3 Determinante

Die Determinante ist nur für **quadratische** Matrizen definiert

- $A = [a_{11}] \Rightarrow \det(A) = a_{11}$
- $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \Rightarrow \det(A) = a_{11}a_{22} - a_{12}a_{21}$
- $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Rightarrow \det(A) = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33} - a_{13}a_{22}a_{31}$
- Bei einer Diagonalmatrix oder einer Dreiecksmatrix entspricht die Determinante dem Produkt der Diagonalelemente

10.2.4 Singuläre / Reguläre Matrizen

- Eine Matrix $A \in \mathbb{C}^{n \times n}$ ist singulär, falls $\det(A) = 0$
- Eine Matrix $A \in \mathbb{C}^{n \times n}$ ist regulär, falls $\det(A) \neq 0$

10.2.5 Rang einer Matrix

- Der Rang einer Matrix $A \in \mathbb{C}^{m \times n}$ bestimmt sich als Anzahl linear unabhängiger Spaltenvektoren (was zugleich der Anzahl linear unabhängiger Zeilenvektoren ist)
- Der Maximal mögliche Rang ist $\min(m, n)$

10.2.6 Inverse Matrizen

Eine Matrix ist invertierbar, wenn sie regulär ist.

$$A = [a_{11}] \Rightarrow A^{-1} = \frac{1}{a_{11}} \quad A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \Rightarrow A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix}$$

- Für $n \geq 3$ empfiehlt sich der Gauss-Jordan Algorithmus
- Die Inverse von Diagonalmatrizen kann durch Inversion der Diagonalelemente berechnet werden

10.2.7 Lineare Gleichungssysteme lösen

$$A \cdot \vec{x} = \vec{b} \Leftrightarrow \vec{x} = A^{-1} \cdot \vec{b}$$

10.2.8 Transformationen

Die Transformationsmatrix T kann eine Drehung, Rotation, Projektion oder Permutation von $\vec{p}$ verursachen.

Permutation: Reihenfolge ändern (⇒ Permutationsmatrizen haben nur eine 1 pro Zeile und Spalte und sind orthogonal)

$$\vec{q} = T \cdot \vec{p}$$

10.2.9 Eigenwerte / Eigenvektoren

Um die Eigenwerte λ_i einer (quadratischen) Matrix A zu berechnen, muss die folgende Gleichung erfüllt sein:

$$\det(\lambda \cdot I - A) \stackrel{!}{=} 0$$

Die Lösungen dieser charakteristischen Gleichung entsprechen den Eigenwerten λ_i .

Die Eigenvektoren berechnen sich durch Einsetzen von λ_i in die Gleichung(en)

$$(\lambda_i I - A) \cdot \vec{v}_i = \vec{0}$$

⇒ Eine Koordinate muss gewählt werden, die andere Koordinate ist linear abhängig!

Eigenschaften von Eigenwerten und Eigenvektoren:

- Bei einer Diagonalmatrix / Dreiecksmatrix entsprechen die Diagonalelemente a_{ii} den Eigenwerten
- Eine singuläre Matrix hat mindestens einen Eigenwert bei Null
- Zwei Eigenwerte $\lambda_i \neq \lambda_j$ haben unabhängig Eigenvektoren $\vec{v}_i$ und $\vec{v}_j$
- Die Eigenwerte von A und A^T sind identisch
- Reelle Eigenwerte haben reelle Eigenvektoren

10.2.10 Positive Definitheit von Matrizen

Die Definitheit einer Matrix kann auf mehrere Arten überprüft werden. Eine Matrix Q ist positiv definit ($Q > 0$), wenn

- Alle** Eigenwerte der Matrix grösser Null sind ⇒ $\lambda_i > 0$
- Alle** führenden Hauptminoren positiv sind
 - Alle Determinanten der gezeigten 'Untermatrizen' müssen > 0 sein

$$Q = \begin{bmatrix} \boxed{q_{11}} & q_{12} & \cdots & q_{1n} \\ q_{12} & \boxed{q_{22}} & \cdots & q_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ q_{1n} & q_{2n} & \cdots & \boxed{q_{nn}} \end{bmatrix}$$

10.2.11 Exponentialfunktion von Matrizen

Die Exponentialfunktion von Matrizen berechnet sich aus der Definition der Taylor-Reihe.

$$e^{At} = I + At + \frac{A^2}{2!}t^2 + \cdots + \frac{A^k}{k!}t^k + \cdots = \sum_{k=0}^{\infty} \frac{A^k t^k}{k!}$$

Spezialfall: Diagonalmatrix:

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix} \Rightarrow e^{At} = \begin{bmatrix} e^{a_{11} \cdot t} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & e^{a_{nn} \cdot t} \end{bmatrix}$$

10.3 Dezibel dB

$$|K|_{\text{dB}} = 20 \text{ dB} \cdot \log_{10} |K| \Leftrightarrow |K| = 10^{\left(\frac{|K|_{\text{dB}}}{20 \text{ dB}}\right)}$$

Hinweis: Die Betragsstriche nur Notation! $|K|$ kann sehr wohl negativ sein!

10.3.1 Rechenregeln

- Multiplikation ⇒ Addition

$$|K_1 \cdot K_2|_{\text{dB}} = |K_1|_{\text{dB}} + |K_2|_{\text{dB}}$$

- Division ⇒ Subtraktion

$$\left| \frac{K_1}{K_2} \right|_{\text{dB}} = |K_1|_{\text{dB}} - |K_2|_{\text{dB}}$$

- Kehrwert ⇒ Negatives Vorzeichen

$$\left| \frac{1}{K_1} \right|_{\text{dB}} = |1|_{\text{dB}} - |K_1|_{\text{dB}} = -|K_1|_{\text{dB}}$$

10.3.2 dB–Umrechnungstabelle (S. 133)

Faktor [1]	Dezibel dB	Faktor [1]	Dezibel dB
100	40	2	6
10	20	$\sqrt{2}$	3
1	0	$\frac{1}{\sqrt{2}}$	-3
0.1	-20	$\frac{1}{2}$	-6
0.01	-40		

10.4 Spezielle Ansätze PBZ

$$f(x) = \frac{5x^2 - 37x + 54}{x^3 - 6x^2 + 9x} = \frac{A}{x} + \frac{B}{x-3} + \frac{C}{(x-3)^2} = \frac{A(x-3)^2 + Bx(x-3) + Cx}{x(x-3)^2}$$

$$f(x) = \frac{1.5x}{x^3 - 6x^2 + 12x - 8} = \frac{A}{x-2} + \frac{B}{(x-2)^2} + \frac{C}{(x-2)^3} = \frac{A(x-2)^2 + B(x-2) + C}{(x-2)^3}$$

$$f(x) = \frac{x^2 - 1}{x^3 + 2x^2 - 2x - 12} = \frac{A}{x-2} + \frac{Bx+C}{x^2+4x+6} = \frac{A(x^2+4x+6) + (Bx+C)(x-2)}{(x-2)(x^2+4x+6)}$$

10.5 Laplace-Transformation

10.5.1 Laplace-Transformierte einiger Funktionen (S. 21)

Alle Funktionen $f(t)$ sind implizit mit dem Einheitssprung $\varepsilon(t)$ multipliziert, sodass die Funktionen nur für $t \geq 0$ definiert sind.

$f(t)$	$\circ \bullet$	$F(s)$	$f(t)$	$\circ \bullet$	$F(s)$
$\delta(t)$	$\circ \bullet$	1	$t^n \cdot e^{at}$	$\circ \bullet$	$\frac{n!}{(s-a)^{n+1}}$
$\delta(t - T_1)$	$\circ \bullet$	$e^{-T_1 s}$	$\sin(bt)$	$\circ \bullet$	$\frac{b}{s^2 + b^2}$
$\varepsilon(t)$	$\circ \bullet$	$\frac{1}{s}$	$\cos(bt)$	$\circ \bullet$	$\frac{s}{s^2 + b^2}$
t	$\circ \bullet$	$\frac{1}{s^2}$	$1 - e^{at}$	$\circ \bullet$	$\frac{-a}{s(s-a)}$
t^2	$\circ \bullet$	$\frac{2}{s^3}$	$e^{at} \cdot \cos(bt)$	$\circ \bullet$	$\frac{s-a}{(s-a)^2 + b^2}$
t^n	$\circ \bullet$	$\frac{n!}{s^{n+1}}$	$e^{at} \cdot \sin(bt)$	$\circ \bullet$	$\frac{b}{(s-a)^2 + b^2}$
e^{at}	$\circ \bullet$	$\frac{1}{s-a}$	$\frac{e^{at} - e^{bt}}{a-b}$	$\circ \bullet$	$\frac{1}{(s-a)(s-b)}$
$t \cdot e^{at}$	$\circ \bullet$	$\frac{1}{(s-a)^2}$	$\frac{a e^{at} - b e^{bt}}{a-b}$	$\circ \bullet$	$\frac{s}{(s-a)(s-b)}$

Hinweis: Der Einheitssprung $\varepsilon(t)$ äussert sich als eine 1 in der Gleichung im Zeitbereich
Hinweis: Teilweise braucht es Umformungen, damit die Tabelle angewendet werden kann

10.5.2 Eigenschaften der Laplace-Transformation (S. 22)

Zeitbereich	Bildbereich	Operation im Zeitbereich
$a_1 f_1(t) + a_2 f_2(t)$	$a_1 F_1(s) + a_2 F_2(s)$	Superposition
$\dot{f}(t)$	$sF(s) - f(0)$	Differentiation ...
$\ddot{f}(t)$	$s^2 F(s) - sf(0) - \dot{f}(0)$	...mehrfach angewendet
$\int_0^t f(\tau) d\tau$	$\frac{1}{s} F(s)$	Integration
$f(t - T_1) \quad [T_1 \text{ reell, } \geq 0]$	$F(s) \cdot e^{-T_1 s}$	Rechtsverschiebung
$f_1(t) * f_2(t) = \int_0^t f_1(\tau) \cdot f_2(t - \tau) dt$	$F_1(s) \cdot F_2(s)$	Faltung
$f(a \cdot t) \quad [a \text{ reell, } > 0]$	$\frac{1}{a} \cdot F\left(\frac{s}{a}\right)$	Skalierung
$e^{\sigma t} \cdot f(t)$	$F(s \pm a)$	Dämpfung

10.5.3 Grenzwertsätze der Laplace-Transformation (S. 22)

Achtung: Grenzwertsätze dürfen nur angewendet werden, wenn Grenzwerte existieren!

- Das Anfangswerttheorem gilt für Regelungstechnik immer
- Das Endwerttheorem gilt **nur**, wenn **System stabil** ist (Polstellen in linker Halbebene bzw. $\sigma < 0$)

Zeitbereich	Bildbereich	Bezeichnung
$\lim_{t \rightarrow 0} f(t) = \lim_{s \rightarrow \infty} s \cdot F(s)$		Anfangswerttheorem
$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} s \cdot F(s)$		Endwerttheorem

11 Wurzelortskurven (WOK)

11.1 Grundsätzliches Aussehen der WOK

- Char. Polynom $N_f(s) = N(s) + K \cdot Z(s)$ hat für jeden K -Wert n Wurzeln (Nullstellen)
- Visualisiert: n kontinuierliche Linien (Äste)
- WOK ist immer **symmetrisch** bezüglich der **reellen Achse**
- Für $K = 0$ sind die n Wurzeln von $N_f(s)$ und $N(s)$ identisch und entsprechen den n Polen von $G(s)$
- Anfangspunkte der Äste: n Wurzeln von $N(s) \Rightarrow$ Kreuze
- Endpunkte der Äste: m Wurzeln von $Z(s) \Rightarrow$ Kreise
- $n - m$ Äste laufen ins Unendliche

11.2 Zeichnen von Wurzelortskurven (S. 96-98)

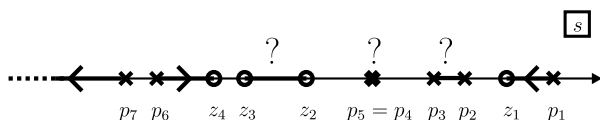
1. Pol- / Nullstellenplan:

Pole / Nullstellen von $G(s)$ werden in komplexer Zahlenebene eingetragen:

- Pole p_k ($k = 1 \dots n$) werden als $\times$ dargestellt $\Rightarrow$ Startpunkte der n Äste
- Nullstellen z_k ($k = 1 \dots m$) werden als $\circ$ dargestellt $\Rightarrow$ Endpunkte von m Ästen

2. Reelle Achse:

- Start 'ganz rechts' (weiter rechts als alle Einträge $\times$ bzw. $\circ$) auf reeller Achse starten; Stift 'nicht schreibend' bzw. 'oben'
 - Nach links wandern und bei **jedem** $\times$ bzw. $\circ$ Stift toggeln ('nicht schreibend' $\leftrightarrow$ 'schreibend')
- $\Rightarrow$ Doppel-Pole bzw. Doppel-Nullstellen entsprechend einem Punkt ('Doppel-Toggel')



$\Rightarrow$ Gültige WOK-Äste verbinden einen Pol mit einer Nullstelle.

An den Stellen mit ? (Verbindung von Pol zu Pol bzw. NS zu NS) ist die Situation noch unbefriedigend, da $n > m$. $\Rightarrow$ siehe 3. und 4.

3. Asymptoten: Schnittpunkte und Richtungen:

Wenn $n > m$ gilt, dann streben $n - m$ Äste (**Geraden**) der WOK asymptotisch gegen Unendlich. Zu berechnen ist der **Schnittpunkt** und die **Richtung der ersten der Asymptoten**. Die weiteren Asymptoten sind dann **gleichmässig** versetzt.

$$\text{Schnittpunkt: } \sigma_A = \frac{\sum_{k=1}^n \text{Re}\{p_k\} - \sum_{k=1}^m \text{Re}\{z_k\}}{n - m} \quad \begin{matrix} n & \# \text{ Pole} \\ m & \# \text{ NS} \end{matrix}$$

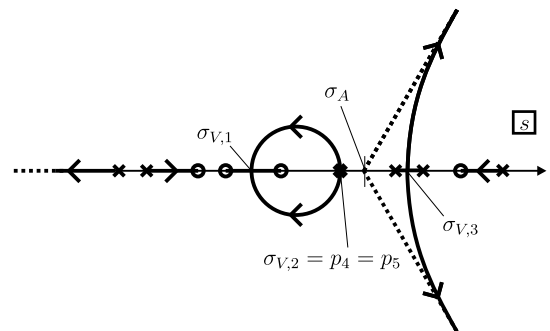
$$\text{Richtung 1. Asymptote } \phi = \frac{\pi}{n - m} \quad \text{gleichmässig versetzt um } \frac{2\pi}{n - m}$$

$\Rightarrow$ Wenn $n - m$ ungerade, dann geht immer eine Asymptote auf reeller Achse nach links

4. Verzweigungspunkte / Vereinigungspunkte:

Bei Verbindung von Pol zu Pol bzw. NS zu NS treten Verzweigungen / Vereinigungen auf. In diesen Punkten gilt Folgendes:

- Äste treten (meist) **rechtwinklig aus der reellen Achse** aus.
- Verzweigungspunkt bzw. Vereinigungspunkt befindet sich **in der Mitte** des Pol- bzw. NS-Paares
- Die **Form** der betreffenden Äste kann auf zwei Arten verlaufen:
 - Kreisförmig in Richtung **Asymptoten**, welche in der Nähe sind
 - Kreisförmig zum nächsten Verzweigungspunkt bzw. Vereinigungspunkt



Die genauen Verzweigungspunkte / Vereinigungspunkte können auch mittels folgender impliziter Formel berechnet werden:

$$\sum_{k=1}^n \frac{1}{\sigma_V - p_k} = \sum_{k=1}^m \frac{1}{\sigma_V - z_k}$$

Pole Nullstellen

Achtung: Die Formel kann unbrauchbare Lösungen (z.B. mit $\text{Im}\{\sigma_{V,i}\} \neq 0$) liefern, welche 'aussortiert' werden müssen.

11.3 WOK zu einigen geläufigen Pol- / NS-Verteilungen (S. 99)

Nr.	WOK	Nr.	WOK
1		10	
2		11	
3		12	
4		13	
5		14	
6		15	
7		16	

11.4 Auslegen eines statischen P-Reglers mittels WOK (S. 101-102)

11.4.1 Vorgehen – P-Regler mit WOK auslegen

1. WOK skizzieren (inkl. Dämpfung) gemäss Abschnitt 11.2
 2. Pole, welche bei variablem K auf Asymptoten wandern so platzieren, dass gewünschte **Dämpfung** erreicht ist
 3. **Bandbreite** ω_n und **Verstärkung** K bestimmen, indem einer der in 2. platzierten Pole betrachtet wird
- Dabei werden von einem solchen Pol aus **Verbindungslinien** zu allen weiteren Polen (θ_i) und allen Nullstellen (λ_i) gezeichnet. Die Verstärkung K berechnet sich dann aus den entsprechenden **Abständen** θ_i und λ_i gemäss

$$K = \frac{\prod_{i=1}^n \theta_i}{\prod_{i=1}^m \lambda_i} = \frac{\theta_1 \cdot \theta_2 \cdot \dots \cdot \theta_n}{\lambda_1 \cdot \lambda_2 \cdot \dots \cdot \lambda_m}$$

⇒ Falls keine Nullstellen vorhanden sind, ist das entsprechende Produkt = 1

4. Weitere Verstärkungen (z.B. K_R) aus K bestimmen

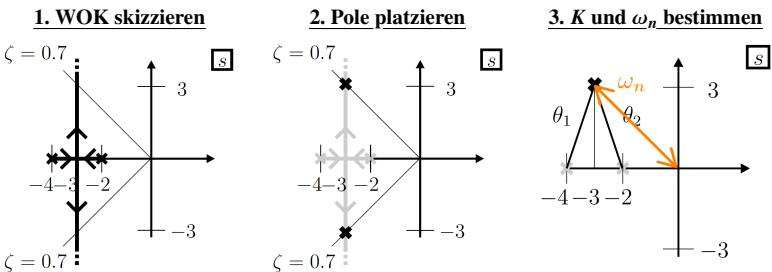
Beispiel: Auslegung eines P-Regler mittels WOK

Eine (stabile) Strecke $G_S(s)$ soll mit einem P-Regler mit $G_R(s) = K_R$ mit **möglichst grosser Verstärkung** K_R geregelt werden, wobei alle Pole eine **minimale Dämpfung** $\zeta = 0.7$ haben sollen.

Der offene Regelkreis $G_0(s)$ wird betrachtet als

$$G_0(s) = K_R \cdot \underbrace{\frac{5}{s^2 + 6s + 8}}_{G_S(s)} = \underbrace{K_R \cdot 5}_K \cdot \underbrace{\frac{1}{s^2 + 6s + 8}}_{G_S(s)}$$

Gemäss Abschnitt 11.4.1 kann nun der Regler ausgelegt werden und somit K_R aus K bestimmt werden.



4. K_R bestimmen

Aus $K = \theta_1 \cdot \theta_2 = \sqrt{1^2 + 3^2} \cdot \sqrt{1^2 + 3^2} = 10$ und $K = 5 \cdot K_R$ ergibt sich $K_R = 5$

11.4.2 Beantwortung der 'Grundfragen' für P-Regler

- Regelkreis ist stabil, wenn K gross genug gewählt wird, dass alle Pole in RHE sind
- Bandbreite: ω_n aus WOK abgelesen: $\omega_n = \sqrt{3^2 + 3^2} = \sqrt{2 \cdot 3^2} = \sqrt{2} \cdot 3 \frac{\text{rad}}{\text{s}}$
- Gewünschte Dämpfung $\zeta = 0.7$ ist einstellbar (Pole konnten so platziert werden)
- Stationärer Fehler: $e_\infty = \frac{E(s)}{R(s)} = \frac{1}{1+G_0} \Big|_{s=0} = \frac{4}{9}$

12 Matlab

12.1 Zustandsraumdarstellung (ZRD)

12.1.1 Zustandsraum-Objekte (LTI-Objekte)

```
1 z = [2, 1]; % Zaehler: Z(s) = 2s + 1
2 n = [3, 5, 2]; % Nenner: N(s) = 3s^2 + 5s + 2
3 sys_tf = tf(z,n) % transfer function Objekt (ausm.)
4 sys_zpk = zpk(sys_tf) % transfer function Objekt (fakt.)
5 sys_ss = ss(sys_tf) % state space Objekt (ZRD)
6
7 [A, B, C, D] = ssdata(sys_tf)
8 [Z, P, K] = zpkmdata(sys_tf, 'v')
9
10 [z, n] = ss2tf(A,B,C,D) % transfer function from matrices
11 (A,B,C,D) = tf2ss(z, n) % matrices from transfer function
```

12.1.2 Spezielle Matrizen im Zustandsraum

```
1 syms t % t als symbolischer Wert
2 A = [0 6; 1 5]; % Matrix A (2x2)
3 B = [1; 4]; % Matrix B (2x1)
4 C = [2; 3]; % Matrix C (1x2)
5
6 SteuerMat = ctrb(A, B); % Steuerbarkeitsmatrix
7 BeobachtMat = obsv(A, C); % Beobachtbarkeitsmatrix
8 expm(A*t) % Transitionsmatrix
```

12.2 Bode- und Nyquist

```
1 z = [1, 2, 5]; % Zaehlerpolynom: Z(s) = s^2 + 2s + 5
2 n = [3, 5]; % Nennerpolynom: N(s) = 3s + 5
3 T_t = 0.5 % Totzeit
4 roots(n) % Nullstellen
5 G = (tf(z, n, 'inputdelay', T_t)) % UTF (mit Totzeit)
6 bode(G) % Bode-Plot des Systems
7 margin(G) % Bode mit Stabilitaets- und Phasenreserve
8 bodemag(G) % Amplitudengang des Systems
9
10 nyquist(G) % Nyquist-Plot des Systems
```

12.3 z-Transformation

```
1 s = tf('s');
2 G_R = K_R * (1 + s * T_N) / (s * T_N); % UTF Regler
3 sysd = c2d(G_R, T_S, 'tustin') % T_S: sampling time
```

12.4 Implementation eines diskreten Reglers

```
1 Function init()
2     e_kml = 0 % e(k-1) initialisieren
3     u_kml = 0 % u(k-1) initialisieren
4
5 Function u_k = loop(e_k)
6     u_k = u_kml + K_R / (2 * T_N) * (e_k * (T + 2 * T_N))
7         + e_kml * (T - 2 * T_N)
8
9     u_kml = u_k
10    e_kml = e_k
```

12.5 Wurzelortskurven

```
1 z = [1, 5, 1]; % Zaehlerpolynom Z(s) = s^2 + 5s + 1
2 n = [1, 4]; % Nennerpolynom N(s) = s + 4
3 G = tf(z, n) % UTF offener Regelkreis
4 rlocus(G) % Wurzelortskurve generieren
```

12.6 Zustandsrückführungen / Beobachter

```
1 A = [0 1; 1 5]; % Matrix A (2x2)
2 B = [0; 1]; % Matrix B (2x1)
3 C = [1, 0]; % Matrix C (1x2)
4 Q = [2 0; 0 1]; % Matrix Q (2x2)
5 R = [1]; % Matrix R (1x1)
6 p = [-5 -10]; % vorgegebene Pole
7
8 % Verstärkungsmatrix der Rueckfuehrung mit LQR-Methode
9 K = lqr(A, B, Q, R)
10
11 % Verstärkungsmatrix der Rueckfuehrung mit Polplatzierung
12 K = place(A, B, p)
13
14 % Verstärkungsmatrix des Beobachters mit LQR-Methode
15 H = lqr(A', C', Q, R)'
16
17 % Verstärkungsmatrix des Beobachters mit Polplatzierung
18 H = place(A', C', p)
```

13 Statische Grundglieder (ohne Gedächtnis)

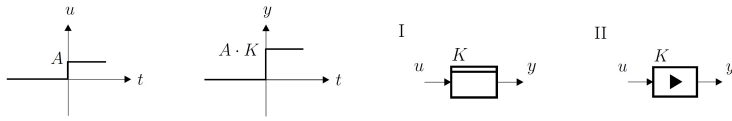
Der Ausgang des Systems ist **nur** vom aktuellen Eingang abhängig

$$y(t) = f(u(t))$$

Hinweis: $u(t)$ entspricht meist $A \cdot \varepsilon(t)$ (skalierter Einheitssprung)

13.1 P-Glied (Proportional)

$$y(t) = K \cdot u(t)$$



14 Dynamische Grundglieder (mit Gedächtnis)

Der Ausgang des Systems ist vom aktuellen **und** von vergangenen Eingängen abhängig. Vergangene Eingänge werden im System in einem Speicher, einem Zustand $x(t)$ abgelegt.

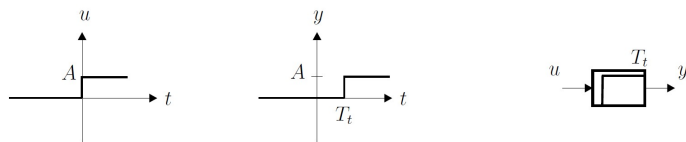
$$y(t) = f(u(t), x(t))$$

14.0.1 Prozesse mit / ohne Ausgleich

- ohne Ausgleich: Sprungantwort wächst grenzenlos an
- mit Ausgleich: Sprungantwort strebt gegen endlichen Wert
→ **Gegenkopplung** am Ausgang

14.1 Totzeit-Glied

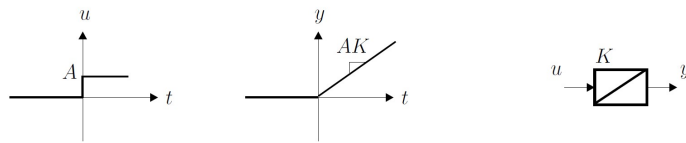
$$y(t) = u(t - T_t)$$



14.2 I-Glied (Integrierer)

$$\text{DGL: } \dot{y}(t) = K \cdot u(t)$$

$$y = \int_{-\infty}^t u(\tau) d\tau = K \int_{-\infty}^t u(\tau) d\tau + y(0)$$

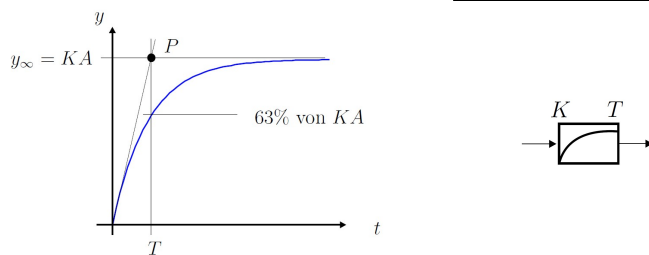


14.3 PT₁-Glied

Ein Energiespeicher ⇒ System schwingt nicht

$$\text{DGL: } T \dot{y}(t) + y(t) = K \cdot u(t)$$

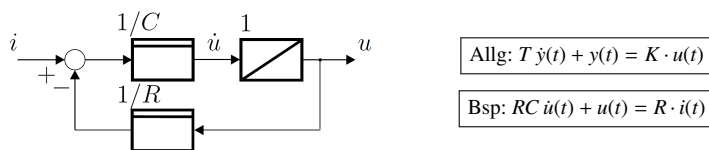
$$y(t) = K A + (y_0 - K A) \cdot e^{-t/T}$$



14.3.1 Modellierung einer PT₁ Regelstrecke

- Ausmessen der Sprungantwort $u(t) = A \cdot \varepsilon(t)$
⇒ Die Sprungantwort entspricht dem Diagramm
- Parameter K berechnen: $K = \frac{y_{\infty}}{A}$
- Parameter T bestimmen: Tangente an Anfang durch Punkt P
oder: Zeit bis $y(T) = 0.63 \cdot K A$ erreicht ist ⇒ T ablesen

Beispiel: PT₁-Glied als Blockschaltbild

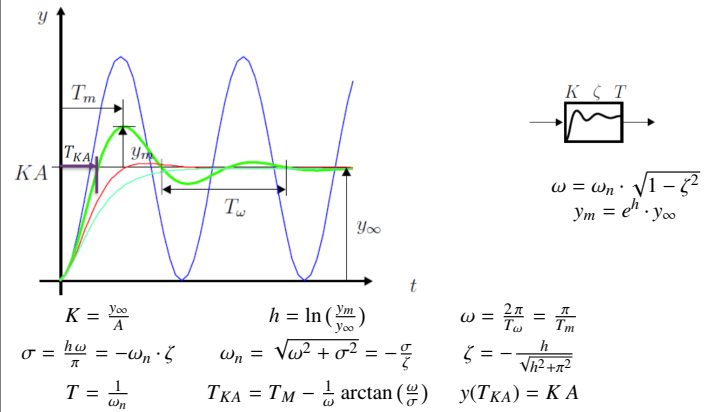


14.4 PT₂-Glied

Zwei Energiespeicher ⇒ System kann schwingen

$$\text{DGL: } T^2 \cdot \ddot{y} + 2 \zeta T \cdot \dot{y} + y = K \cdot u$$

$$y(t) = K A [1 + e^{\sigma t} (-\cos(\omega t) + \frac{\sigma}{\omega} \sin(\omega t))]$$



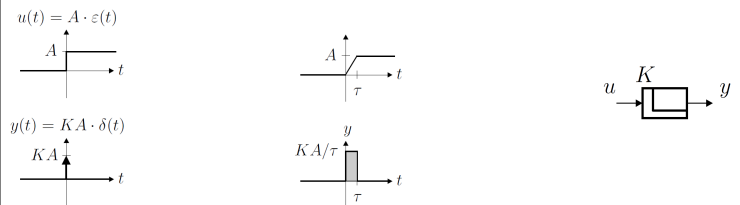
⇒ $\zeta = \frac{1}{\sqrt{2}}$ entspricht 5 % Überschwingen!

14.5 D-Glied (Idealer Differenzierer)

$$y(t) = K \cdot \frac{du(t)}{dt} = K \cdot \dot{u}(t)$$

ideal

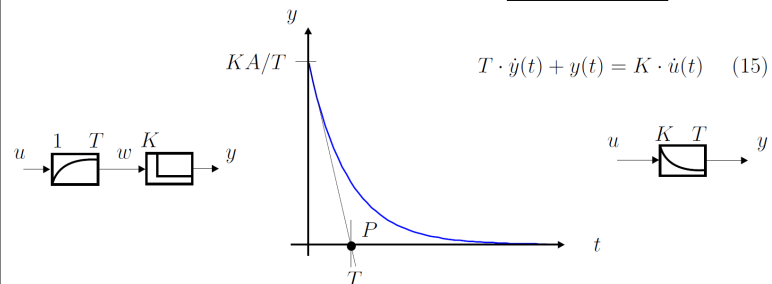
real



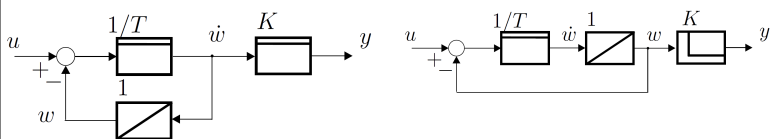
14.6 DT₁-Glied (Realisierbarer Differenzierer)

$$w(t) = A \cdot (1 - e^{-t/T})$$

$$y(t) = \frac{K \cdot A}{T} e^{-t/T}$$



Beispiel: Mögliche Realisierungen von DT₁-Gliedern



14.7 Lead-Glied / Lag-Glied / Lead-Lag-Glied

Lead-/ Lag-Glieder sind zwei Varianten von PDT₁-Gliedern.

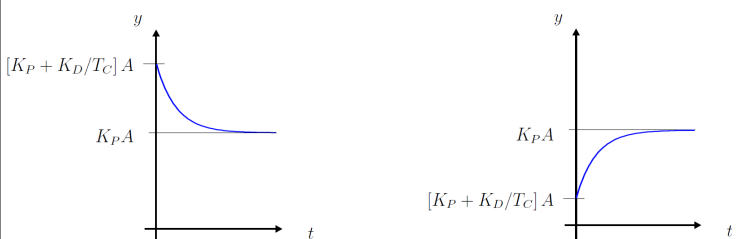
Lead: K_P und K_D haben **gleiches** Vorzeichen

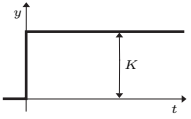
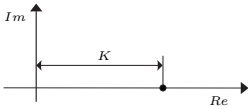
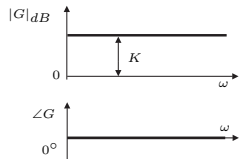
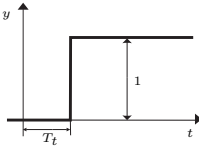
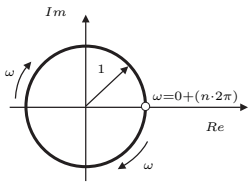
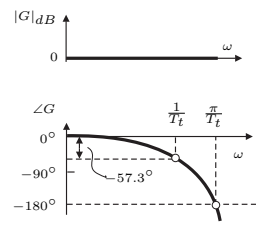
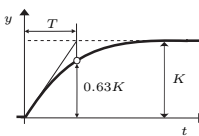
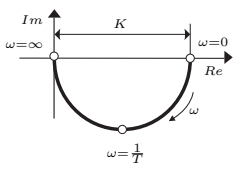
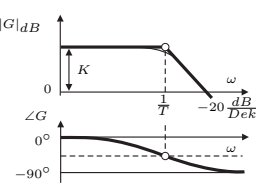
Lag: K_P und K_D haben **unterschiedliches** Vorzeichen

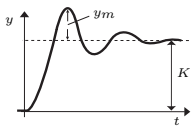
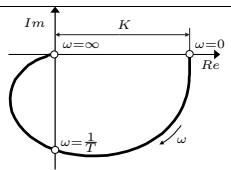
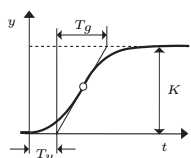
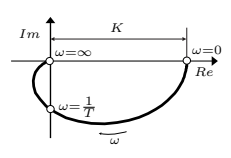
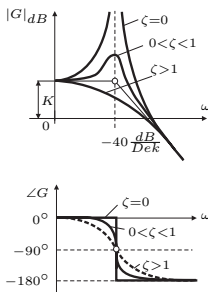
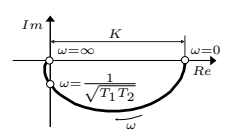
Lead- und Lag-Glieder können auch als Parallelschaltungen eines P- und eines PT₁-Glieds aufgefasst werden. Das Lead-Lag-Glied ist eine Serieschaltung aus Lead- und Lag-Glied.

Lead-Glied

Lag-Glied

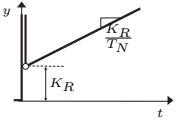
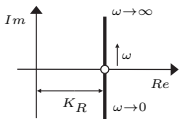
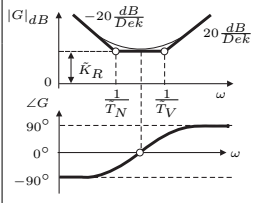
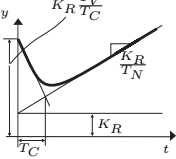
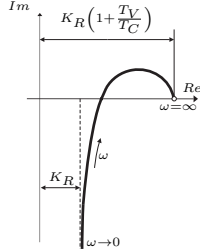
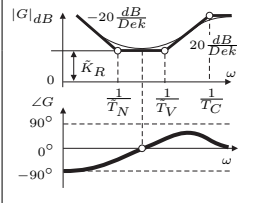


Typ	Differential- gleichung	Frequenzgang $G(j\omega)$	Schrittantwort $u(t) = \varepsilon(t)$	Nyquistdiagramm (Ortskurve)	Bodediagramm (dB $\hat{=} 20 \cdot \log_{10}$)
P	$y(t) = Ku(t)$	K			
T_t	$y(t) = u(t - T_t)$	$e^{-j\omega T_t}$			
PT_1	$T\dot{y}(t) + y(t) = Ku(t)$	$\frac{K}{j\omega T + 1}$			

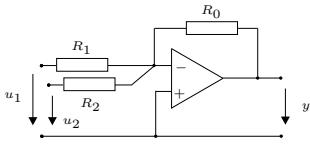
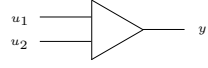
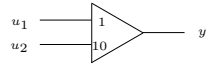
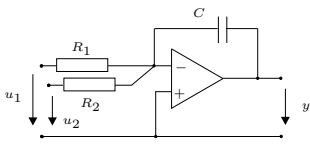
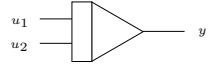
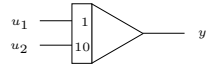
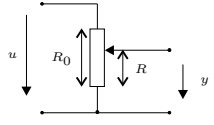
Typ	Differentialgleichung	Frequenzgang $G(j\omega)$	Schrittantwort $u(t) = \varepsilon(t)$	Nyquistdiagramm (Ortskurve)	Bodediagramm (dB $\hat{=} 20 \cdot \log_{10}$)
PT ₂	allgemein: $T^2\ddot{y}(t) + 2\zeta T\dot{y}(t) + y(t) = Ku(t)$	$\frac{K}{(j\omega)^2 T^2 + 2j\omega\zeta T + 1}$			
	$0 < \zeta < 1$ unterkritisch gedämpft, periodisch	$\frac{K}{(j\omega)^2 T^2 + 2j\omega\zeta T + 1}$ (nicht reell faktorisierbar)			
	$\zeta = 1$ kritisch gedämpft, aperiodisch ($\hat{=}$ zwei identische PT ₁ -Glieder in Serie)	$\frac{K}{(j\omega T + 1)^2}$			
	$\zeta > 1$ überkritisch gedämpft, aperiodisch ($\hat{=}$ zwei PT ₁ -Glieder in Serie)	$\frac{K}{(j\omega T_1 + 1)(j\omega T_2 + 1)}$ $T_{1,2} = T(\zeta \pm \sqrt{\zeta^2 - 1})$ $T = \sqrt{T_1 T_2}$ $\zeta = \frac{T_1 + T_2}{2\sqrt{T_1 T_2}}$	Ersatzmodell: $\frac{K}{1 + j\omega T_g} \cdot e^{-j\omega T_u}$		

Typ	Differentialgleichung	Frequenzgang $G(j\omega)$	Schrittantwort $u(t) = \varepsilon(t)$	Nyquistdiagramm (Ortskurve)	Bodediagramm $(dB \hat{=} 20 \cdot \log_{10})$
I	$y(t) = K \int_0^t u(\tau) d\tau$ $\dot{y} = Ku(t)$	$\frac{K}{j\omega}$			
PI	$y(t) = K[u(t) + \frac{1}{T} \int_0^t u(\tau) d\tau]$	$K \left(1 + \frac{1}{j\omega T}\right)$ bzw. $K \frac{1+j\omega T}{j\omega T}$			
D	$y(t) = K\dot{u}(t)$	$j\omega K$			
DT ₁	$T\dot{y}(t) + y(t) = K\dot{u}(t)$	$\frac{j\omega K}{1+j\omega T}$			

Typ	Differential- gleichung	Frequenzgang $G(j\omega)$	Schrittantwort $u(t) = \varepsilon(t)$	Nyquistdiagramm (Ortskurve)	Bodediagramm $(dB \hat{=} 20 \cdot \log_{10})$
PD	$y(t) = K[u(t) + T\dot{u}(t)]$	$K(1+j\omega T)$			
PDT ₁ (Lead-Glied) mit $T_1 > T_2$	$T_2\dot{y}(t) + y(t) = K[u(t) + T_1\dot{u}(t)]$	$K \frac{1+j\omega T_1}{1+j\omega T_2}$			
PPT ₁ (Lag-Glied) mit $T_1 < T_2$	$T_2\dot{y}(t) + y(t) = K[u(t) + T_1\dot{u}(t)]$	$K \frac{1+j\omega T_1}{1+j\omega T_2}$			

Typ	Differential- gleichung	Frequenzgang $G(j\omega)$	Schrittantwort $u(t) = \varepsilon(t)$	Nyquistdiagramm (Ortskurve)	Bodediagramm ($\text{dB} \hat{=} 20 \cdot \log_{10}$)
PID	$y(t) = K_R [u(t) + \frac{1}{T_N} \int_0^t u(\tau) d\tau + T_V \dot{u}(t)]$	Additive Form: $K_R \left(1 + \frac{1}{j\omega T_N} + j\omega T_V \right)$ Multiplikative Form: $\tilde{K}_R \frac{(1+j\omega \tilde{T}_N)(1+j\omega \tilde{T}_V)}{j\omega \tilde{T}_N}$	Parameter zu additiver Form 	Parameter zu additiver Form 	Parameter zu multiplikativer Form 
PIDT ₁	Kann aus dem Frequenzgang bestimmt werden: mit $K_R =$ $\tilde{K}_R \left(1 + \frac{\tilde{T}_V}{\tilde{T}_N} \right)$ $T_N = \tilde{T}_N + \tilde{T}_V$ $T_V = \frac{\tilde{T}_N \tilde{T}_V}{\tilde{T}_N + \tilde{T}_V}$	Additive Form: $K_R \left(1 + \frac{1}{j\omega T_N} + \frac{j\omega T_V}{1+j\omega T_C} \right)$ Multiplikative Form: $\tilde{K}_R \frac{(1+j\omega \tilde{T}_N)(1+j\omega [T_C + \tilde{T}_V])}{j\omega \tilde{T}_N (1+j\omega T_C)}$	Parameter zu additiver Form 	Parameter zu additiver Form 	Parameter zu multiplikativer Form 

16 Analogrechner

Schaltung	Komponenten	I/O-Gleichung	Symbol Analogrechner
 Summator mit Vorzeichenwechsel	$R_0 = R_1 = R_2 = 1 \text{ [M}\Omega\text{]}$ $K = \frac{R_0}{R_1} = \frac{R_0}{R_2} = 1$ $R_0 = R_1 = 1 \text{ [M}\Omega\text{]}$ $R_2 = R_1/10$ $K_1 = \frac{R_0}{R_1} = 1 \quad K_2 = \frac{R_0}{R_2} = 10$	$y(t) = -K[u_1(t) + u_2(t)]$ $= -u_1(t) - u_2(t)$ $y(t) = -K_1 u_1(t) - K_2 u_2(t)$ $= -u_1(t) - 10 \cdot u_2(t)$	 
 Integrator mit Vorzeichenwechsel	$C = 1 \text{ [}\mu\text{F}\text{]}$ $R_1 = R_2 = 1 \text{ [M}\Omega\text{]}$ $K = \frac{1}{R_1 C} = \frac{1}{R_2 C} = 1 \text{ [s}^{-1}\text{]}$ $C = 1 \text{ [}\mu\text{F}\text{]}$ $R_1 = 1 \text{ [M}\Omega\text{]} \quad R_2 = R_1/10$ $K_1 = \frac{1}{R_1 C} = 1 \text{ [s}^{-1}\text{]} \quad K_2 = 10 \text{ [s}^{-1}\text{]}$	$\frac{dy(t)}{dt} = -K[u_1(t) + u_2(t)]$ $= -u_1(t) - u_2(t)$ $\frac{dy(t)}{dt} = -K_1 u_1(t) - K_2 u_2(t)$ $= -u_1(t) - 10 \cdot u_2(t)$	 
 Spannungsteiler (passiv)	$R_0 = 1 \text{ [M}\Omega\text{]}$ $K = \frac{R}{R_0} \in [0 \dots 1]$	$y = K \cdot u(t)$	