

Zustandsraum und Digitale Regelung

von

Prof. Dr.-Ing. Oliver Nelles

Inhaltsverzeichnis

1. Zustandsraumregelung
2. Digitale Regelung: Einführung
3. Kurzübersicht: Zeitdiskrete Systeme
4. Stabilität zeitdiskreter Systeme
5. Deadbeat-Regler
6. Adaptive Regelung

Literatur

Zustandsraum:

- Lunze: „Regelungstechnik 1“, 9. Aufl., Springer 2012

Digitale Regelung:

- Lunze: „Regelungstechnik 2“, 7. Aufl., Springer 2013
- Åström und Wittenmark: „Computer-Controlled Systems“, Prentice Hall, 1996
- Isermann: „Digitale Regelsysteme, Band 1“, 2. Aufl., Springer, 2008
- Lutz, Wendt: „Taschenbuch der Regelungstechnik“, 9. Aufl., Deutsch, 2012

1. Zustandsraumregelung

Inhalt Kapitel 2

1. Zustandsraumregelung

- 1.1 Beschreibung dynamischer Systeme im Zustandsraum
- 1.2 Lösung der Zustandsgleichungen
- 1.3 Eigenschaften der Zustandsgleichungen
- 1.4 Zustandsregler durch Polvorgabe
- 1.5 Zustandsregler durch Optimierung (LQ)
- 1.6 Beobachter
- 1.7 Zustandsregler mit Beobachter (LQG)
- 1.8 Folgeregelung (*Tracking*)
- 1.9 Führungs- und Störgrößenmodelle

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Eigenschaften der Zustandsraummethoden

- DGL n . Ordnung \rightarrow System aus n DGLs jeweils 1. Ordnung.
- Beschreibung des Systems nicht nur durch das Ein-/Ausgangsverhalten sondern durch (innere) Zustände mittels Zustandsvariablen ermöglicht tiefere Einsichten.
- Entwurfsverfahren vorwiegend im Zeitbereich.
- Mathematisches Modell der Regelstrecke erforderlich.
- Mächtigkeit der linearen Algebra wird voll ausgenutzt.
- Numerisch robuste Verfahren \rightarrow MATLAB.
- Entstand in den 1960er Jahren mit bahnbrechenden Arbeiten von Kalman.
- Wesentlicher technologischer Treiber für die Luft- und Raumfahrt (Flug zum Mond).
- Gut erweiterbar auf Mehrgrößensysteme.
- Gut erweiterbar auf zeitvariante Systeme.
- Gut erweiterbar auf nichtlineare Systeme.
- Komplexe Systeme hoher Ordnung sind übersichtlicher darstellbar.

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Matrix- und Vektorrechnung

Einen Vektor kann man verschieden interpretieren:

- Die Informatik-Sichtweise: Als 1-dimensionales Feld, z.B. zur Implementierung einer Warteschlange oder eines Stacks.
- Die Ingenieur-Sichtweise: Als Punkt in einem n -dimensionalen Raum bzw. als Zeiger vom Ursprung des Koordinatensystems zu diesem Punkt.

Bei Vektoren unterscheiden wir zwischen Zeilen und Spaltenvektoren.

- Standardmäßig liegt ein Vektor immer in *Spaltenform* vor:

$$\underline{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

- Um einen *Zeilenvektor* zu erhalten, muss ein Spaltenvektor transponiert werden:

$$\underline{a}^T = (a_1 \quad a_2 \quad \cdots \quad a_n)$$

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Matrix- und Vektorrechnung

- Das *innere Produkt* oder *Skalarprodukt* zweier Vektoren liefert einen Skalar. Hierfür müssen beide Vektoren die selbe Dimension haben:

$$\underline{a}^T \underline{b} = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} = a_1 b_1 + a_2 b_2 + \dots + a_n b_n = \sum_{i=1}^n a_i b_i$$

- Das *äußere Produkt* oder *dyadische Produkt* zweier Vektoren liefert eine Matrix. Hierfür müssen beide Vektoren **nicht** die selbe Dimension haben:

$$\underline{a} \underline{b}^T = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{pmatrix} \begin{pmatrix} b_1 & b_2 & \cdots & b_n \end{pmatrix} = \begin{pmatrix} a_1 b_1 & a_1 b_2 & \cdots & a_1 b_n \\ a_2 b_1 & a_2 b_2 & \cdots & a_2 b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_m b_1 & a_m b_2 & \cdots & a_m b_n \end{pmatrix}$$

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Matrix- und Vektorrechnung

Eine Matrix kann man auf vielerlei Arten interpretieren:

- Die Sicht des Informatikers: Eine Matrix ist ein 2-dimensionales Feld. Die Spalten stellen z.B. x -Koordinaten und die Zeilen y -Koordinaten dar. Eine $m \times n$ -Matrix könnte die Bildpunkte eines Bildes mit n Pixeln in der Waagrechten und m Pixeln in der Senkrechten repräsentieren, z.B. $n = 1024$, $m = 768$.

$$\underline{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

- Jedes Element (Matrizeintrag) könnte bei einem Graubild den Schwarzanteil angeben: 0 (0% Schwarz = Weiß), 127 (50% Schwarz = Grau), 255 (100% Schwarz = Schwarz).
- Die logische Erweiterung eines solchen Matrixbegriffs ist ein Tensor 3. Stufe, der durch ein 3-dimensionales Feld realisiert wird, das Voxel in einem Volumen beschreibt (statt Pixel auf einer Fläche).

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Matrix- und Vektorrechnung

- Die Sicht des Ingenieurs: Eine Matrix \underline{A} repräsentiert eine (lineare) Abbildung eines Vektors \underline{x} (Eingang) auf einen anderen Vektor \underline{y} (Ausgang):

$$\underline{y} = \underline{A} \underline{x}$$

- Im allgemeinen Fall können die Vektoren unterschiedliche Dimensionen aufweisen, z.B. $\dim(\underline{x}) = n$, $\dim(\underline{y}) = m$. Dann ist die Matrix rechteckig:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

$m \times 1$ $m \times n$ $n \times 1$

$$y_i = \sum_{j=1}^n a_{ij} x_j$$

\nearrow \nwarrow
i. Ausgang j. Eingang

- Gilt $n > m$ (mehr Spalten als Zeilen), dann bezeichnet man die Matrix als *fett*.
- Gilt $n < m$ (mehr Zeilen als Spalten), dann bezeichnet man die Matrix als *schlank* (*skinny*).
- Gilt $n = m$ (gleich viele Spalten und Zeilen), dann ist die Matrix *quadratisch*.

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Matrix- und Vektorrechnung

Wir haben es in MRT II meist mit quadratischen Matrizen zu tun. In diesem Fall kann die Abbildung \underline{A} als Skalierungs- und Rotationsoperation verstanden werden, d.h. der Vektor \underline{x} wird durch Multiplikation mit \underline{A} in einen Vektor \underline{y} abgebildet, indem die Länge und der Winkel von \underline{x} verändert werden. Jede (lineare) Transformation in ein anderes Koordinaten-system kann durch eine Matrixmultiplikation durchgeführt werden.

Beispiel: Rotation des Koordinatensystems

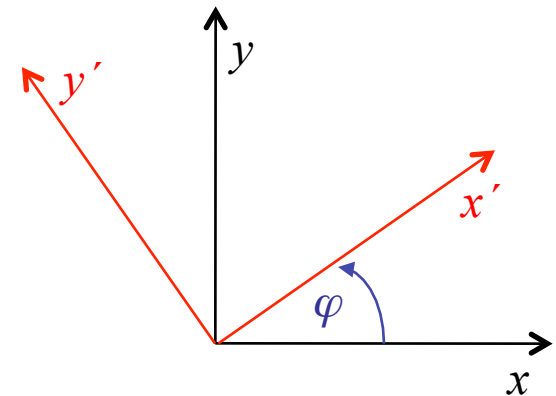
Die x - und y -Achsen sollen jeweils um den Winkel φ im Gegenuhrzeigersinn gedreht werden.

Folgende Transformations-Matrix tut dies:

$$\underline{T} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}$$

Mit ihr können wir die alten Koordinaten x und y in die neuen Koordinaten x' und y' transformieren:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \underline{T} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \cos\varphi - y \sin\varphi \\ x \sin\varphi + y \cos\varphi \end{pmatrix}$$



1.1 Beschreibung dynamischer Systeme im Zustandsraum

Matrix- und Vektorrechnung

Eine Matrix-Vektor-Multiplikation kann auch spaltenweise interpretiert werden:

- Je eine Spalte \underline{a}_j der Matrix \underline{A} ist einem Eingang x_j zugeordnet:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

- Daher kann man die Matrix-Vektor-Multiplikation auch als gewichtete Summe der Spaltenvektoren \underline{a}_j der Matrix \underline{A} mit den Eingängen x_j als Gewichten verstehen:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = x_1 \underline{a}_1 + x_2 \underline{a}_2 + x_3 \underline{a}_3 + x_4 \underline{a}_4 \quad \text{mit} \quad \underline{A} = \left(\underline{a}_1 \quad \underline{a}_2 \quad \underline{a}_3 \quad \underline{a}_4 \right)$$

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Matrix- und Vektorrechnung

Eine Matrix-Vektor-Multiplikation kann auch zeilenweise interpretiert werden:

- Je eine Zeile \underline{a}_i^T der Matrix \underline{A} ist einem Ausgang y_i zugeordnet:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

- Daher kann man die Matrix-Vektor-Multiplikation auch als Satz von Skalarprodukten aus den Zeilenvektoren \underline{a}_i^T der Matrix \underline{A} mit den Eingangsvektoren \underline{x} verstehen:

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} \underline{a}_1^T \underline{x} \\ \underline{a}_2^T \underline{x} \\ \underline{a}_3^T \underline{x} \\ \underline{a}_4^T \underline{x} \end{pmatrix} \quad \text{mit} \quad \underline{A} = \begin{pmatrix} \underline{a}_1^T \\ \underline{a}_2^T \\ \underline{a}_3^T \\ \underline{a}_4^T \end{pmatrix}$$

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Das Ein-/Ausgangsverhalten eines linearen dynamischen Systems n . Ordnung mit Eingang $u(t)$ und Ausgang $y(t)$ lässt sich durch folgende Übertragungsfunktion beschreiben:

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

Ohne Einschränkung wurde hier $a_n = 1$ gesetzt. Wir unterscheiden 2 Fälle:

- $m < n$: Normalfall. Das System ist *nicht* sprunghfähig.
- $m = n$: Kommt nur ausnahmsweise vor. Das System ist sprunghfähig.

Im Zustandsraum lässt sich ein solches System wie folgt beschreiben:

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{b} u(t)$$

(Zustandsdifferentialgleichungen)

$$y(t) = \underline{c}^T \underline{x}(t) + d u(t)$$

(Ausgangsgleichung)

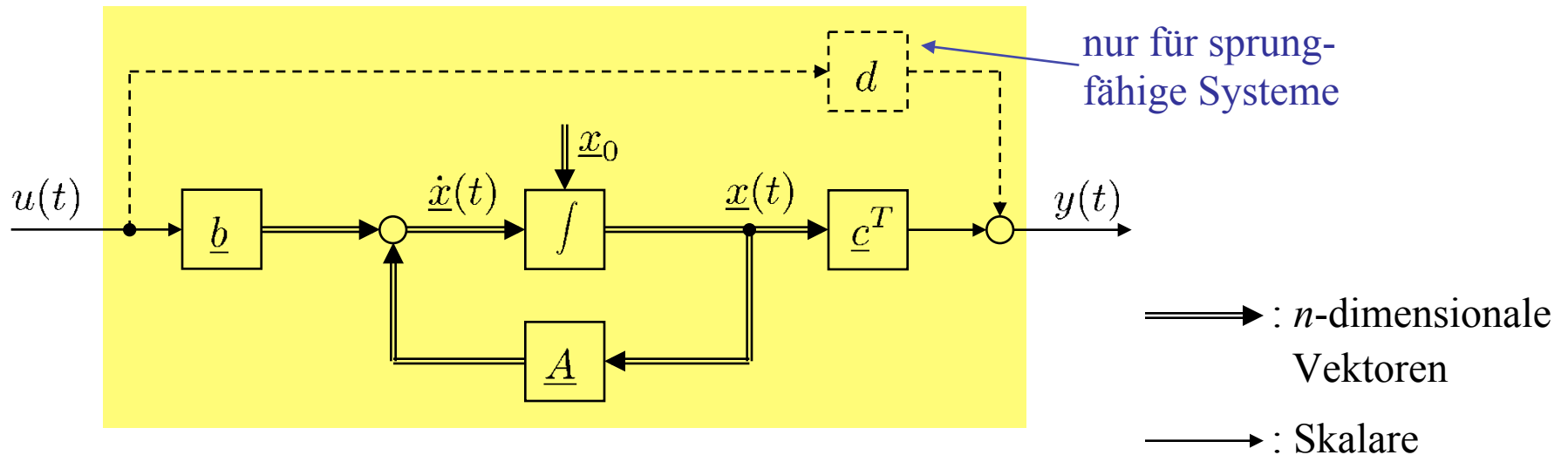
\underline{A} : $n \times n$ Systemmatrix

\underline{c}^T : $1 \times n$ Ausgangsvektor

\underline{b} : $n \times 1$ Eingangsvektor

d : Durchgriff (= 0 für $m < n$, also normalerweise)

1.1 Beschreibung dynamischer Systeme im Zustandsraum



$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} u(t)$$

$$y(t) = (c_1 \ c_2 \ \cdots \ c_n) \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix} + d u(t)$$

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Wie hängen die Matrizen und Vektoren der Zustandsgleichungen mit den Koeffizienten der DGL bzw. der Übertragungsfunktion zusammen?

Betrachten wir zur Vereinfachung zunächst den *Spezialfall*, bei dem der Zähler der Übertragungsfunktion gleich 1 ist:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0} = \frac{1}{A(s)} \quad \rightarrow \quad Y(s) = \frac{1}{A(s)}U(s)$$

$$\rightarrow s^n Y(s) + a_{n-1}s^{n-1}Y(s) + \dots + a_1sY(s) + a_0Y(s) = U(s)$$

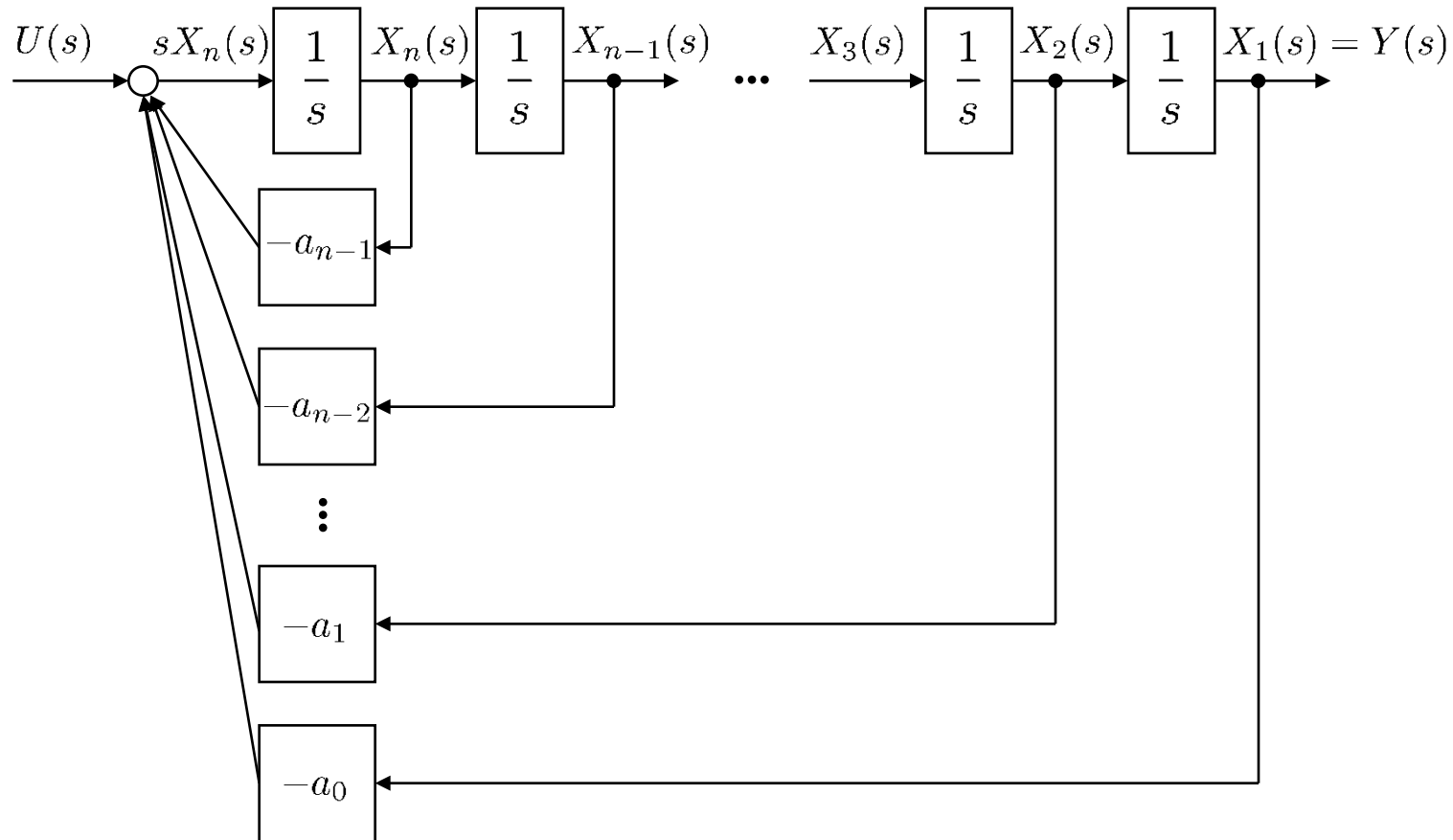
Da ein System n . Ordnung vorliegt, müssen wir n Zustände definieren. Diese Wahl ist *nicht eindeutig*. Jeder Zustand muss aber einem Energiespeicher bzw. Integrator des Systems zugeordnet werden.

$$\underbrace{s^n Y(s)}_{sX_n(s)} + a_{n-1} \underbrace{s^{n-1} Y(s)}_{X_n(s)} + \dots + a_1 \underbrace{sY(s)}_{X_2(s)} + a_0 \underbrace{Y(s)}_{X_1(s)} = U(s)$$

↑
Ableitung von $X_n(s)$ } n Zustände

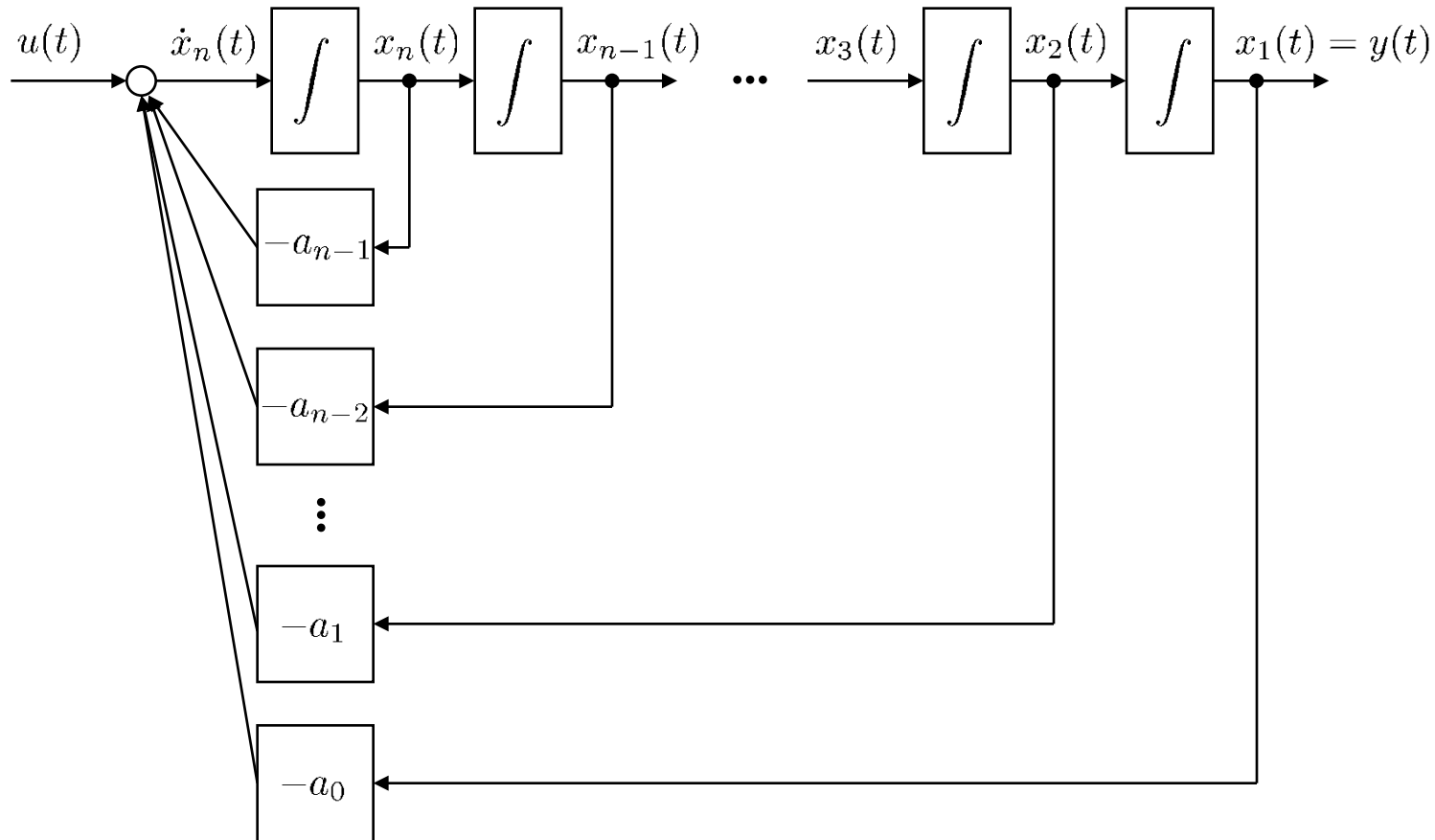
1.1 Beschreibung dynamischer Systeme im Zustandsraum

Schema im Frequenzbereich



1.1 Beschreibung dynamischer Systeme im Zustandsraum

Schema im Zeitbereich



1.1 Beschreibung dynamischer Systeme im Zustandsraum

Aus dem Schema kann man leicht die Zustandsgleichungen aufstellen:

$$\dot{x}_1(t) = x_2(t)$$

$$\dot{x}_2(t) = x_3(t)$$

\vdots

$$\dot{x}_{n-1}(t) = x_n(t)$$

$$\dot{x}_n(t) = -a_{n-1}x_n(t) - a_{n-2}x_{n-1}(t) - \dots - a_1x_2(t) - a_0x_1(t) + u(t)$$

$$y(t) = x_1(t)$$

In Matrix/Vektor-Schreibweise:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{n-1}(t) \\ \dot{x}_n(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u(t)$$

$$y(t) = (1 \ 0 \ \dots \ 0) \underline{x}(t)$$

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Erweitern wir dieses Ergebnis auf Systeme mit beliebigem Zählerpolynom, aber zunächst nur für nicht sprungfähige Systeme ($m < n$):

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

$$\rightarrow Y(s) = \left(\frac{b_m s^m}{A(s)} + \frac{b_{m-1} s^{m-1}}{A(s)} + \dots + \frac{b_1 s}{A(s)} + \frac{b_0}{A(s)} \right) U(s)$$

Beim vorherigen Beispiel mit Zähler = 1 war $X_1(s) = Y(s)$ und $Y(s) = \frac{1}{A(s)} U(s)$.
Behalten wir die Definition der Zustände bei so gilt weiterhin:

$$X_1(s) = \frac{1}{A(s)} U(s)$$

ACHTUNG: $Y(s)$ ist nun natürlich wegen des vollständigen Zählerpolynoms anders definiert.
Wir übernehmen aus dem einfachen Fall aber die Definition der Zustände. Damit ergibt sich:

$$\begin{aligned} Y(s) &= b_m s^m X_1(s) + b_{m-1} s^{m-1} X_1(s) + \dots + b_1 s X_1(s) + b_0 X_1(s) \\ &= b_m X_{m+1}(s) + b_{m-1} X_m(s) + \dots + b_1 X_2(s) + b_0 X_1(s) \end{aligned}$$

1.1 Beschreibung dynamischer Systeme im Zustandsraum

X_{m+1} existiert natürlich nur, wenn $m < n$. Für den Fall $m = n$ muss es durch sX_n ersetzt werden. Für nicht sprungfähige Systeme (also $m < n$) ergeben sich die Zustandsgleichungen:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{n-1}(t) \\ \dot{x}_n(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u(t)$$

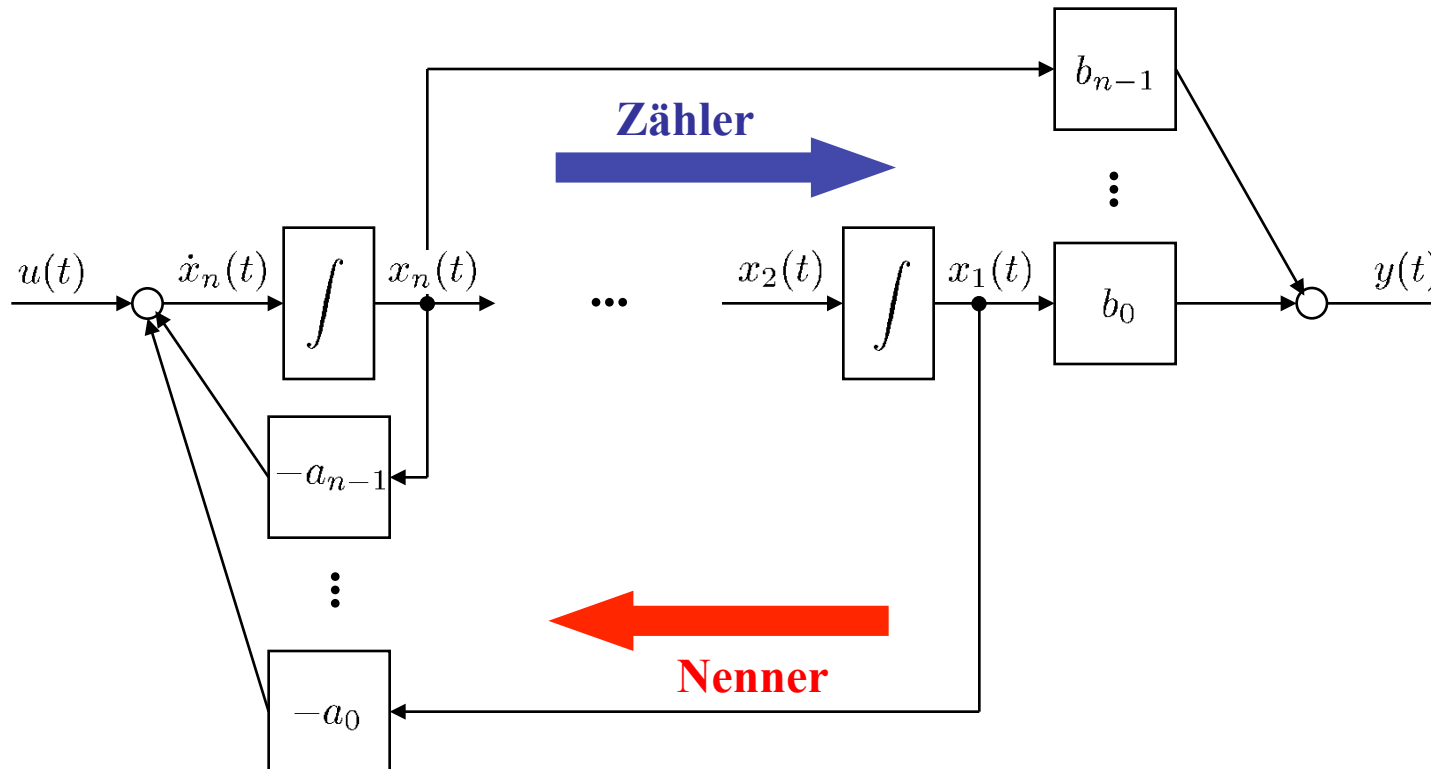
$$y(t) = (b_0 \quad b_1 \quad \cdots \quad b_{n-2} \quad b_{n-1}) \underline{x}(t)$$

Diese spezielle Form der Zustandsgleichungen nennt man **Regelungsnormalform**. Sie hat u.a. folgenden schöne Eigenschaften:

- Die Zählerkoeffizienten der Übertragungsfunktion tauchen direkt in \underline{c}^T auf.
- Die Nennerkoeffizienten der Übertragungsfunktion tauchen direkt in \underline{A} auf.
- Die Systemmatrix \underline{A} hat eine ganz spezielle Struktur; von den $n \times n$ Elementen sind nur die n Elemente der letzten Zeile vom System abhängig.
- Der Eingangsvektor \underline{b} ist ganz unabhängig von den Systemeigenschaften.

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Schema der Regelungsnormform für *nicht* sprungfähige Systeme im Zeitbereich



Aus jeder Übertragungsfunktion lassen sich also direkt die Zustandsgleichungen in *Regelungsnormform* aufstellen. Dazwischen existiert eine 1:1-Beziehung.

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Für sprungfähige System ist $m = n$ und die Gleichung

$$Y(s) = b_m X_{m+1}(s) + b_{m-1} X_m(s) + \dots + b_1 X_2(s) + b_0 X_1(s)$$

geht über in:

$$Y(s) = b_n s X_n(s) + b_{n-1} X_n(s) + \dots + b_1 X_2(s) + b_0 X_1(s)$$

Eliminiert man den ersten Term mittels (siehe Schema im Frequenzbereich)

$$sX_n(s) = -a_{n-1}X_n(s) - \dots - a_1X_2(s) - a_0X_1(s) + U(s)$$

ergeben sich die Zustandsgleichungen in Regelungsnormalform für sprungfähige Systeme:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{n-1}(t) \\ \dot{x}_n(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u(t)$$

$$y(t) = (b_0 \underbrace{-b_n a_0}_{\text{zusätzliche Terme}} \quad b_1 \underbrace{-b_n a_1}_{\text{zusätzliche Terme}} \quad \dots \quad b_{n-2} \underbrace{-b_n a_{n-2}}_{\text{zusätzliche Terme}} \quad b_{n-1} \underbrace{-b_n a_{n-1}}_{\text{Durchgriff}}) \underline{x}(t) + \underbrace{b_n}_{\text{Durchgriff}} u(t)$$

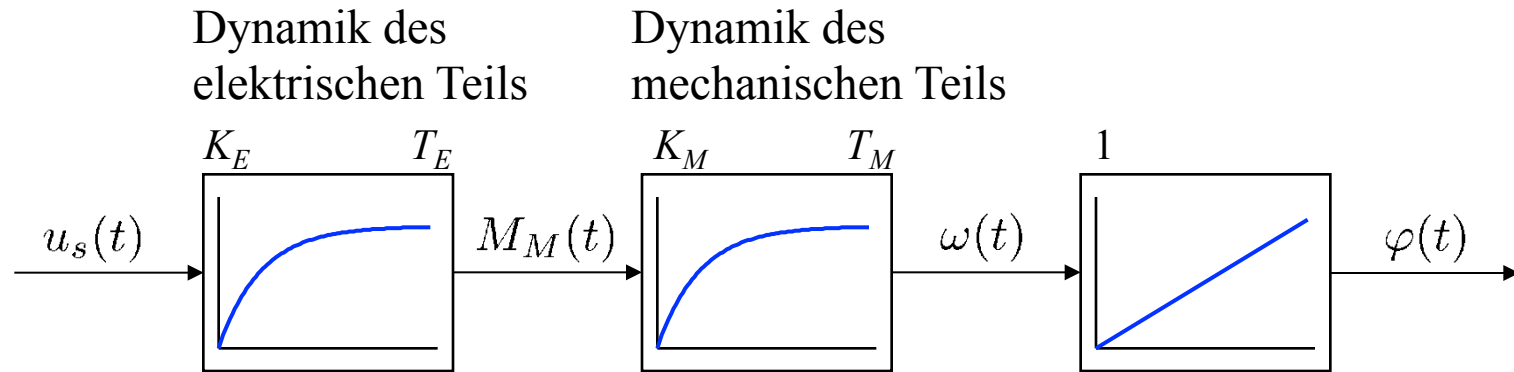
zusätzliche Terme

Durchgriff

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Direktes Aufstellen der Zustandsgleichungen (ohne Umweg über Übertragungsfunktion)

Lageregelstrecke eines elektrischen Antriebs



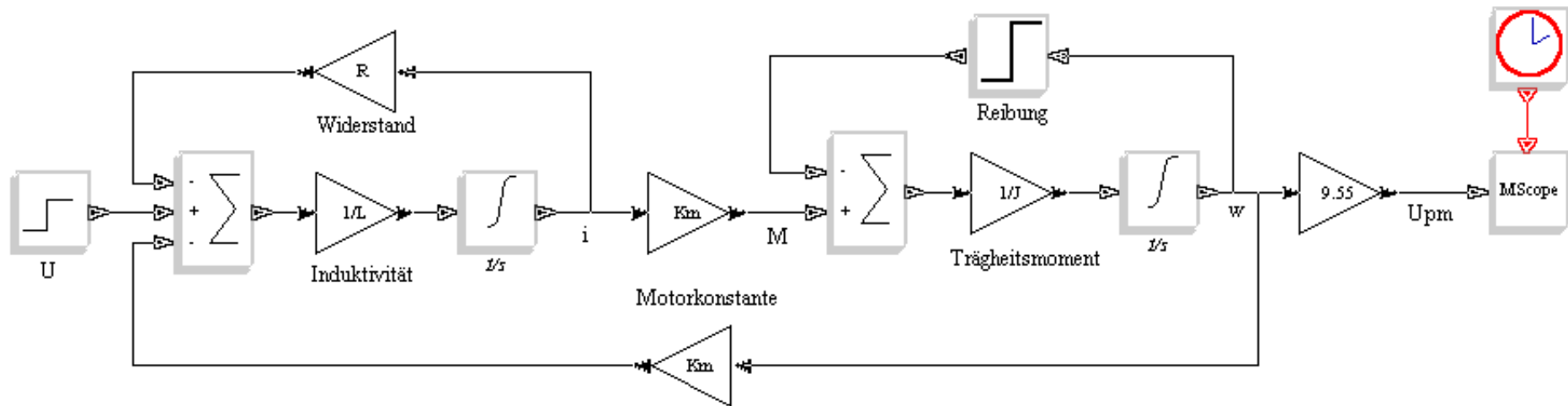
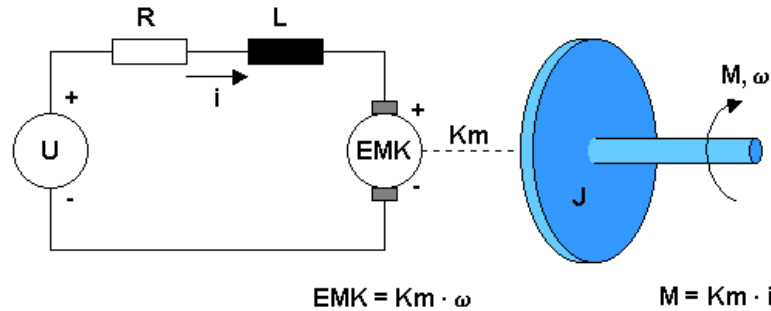
Eingangsgröße: $u(t) = u_s(t)$ Stromrichtereingangsspannung

Ausgangsgröße: $y(t) = \varphi(t)$ Drehwinkel

Zustandsgrößen: $x_1(t) = \varphi(t)$ Drehwinkel
 $x_2(t) = \omega(t)$ Drehzahl / Winkelgeschwindigkeit
 $x_3(t) = M_M(t)$ Motormoment

} Reihenfolge
ist willkürlich

1.1 Beschreibung dynamischer Systeme im Zustandsraum



Quelle: <http://www.rn-wissen.de/index.php/Regelungstechnik>

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Mit dieser Wahl der Zustände ergeben sich folgende Zustandsgleichungen (Mit einer anderen Wahl der Zustände ergäben sich andere Zustandsgleichungen, die aber dasselbe Ein-/Ausgangsverhalten zeigen würden!):

$$\dot{x}_1(t) = x_2(t) \quad (\text{Integrator})$$

$$x_2(t) + T_M \dot{x}_2(t) = K_M x_3(t) \quad \rightarrow \quad \dot{x}_2(t) = -\frac{1}{T_M} x_2(t) + \frac{K_M}{T_M} x_3(t) \quad (\text{mechanisches PT}_1)$$

$$x_3(t) + T_E \dot{x}_3(t) = K_E u(t) \quad \rightarrow \quad \dot{x}_3(t) = -\frac{1}{T_E} x_3(t) + \frac{K_E}{T_E} u(t) \quad (\text{elektrisches PT}_1)$$

$$y(t) = x_1(t)$$

In Matrix/Vektor-Schreibweise:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -\frac{1}{T_M} & \frac{K_M}{T_M} \\ 0 & 0 & -\frac{1}{T_E} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{K_E}{T_E} \end{pmatrix} u(t)$$

$$y(t) = (1 \ 0 \ 0) \underline{x}(t)$$

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Wie können wir die Zustandsgleichungen nun z.B. in die Regelungsnormalform überführen?

Es gibt 2 alternative Wege:

1. Aufstellen der Übertragungsfunktion. → Erstellen der Zustandsgleichungen.
2. Direkte Transformation im Zustandsraum (das lernen wir in Kapitel 1.3).

Durchführen des 1. Wegs:

$$G(s) = \frac{K_E}{1 + T_E s} \frac{K_M}{1 + T_M s} \frac{1}{s} = \frac{b_2 s^2 + b_1 s + b_0}{s^3 + a_2 s^2 + a_1 s + a_0}$$

b_2 b_1 b_0
 $0s^2 + 0s + \frac{K_E K_M}{T_E T_M}$
 $s^3 + \frac{T_E + T_M}{T_E T_M} s^2 + \frac{1}{T_E T_M} s + 0$
 a_2 a_1 a_0

Aus den Zähler- und Nennerkoeffizienten ergeben sich direkt die Zustandsgleichungen in Regelungsnormalform:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\frac{1}{T_E T_M} & -\frac{T_E + T_M}{T_E T_M} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u(t)$$

$$y(t) = \begin{pmatrix} \frac{K_E K_M}{T_E T_M} & 0 & 0 \end{pmatrix} \underline{x}(t)$$

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Definition: Zustand eines dynamischen Systems

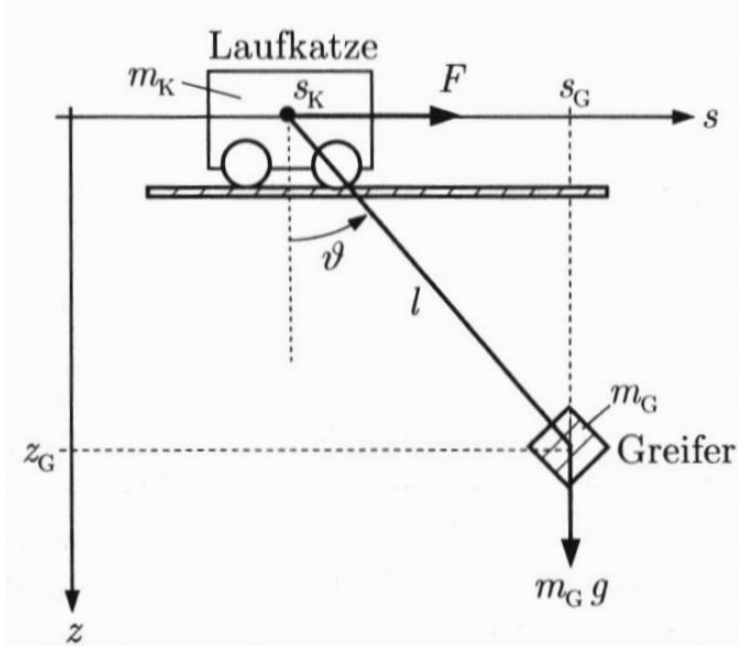
Ein Vektor \underline{x} wird *Zustandsvektor* (und seine Elemente *Zustände*) eines Systems genannt, wenn sich allein mittels seines Anfangswertes $\underline{x}(0)$ und des Verlaufs der Eingangsgröße $u(t)$, $0 \leq t \leq t_e$, die Ausgangsgröße $y(t_e)$ bestimmen lässt, und dies für eine beliebige Zeit t_e .

Dies bedeutet u.a.:

- Im Zustandsvektor muss alle innere Information über das System stecken. Ansonsten wäre die Berechnung der Ausgangsgröße nicht eindeutig möglich.
- Wenn wir jedem Energiespeicher bzw. jedem Integrator einen Zustand zuordnen, haben wir diese Forderung erfüllt.
- Der Zustandsvektor *kann* mehr Elemente enthalten als nötig. Wenn der Zustandsvektor nur die mindestens notwendige Anzahl von Zuständen enthält, dann spricht man von einer **minimalen Realisierung** (*minimal realization*).
- Es gibt unendlich viele Wahlmöglichkeiten für den Zustandsvektor (Realisierungen), die auf dasselbe Ein-/Ausgangsverhalten führen. Realisierungen, die bestimmte nützliche Eigenschaften aufweisen, nennt man **kanonische Formen** oder **Normalformen**.

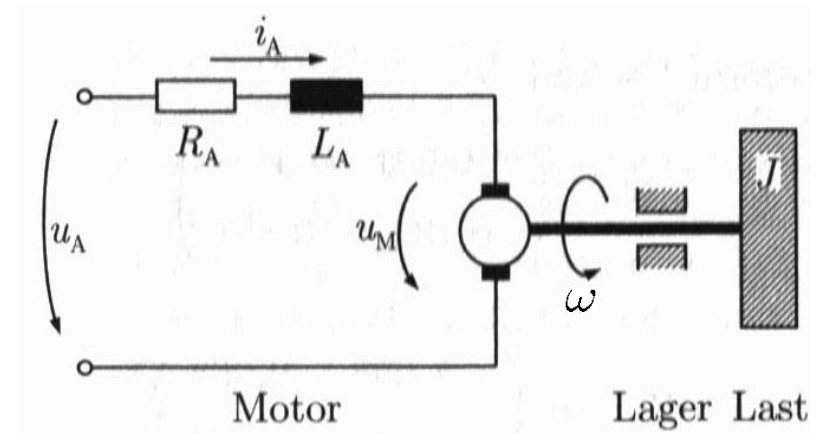
1.1 Beschreibung dynamischer Systeme im Zustandsraum

Weitere Beispiele für die Wahl der Zustände



2. Ordnung

Zustände: Position Katze s_K
Geschwindigkeit Katze \dot{s}_K
Position Greifer s_G
Geschwindigkeit Greifer \dot{s}_G



1. Ordnung (wenn Ausgang = Drehzahl)

Zustände: Ankerstrom i_A
Winkelgeschwindigkeit ω

3. Ordnung (wenn Ausgang = Drehwinkel)

Zustände: Ankerstrom i_A
Winkelgeschwindigkeit $\dot{\varphi}$
Drehwinkel φ

1.1 Beschreibung dynamischer Systeme im Zustandsraum

Warum spricht man vom Zustandsraum?

Wir können uns den Zustandsvektor als Punkt in einem n -dimensionalen Raum vorstellen. Die Zustandsdifferentialgleichung

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{b} u(t)$$

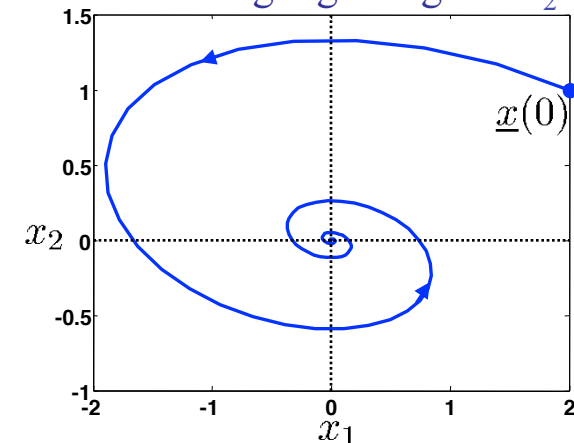
beschreibt dann, wie sich dieser Punkt im Zustandsraum bewegt. Wenn die Eingangsgröße $u(t) = 0$ ist, dann entwickelt sich der Punkt entsprechend der Eigenbewegung des Systems:

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t)$$

Wenn es sich um ein stabiles System handelt, dann wird der Punkt $\underline{x}(t)$ gegen den Ursprung des Zustandsraums streben, d.h. $\underline{x}(t \rightarrow \infty) \rightarrow \underline{0}$.

Die Funktion $\underline{f}(\underline{x}) = \underline{A} \underline{x}$ bildet einen Vektor auf einen Vektor ab. Man nennt eine solche Funktion ein **Vektorfeld**. Der Punkt bewegt sich in diesem Vektorfeld entlang der Feldlinien, wie ein Teilchen in einem Strömungsfeld entlang der Strömungslinien. Für 2- und 3-dimensionale Vektorfelder ist eine graphische Veranschaulichung in der Ebene bzw. im Raum möglich.

Zustandsvektor eines schwingungsfähigen PT_2



1.1 Beschreibung dynamischer Systeme im Zustandsraum

Erweiterung auf Mehrgrößensysteme

Wir wollen uns hier bewusst auf Systeme mit einem Ein- und Ausgang beschränken, da die Behandlung von Mehrgrößensystemen viele zusätzliche Überlegungen erfordert, die das unmittelbare Verständnis erschweren. Es soll aber kurz gezeigt werden, wie leicht sich die Beschreibung dynamischer Systeme im Zustandsraum formal auf den Mehrgrößenfall erweitern lässt:

1 Eingang, 1 Ausgang:

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{b} u(t)$$

$$y(t) = \underline{c}^T \underline{x}(t) + d u(t)$$

$$\underline{A}: n \times n, \quad \underline{b}: n \times 1$$

$$\underline{c}^T: 1 \times n, \quad d: 1 \times 1$$

n_u Eingänge, n_y Ausgänge:

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{B} \underline{u}(t)$$

$$\underline{y}(t) = \underline{C} \underline{x}(t) + \underline{D} \underline{u}(t)$$

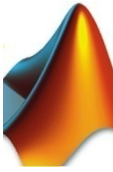
$$\underline{A}: n \times n, \quad \underline{B}: n \times n_u$$

$$\underline{C}: n_y \times n, \quad \underline{D}: n_y \times n_u$$

Die meisten Gleichungen lassen sich leicht formal vom Ein- auf den Mehrgrößenfall erweitern. Im Mehrgrößenfall treten aber viel zusätzliche Schwierigkeiten auf, die eine gesonderte und ausführlichere Behandlung erforderlich machen.

1.1 Beschreibung dynamischer Systeme im Zustandsraum

MATLAB



Erzeugen eines dynamische Systems im Zustandsraum:

```
System = ss(A,b,c,d);           % Erzeugt Zustandsgleichungen
[A,b,c,d] = ssdata(System);    % Gibt Matrizen und Vektoren
                                % zurück
```

Umwandeln von Übertragungsfunktionen in den Zustandsraum und umgekehrt:

```
[A,b,c,d] = tf2ss(Zaehler,Nenner) % Wandelt Übertragungsfunktion
                                     % Zaehler- und Nennerpolynom in
                                     % Zustandsraumform um.
[Zaehler,Nenner] = ss2tf(A,b,c,d); % Umwandlung anders herum.
```

Berechnung *einer* minimalen Realisierung:

```
System_min = minreal(System)     % Berechnet eine minimale
                                   % Realisierung von
"System"
```

1.2 Lösung der Zustandsgleichungen

Wir wollen nun berechnen, wie sich der Zustandsvektor über die Zeit entwickelt; und zwar in Abhängigkeit eines Anfangswertes $\underline{x}(0)$ und des zeitlichen Verlaufs des Eingangs $u(t)$. Wenn der Zustandsvektor $\underline{x}(t)$ erst einmal berechnet worden ist, dann können wir mit der Ausgangsgleichung den Ausgang $y(t)$ ganz leicht bestimmen.

Starten wir also mit den Zustandsdifferentialgleichungen:

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{b} u(t) \quad \text{mit} \quad \underline{x}(0) = \underline{x}_0$$

Analog zum eindimensionalen Fall (siehe MRT I) machen wir einen e-Funktionen-Ansatz:

$$\underline{x}(t) = e^{\underline{A}t} \underline{k}(t)$$

Hierbei ist $\underline{k}(t)$ ein n -dimensionaler Vektor und die sog. **Matrixexponentialfunktion** ist wie folgt definiert:

$$e^{\underline{A}t} = \sum_{i=0}^{\infty} \frac{\underline{A}^i t^i}{i!} = \underbrace{\underline{I}}_{\text{Einheitsmatrix}} + \underline{A}t + \frac{\underline{A}^2}{2!} t^2 + \frac{\underline{A}^3}{3!} t^3 + \dots$$

Ganz analog zur Reihenentwicklung der skalaren e-Funktion:

$$e^{at} = \sum_{i=0}^{\infty} \frac{a^i t^i}{i!} = 1 + at + \frac{a^2}{2!} t^2 + \frac{a^3}{3!} t^3 + \dots$$

1.2 Lösung der Zustandsgleichungen

Die Ableitung der Matrixexponentialfunktion berechnet sich wie folgt:

$$\begin{aligned}\frac{d}{dt}e^{\underline{A}t} &= \frac{d}{dt} \left[\underline{I} + \underline{A}t + \frac{\underline{A}^2}{2!}t^2 + \frac{\underline{A}^3}{3!}t^3 + \dots \right] = \underline{A} + \underline{A}^2t + \frac{\underline{A}^3}{2!}t^2 + \frac{\underline{A}^4}{3!}t^3 + \dots \\ &= \underline{A} \left(\underline{I} + \underline{A}t + \frac{\underline{A}^2}{2!}t^2 + \frac{\underline{A}^3}{3!}t^3 + \dots \right) = \left(\underline{I} + \underline{A}t + \frac{\underline{A}^2}{2!}t^2 + \frac{\underline{A}^3}{3!}t^3 + \dots \right) \underline{A}\end{aligned}$$

Da sich \underline{A} sowohl links- als auch rechtsseitig ausklammern lässt, ergibt sich:

$$\frac{d}{dt}e^{\underline{A}t} = \underline{A}e^{\underline{A}t} = e^{\underline{A}t}\underline{A}$$

Einsetzen von $\underline{x}(t) = e^{\underline{A}t}\underline{k}(t)$ (Variation der Konstanten) in die Zustandsdifferentialgleichungen liefert:

$$\underline{A}e^{\underline{A}t}\underline{k}(t) + e^{\underline{A}t}\dot{\underline{k}}(t) = \underline{A}e^{\underline{A}t}\underline{k}(t) + \underline{b}u(t) \quad \rightarrow \quad e^{\underline{A}t}\dot{\underline{k}}(t) = \underline{b}u(t)$$

Eine Inversion der Matrixexponentialfunktion ergibt: $(e^{\underline{A}t})^{-1} = e^{-\underline{A}t}$. Daraus folgt:

$$\dot{\underline{k}}(t) = e^{-\underline{A}t}\underline{b}u(t)$$

1.2 Lösung der Zustandsgleichungen

Integration liefert:

$$\int_0^t \dot{\underline{k}}(\tau) d\tau = \underline{k}(t) - \underline{k}(0) = \int_0^t e^{-\underline{A}\tau} \underline{b} u(\tau) d\tau \quad \rightarrow \quad \underline{k}(t) = \underline{k}(0) + \int_0^t e^{-\underline{A}\tau} \underline{b} u(\tau) d\tau$$

Mit $\underline{x}(t) = e^{\underline{A}t} \underline{k}(t)$ und $\underline{x}(0) = \underline{x}_0 = \underline{k}(0)$ erhält man:

Bewegungsgleichung: $\underline{x}(t) = e^{\underline{A}t} \underline{x}_0 + \int_0^t e^{\underline{A}(t-\tau)} \underline{b} u(\tau) d\tau$

Oft wird folgende Abkürzung eingeführt:

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

Diese Funktion wird **Übergangsmatrix** oder **Transitionsmatrix** oder **Fundamentalmatrix** genannt. Mit ihr wird die Bewegungsgleichung:

$$\underline{x}(t) = \underline{\Phi}(t) \underline{x}_0 + \int_0^t \underline{\Phi}(t - \tau) \underline{b} u(\tau) d\tau$$

entspricht der Gewichtsfunktion $g(t)$ im skalaren Fall

1.2 Lösung der Zustandsgleichungen

Die Lösung der Zustandsdifferentialgleichungen setzt sich aus 2 Teilen zusammen:

$$\underline{x}(t) = \underbrace{e^{\underline{A}t} \underline{x}_0}_{\underline{x}_{\text{frei}}(t)} + \underbrace{\int_0^t e^{\underline{A}(t-\tau)} \underline{b} u(\tau) d\tau}_{\underline{x}_{\text{erzw}}(t)}$$

Die homogene Lösung $\underline{x}_{\text{frei}}(t)$ beschreibt die Eigenbewegung des Systems ausgehend von einem Anfangswert \underline{x}_0 ohne Eingangssignal, d.h. mit $u(t) = 0$. Die partikuläre Lösung $\underline{x}_{\text{erzw}}(t)$ beschreibt die Reaktion des Systems auf einen bestimmten Verlauf der Eingangsgröße $u(\tau)$, $0 \leq \tau \leq t$. Für stabile Systeme klingt der homogene Lösungsanteil exponentiell ab und spielt daher nach einer gewissen Zeit für das Ein-/Ausgangsverhalten keine Rolle mehr.

Der Ausgang des Systems berechnet sich direkt aus dem Zustandsvektor (und für sprungfähige Systeme auch aus dem Eingang) $y(t) = \underline{c}^T \underline{x}(t) + d u(t)$. Dies führt zu:

$$\text{Ausgangsbewegungsgleichung: } y(t) = \underbrace{\underline{c}^T e^{\underline{A}t} \underline{x}_0}_{y_{\text{frei}}(t)} + \underbrace{\int_0^t \underline{c}^T e^{\underline{A}(t-\tau)} \underline{b} u(\tau) d\tau + d u(t)}_{y_{\text{erzw}}(t)}$$

1.2 Lösung der Zustandsgleichungen

Lösung der Zustandsdifferentialgleichungen im Frequenzbereich:

$$s\underline{X}(s) - \underline{x}_0 = \underline{A} \underline{X}(s) + \underline{b}U(s)$$

$$Y(s) = \underline{c}^T \underline{X}(s) + dU(s)$$

Auflösen der Zustandsdifferentialgleichungen nach $\underline{X}(s)$ liefert:

$$(s\underline{I} - \underline{A})\underline{X}(s) = \underline{b}U(s) + \underline{x}_0 \quad \rightarrow \quad \underline{X}(s) = (s\underline{I} - \underline{A})^{-1}(\underline{b}U(s) + \underline{x}_0)$$

Einheitsmatrix
↓
 $s\underline{X}(s) = s\underline{I} \underline{X}(s)$

Die Matrix $s\underline{I}$ hat immer vollen Rang; deswegen ist $s\underline{I} - \underline{A}$, außer für einzelne Werte $s = s_i$, invertierbar. Der Ausgang ergibt sich dann zu:

$$Y(s) = \underline{c}^T (s\underline{I} - \underline{A})^{-1}(\underline{b}U(s) + \underline{x}_0) + dU(s)$$

Auch im Frequenzbereich zeigen sich die beiden Lösungsanteile abhängig vom Anfangswert und vom Eingang Y_{frei} und Y_{erzw} . Die Übertragungsfunktion lautet demnach:

$$G(s) = \frac{Y(s)}{U(s)} = \underline{c}^T (s\underline{I} - \underline{A})^{-1} \underline{b} + d$$

1.2 Lösung der Zustandsgleichungen

Beispiel: Gegeben sei folgende Übertragungsfunktion: $G(s) = \frac{Y(s)}{U(s)} = \frac{4}{s^2 + 2s + 3}$

Das entspricht in Regelungsnormform den Zustandsgleichung mit

$$\underline{A} = \begin{pmatrix} 0 & 1 \\ -3 & -2 \end{pmatrix} \quad \underline{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \underline{c} = \begin{pmatrix} 4 \\ 0 \end{pmatrix} \quad d = 0$$

Daraus berechnet sich die Übertragungsfunktion zu:

$$\begin{aligned} G(s) &= \underline{c}^T (s\underline{I} - \underline{A})^{-1} \underline{b} + d = (4 \quad 0) \left[\begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ -3 & -2 \end{pmatrix} \right]^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + 0 \\ &= (4 \quad 0) \begin{pmatrix} s & -1 \\ 3 & s+2 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = (4 \quad 0) \frac{1}{s(s+2) + 3 \cdot 1} \begin{pmatrix} s+2 & 1 \\ -3 & s \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= (4 \quad 0) \begin{pmatrix} \frac{s+2}{s^2+2s+3} & \frac{1}{s^2+2s+3} \\ \frac{-3}{s^2+2s+3} & \frac{s}{s^2+2s+3} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = (4 \quad 0) \begin{pmatrix} \frac{1}{s^2+2s+3} \\ \frac{s}{s^2+2s+3} \end{pmatrix} = \frac{4}{s^2 + 2s + 3} \end{aligned}$$

1.2 Lösung der Zustandsgleichungen

Stabilität der Zustandsgleichungen

Vorüberlegungen:

- Die Ausgangsgleichung bildet lediglich den Zustandsvektor auf den Ausgang ab. Sie ist daher für Stabilitätsbetrachtungen irrelevant.
- In den Zustandsdifferentialgleichungen kann für die Stabilitätsuntersuchung $u(t) = 0$ gesetzt werden, denn der Verlauf des Eingangssignals hat nichts mit der Stabilität zu tun.

Wir untersuchen also nur die homogenen Zustandsdifferentialgleichungen:

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t)$$

Die Stabilität des Systems hängt also offensichtlich nur von der Systemmatrix \underline{A} ab!

Im Folgenden werden wir sehen, dass die Stabilität von den **Eigenwerten** von \underline{A} abhängt.
Genauer formuliert:

Ein dynamisches System sei durch Zustandsgleichungen beschrieben. Dieses System ist genau dann stabil, wenn *alle* Eigenwerte der Systemmatrix \underline{A} einen negativen Realteil haben.

1.2 Lösung der Zustandsgleichungen

Wie hängen anschaulich Eigenwerte und Eigenvektoren mit der Stabilität zusammen?

Erinnern wir uns an die geometrische Bedeutung von Eigenwerten und Eigenvektoren. Sie sind durch folgenden Zusammenhang bestimmt:

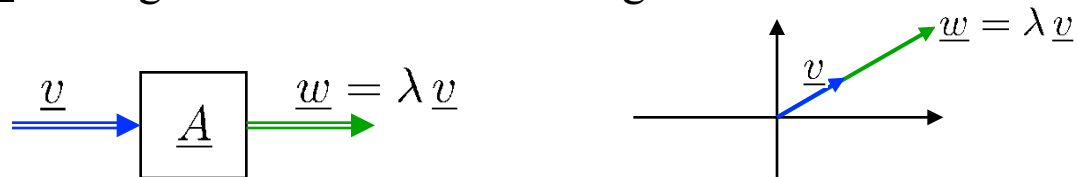
$$\underline{A} \underline{v} = \lambda \underline{v}$$

Ein **Eigenvektor** der Matrix \underline{A} wird durch die Matrix \underline{A} auf ein Vielfaches seiner selbst abgebildet. Dieses Vielfache (kann auch < 1 sein) nennt man **Eigenwert**.

Generell bildet eine Matrix einen Vektor auf einen Vektor ab. Bei einer quadratischen Matrix sind Eingangsvektor und Ausgangsvektor von der selben Dimension:



Eine solche Abbildung kann man sich als eine Drehung (im n -dimensionalen Raum) und eine Skalierung des Vektors vorstellen. Findet keine Drehung statt, die Richtung des Eingangsvektors \underline{v} bleibt also erhalten, dann ist \underline{v} ein Eigenvektor. Der Skalierungsfaktor λ ist dann der zugehörige Eigenwert.



1.2 Lösung der Zustandsgleichungen

Eine $n \times n$ Matrix hat n Eigenvektoren und n zugehörige Eigenwerte (die nicht alle unterschiedlich sein müssen). Unter bestimmten Bedingungen, z.B. wenn die Matrix \underline{A} lauter verschiedene reelle Eigenwerte hat, sind die Eigenvektoren linear unabhängig. Dann spannen die Eigenvektoren den gesamten n -dimensionalen Zustandsraum auf. D.h. jeder Zustandsvektor kann als Linearkombination der Eigenvektoren geschrieben werden:

$$\underline{x} = \alpha_1 \underline{v}_1 + \alpha_2 \underline{v}_2 + \dots + \alpha_n \underline{v}_n$$

Kehren wir zurück zur Stabilitätsuntersuchung:

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t)$$

Näherungsweise können wir auch schreiben:

$$\Delta \underline{x}(t) = \underline{A} \underline{x}(t) \Delta t$$

Der Zustandsvektor \underline{x} ändert sich also pro Zeiteinheit entsprechend dem, was auf der rechten Seite der Gleichung steht. Das System ist genau dann stabil, wenn der Zustandsvektor in jedem Schritt kleiner wird, d.h. dessen Änderung *negativ* ist. Da sich jedes \underline{x} aus den Eigenvektoren von \underline{A} zusammensetzen lässt, können wir Stabilität garantieren, wenn *alle* zugehörigen Eigenwerte *negativ* sind. Alle Eigenvektoren schrumpfen und deshalb auch jedes \underline{x} , das sich aus diesen Eigenvektoren zusammensetzt.

1.2 Lösung der Zustandsgleichungen

Berechnung der Eigenwerte und Eigenvektoren

Ausgehend von der Definitionsgleichung $\underline{A}\underline{v} = \lambda \underline{v}$ erhalten wir:

$$(\lambda \underline{I} - \underline{A})\underline{v} = \underline{0}$$

Da wir hier an der Trivillösung $\underline{v} = \underline{0}$ nicht interessiert sind, muss $\lambda \underline{I} - \underline{A}$ singulär sein, d.h.:

$$\det(\lambda \underline{I} - \underline{A}) = 0$$

Dies ist die **charakteristische Gleichung** des Systems. Das Polynom $\det(\lambda \underline{I} - \underline{A})$ heißt **charakteristisches Polynom** und hat den Grad n . Seine n Nullstellen sind die Eigenwerte der Matrix \underline{A} und damit Pole der zugehörigen Übertragungsfunktion (vorausgesetzt es gibt keine Pol/Nullstellen-Kürzungen).

Um die Eigenvektoren zu berechnen, setzen wir nach und nach die Eigenwerte in $\underline{A}\underline{v} = \lambda \underline{v}$ ein und bekommen für jeden Eigenwert ein System aus n Gleichungen mit n Unbekannten, nämlich den n Komponenten des jeweiligen Eigenvektors. Diese Gleichungssysteme sind unterbestimmt, weil wir nur die Richtung der Eigenvektoren, nicht aber deren Betrag, eindeutig bestimmen können. Ist \underline{v} ein Eigenvektor, so gilt dies für $\alpha \underline{v}$ ebenfalls. Oft werden die Eigenvektoren oder eine ihrer Komponenten daher auf 1 normiert.

1.2 Lösung der Zustandsgleichungen

Beispiel: Berechnen der Eigenwerte und Eigenvektoren einer 3 x 3 Matrix

$$\underline{A} = \begin{pmatrix} 6 & -3 & 4 \\ 0 & 7 & -1 \\ 0 & 24 & -3 \end{pmatrix}$$

Berechnen des charakteristischen Polynoms:

$$\begin{aligned} \det(\lambda \underline{I} - \underline{A}) &= \det \begin{pmatrix} \lambda - 6 & 3 & -4 \\ 0 & \lambda - 7 & 1 \\ 0 & -24 & \lambda + 3 \end{pmatrix} = (\lambda - 6)((\lambda - 7)(\lambda + 3) + 24) \\ &= (\lambda - 6)(\lambda^2 - 4\lambda + 3) = (\lambda - 6)(\lambda - 3)(\lambda - 1) = 0 \end{aligned}$$

Dies liefert die Eigenwerte:

$$\lambda_1 = 6$$

$$\lambda_2 = 3$$

$$\lambda_3 = 1$$

1.2 Lösung der Zustandsgleichungen

Berechnung des Eigenvektors zu $\lambda_1 = 6$:

$$\begin{pmatrix} 6 & -3 & 4 \\ 0 & 7 & -1 \\ 0 & 24 & -3 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = 6 \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad \rightarrow \quad \begin{aligned} 6v_1 - 3v_2 + 4v_3 &= 6v_1 \\ 7v_2 - v_3 &= 6v_2 \\ 24v_2 - 3v_3 &= 6v_3 \end{aligned}$$

Lösen wir die beiden ersten Gleichungen auf:

$$7v_2 - v_3 = 6v_2 \quad \rightarrow \quad v_2 = v_3$$

$$6v_1 - 3v_2 + 4v_3 = 6v_1 \quad \rightarrow \quad 6v_1 + v_2 = 6v_1 \quad \rightarrow \quad v_2 = 0 \quad \rightarrow \quad v_3 = 0$$

Daraus erhalten wir den 1. Eigenvektor:

$$\underline{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

1.2 Lösung der Zustandsgleichungen

Berechnung des Eigenvektors zu $\lambda_2 = 3$:

$$\begin{pmatrix} 6 & -3 & 4 \\ 0 & 7 & -1 \\ 0 & 24 & -3 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = 3 \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad \rightarrow \quad \begin{aligned} 6v_1 - 3v_2 + 4v_3 &= 3v_1 \\ 7v_2 - v_3 &= 3v_2 \\ 24v_2 - 3v_3 &= 3v_3 \end{aligned}$$

Die beiden letzten Gleichungen sind äquivalent. Wir brauchen also nur die beiden ersten:

$$7v_2 - v_3 = 3v_2 \quad \rightarrow \quad 4v_2 = v_3$$

$$6v_1 - 3v_2 + 4v_3 = 3v_1 \quad \rightarrow \quad 6v_1 + 13v_2 = 3v_1 \quad \rightarrow \quad v_1 = -\frac{13}{3}v_2$$

Daraus erhalten wir den 1. Eigenvektor:

$$\underline{v}_2 = \begin{pmatrix} -\frac{13}{3} \\ 1 \\ 4 \end{pmatrix}$$

1.2 Lösung der Zustandsgleichungen

Berechnung des Eigenvektors zu $\lambda_3 = 1$:

$$\begin{pmatrix} 6 & -3 & 4 \\ 0 & 7 & -1 \\ 0 & 24 & -3 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = 1 \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \quad \rightarrow \quad \begin{aligned} 6v_1 - 3v_2 + 4v_3 &= v_1 \\ 7v_2 - v_3 &= v_2 \\ 24v_2 - 3v_3 &= v_3 \end{aligned}$$

Die beiden letzten Gleichungen sind äquivalent. Wir brauchen also nur die beiden ersten:

$$7v_2 - v_3 = v_2 \quad \rightarrow \quad 6v_2 = v_3$$

$$6v_1 - 3v_2 + 4v_3 = v_1 \quad \rightarrow \quad 6v_1 + 21v_2 = v_1 \quad \rightarrow \quad v_1 = -\frac{21}{5}v_2$$

Daraus erhalten wir den 3. Eigenvektor:

$$\underline{v}_3 = \begin{pmatrix} -\frac{21}{5} \\ 1 \\ 6 \end{pmatrix}$$

1.2 Lösung der Zustandsgleichungen

Systemmatrix \underline{A} in Diagonalform

Eine sehr wichtige kanonische Form der Zustandsgleichungen liegt vor, wenn \underline{A} **Diagonalform** (auch **Modalform** genannt) aufweist. Die Systemmatrix hat dann folgende Struktur:

$$\underline{A} = \text{diag}\{a_{ii}\} = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{pmatrix}$$

Es wird sehr selten vorkommen, dass \underline{A} gleich nach der Modellbildung Diagonalform hat. Vielmehr wird es meist nötig sein, die Zustandsgleichungen entsprechend zu transformieren. Wie das geht behandelt Kapitel 1.3.

Bemerkung: Nur wenn \underline{A} ausschließlich reelle und verschiedene Eigenwerte hat, ist es sichergestellt, dass sich die Zustandsgleichungen auf Diagonalform transformieren lassen. Andernfalls müssen zusätzliche Bedingungen eingehalten werden bzw. es ist nur eine Transformation auf "näherungsweise" Diagonalformen möglich.

1.2 Lösung der Zustandsgleichungen

Wichtige Eigenschaften der Diagonalform

Die Diagonalelemente a_{ii} von \underline{A} sind die Eigenwerte λ_i von \underline{A} und die Eigenvektoren sind die Einheitsvektoren:

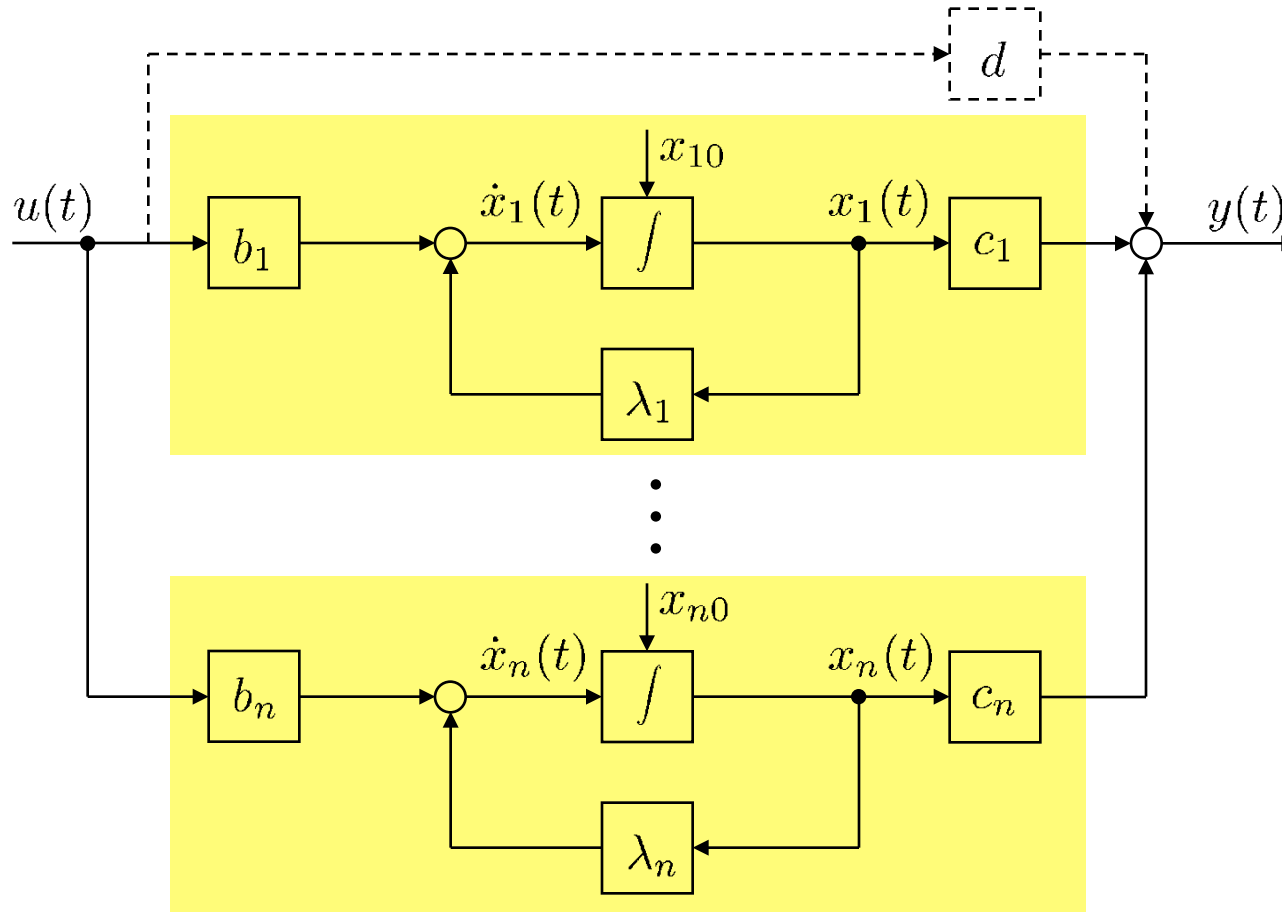
$$\underline{A} = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix} \quad \underline{v}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \underline{v}_2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \quad \dots, \quad \underline{v}_n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

Das lässt sich leicht zeigen:

$$\underline{A} \underline{v}_1 = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} \lambda_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \lambda_1 \underline{v}_1$$
$$\underline{A} \underline{v}_2 = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \lambda_2 \\ \vdots \\ 0 \end{pmatrix} = \lambda_2 \underline{v}_2 \quad \dots$$

1.2 Lösung der Zustandsgleichungen

Das bedeutet, die n Zustandsgleichungen sind *entkoppelt*. Kein Zustand beeinflusst einen anderen; n PT₁-Systeme laufen parallel nebeneinander her (das erklärt auch, warum das bei konjugiert komplexen Polen nicht einfach genauso funktionieren kann).



1.2 Lösung der Zustandsgleichungen

Die Diagonalform bietet also den Vorteil, dass man die Eigenwerte (Pole) des Systems sofort aus \underline{A} ablesen kann. Die Diagonalform entspricht damit der *Partialbruchzerlegung* der assoziierten Übertragungsfunktion.

Wie vereinfacht sich die Zeitlösung der Zustandsdifferentialgleichungen, wenn \underline{A} Diagonalgestalt aufweist?

Die Berechnung der Matrixexponentialfunktion vereinfacht sich erheblich, denn es gilt:

$$\underline{A}^i = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix}^i = \begin{pmatrix} \lambda_1^i & 0 & \cdots & 0 \\ 0 & \lambda_2^i & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n^i \end{pmatrix}$$
$$\rightarrow e^{\underline{A}t} = e^{\text{diag}\{\lambda_i t\}} = \begin{pmatrix} e^{\lambda_1 t} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2 t} & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & e^{\lambda_n t} \end{pmatrix} = \text{diag}\{e^{\lambda_i t}\}$$

1.2 Lösung der Zustandsgleichungen

Die Zeitlösung der Zustandsdifferentialgleichungen ist allgemein:

$$\underline{x}(t) = e^{\underline{A}t} \underline{x}_0 + \int_0^t e^{\underline{A}(t-\tau)} \underline{b} u(\tau) d\tau$$

Mit einer Diagonalmatrix \underline{A} vereinfacht sich das zu:

$$\underline{x}(t) = \begin{pmatrix} e^{\lambda_1 t} x_{10} \\ e^{\lambda_2 t} x_{20} \\ \vdots \\ e^{\lambda_n t} x_{n0} \end{pmatrix} + \begin{pmatrix} \int_0^t e^{\lambda_1(t-\tau)} b_1 u(\tau) d\tau \\ \int_0^t e^{\lambda_2(t-\tau)} b_2 u(\tau) d\tau \\ \vdots \\ \int_0^t e^{\lambda_n(t-\tau)} b_n u(\tau) d\tau \end{pmatrix}$$

Die Funktionen $e^{\lambda_i t}$ werden auch **Modi** genannt. Daher kommt auch der Name *Modalform*.

Sobald die Zustandsgleichungen in Diagonalform transformiert worden sind, lassen sie sich also leicht als n voneinander unabhängige DGLs 1. Ordnung lösen. Für eine solche Transformation müssen wir die Eigenwerte und -vektoren von \underline{A} ausrechnen (s. Kapitel 1.3)!

1.2 Lösung der Zustandsgleichungen

Diagonalform bei komplexen Eigenwerten

Treten komplexe Eigenwerte auf (und damit auch komplexe Eigenvektoren), so ist auch die Diagonalform komplex. Eine Diagonalmatrix mit den Eigenwerten λ , $\alpha + i\beta$, $\alpha - i\beta$ sieht z.B. so aus:

$$\begin{pmatrix} \lambda & 0 & 0 \\ 0 & \alpha + i\beta & 0 \\ 0 & 0 & \alpha - i\beta \end{pmatrix} \quad (\text{alle Zustände sind entkoppelt})$$

Da alle Signale und Parameter des Systems reell sind, rechnet man lieber mit rein reellen Größen. Dies kann erreicht werden, indem obige Matrix in folgende Form transformiert wird:

$$\begin{pmatrix} \lambda & 0 & 0 \\ 0 & \alpha & \beta \\ 0 & -\beta & \alpha \end{pmatrix} \quad (\text{Zustände } x_2 \text{ und } x_3 \text{ sind gekoppelt})$$

Die zugehörigen Eigenvektoren sind nun auch reell.

Generell können so immer 2 entkoppelte (Diagonalform!) DGLs mit konjugiert komplexen Parametern zu 2 gekoppelten DGLs mit reellen Parametern (PT₂) zusammengefasst werden.

1.2 Lösung der Zustandsgleichungen

Jordanform bei mehrfachen Eigenwerten

Treten mehrfache Eigenwerte auf, *kann* es sein, dass sich die Systemmatrix nicht mehr in eine reine Diagonalform bringen lässt. Z.B. kann eine Matrix mit einem doppelten Eigenwert bei λ_1 und einen einfachen bei λ_2 in eine der folgenden Formen gebracht werden:

$$\begin{pmatrix} \boxed{\lambda_1} & 0 & 0 \\ 0 & \boxed{\lambda_1} & 0 \\ 0 & 0 & \boxed{\lambda_2} \end{pmatrix} \quad \text{oder} \quad \begin{pmatrix} \boxed{\lambda_1} & 1 & 0 \\ 0 & \boxed{\lambda_1} & 0 \\ 0 & 0 & \boxed{\lambda_2} \end{pmatrix}$$

Für eine Matrix mit einem dreifachen Eigenwert bei λ_1 und einen einfachen bei λ_2 gilt eine der folgenden Formen:

$$\begin{pmatrix} \boxed{\lambda_1} & 0 & 0 & 0 \\ 0 & \boxed{\lambda_1} & 0 & 0 \\ 0 & 0 & \boxed{\lambda_1} & 0 \\ 0 & 0 & 0 & \boxed{\lambda_2} \end{pmatrix} \quad \text{oder} \quad \begin{pmatrix} \boxed{\lambda_1} & 1 & 0 & 0 \\ 0 & \boxed{\lambda_1} & 0 & 0 \\ 0 & 0 & \boxed{\lambda_1} & 0 \\ 0 & 0 & 0 & \boxed{\lambda_2} \end{pmatrix} \quad \text{oder} \quad \begin{pmatrix} \boxed{\lambda_1} & 1 & 0 & 0 \\ 0 & \boxed{\lambda_1} & 1 & 0 \\ 0 & 0 & \boxed{\lambda_1} & 0 \\ 0 & 0 & 0 & \boxed{\lambda_2} \end{pmatrix}$$

1er-Jordanblock

Eine Matrix, die mindestens einen Jordanblock der Größe 2 (oder mehr) enthält, liegt in der **Jordanform** vor. Ein Jordanblock enthält "Einsen" auf der ersten Nebendiagonale.

1.3 Eigenschaften der Zustandsgleichungen

Bevor wir zum Reglerentwurf kommen können, benötigen wir noch einige grundlegende Eigenschaften der Zustandsgleichungen.

Theorem von Cayley-Hamilton

$P(s)$ sei das charakteristische Polynom einer Matrix \underline{A} , dann lautet die charakteristische Gleichung:

$$P(s) = s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0 = 0$$

Setzt man in diese (eigentlich skalare) Gleichung die Matrix \underline{A} ein, so erfüllt sie selbst ihre eigene charakteristische Gleichung (mit $\underline{I} = \underline{A}^0$):

$$P(\underline{A}) = \underline{A}^n + a_{n-1}\underline{A}^{n-1} + \dots + a_1\underline{A} + a_0\underline{I} = \underline{0}$$

D.h. die Matrix \underline{A}^n (und alle höheren Potenzen von \underline{A}) sind linear abhängig von den Matrizen \underline{A}^i mit $i < n$:

$$\underline{A}^n = -a_{n-1}\underline{A}^{n-1} - \dots - a_1\underline{A} - a_0\underline{I}$$

1.3 Eigenschaften der Zustandsgleichungen

Ljapunow-Gleichung

In der Regelungstechnik spielen **Ljapunow-Gleichungen** an vielerlei Stellen eine wichtige Rolle. Deshalb wollen wir sie hier kurz vorstellen.

Die $n \times n$ -Matrizen \underline{A} , \underline{B} , und \underline{C} seien gegeben. Gesucht ist die Matrix \underline{M} , welche folgende Gleichung erfüllt:

$$\underline{A} \underline{M} + \underline{M} \underline{B} = \underline{C}$$

Diese Gleichung enthält n^2 Unbekannte, nämlich die Elemente der $n \times n$ -Matrix \underline{M} . Nach dem Ausmultiplizieren der Matrizen entsteht ein *lineares Gleichungssystem* mit n^2 Gleichungen, das leicht nach diesen Unbekannten aufgelöst werden kann.

Man kann die Ljapunow-Gleichung sogar noch etwas allgemeiner formulieren, indem man unterschiedliche Dimensionen für \underline{A} und \underline{B} zulässt:

$$\underline{A}: n \times n, \underline{B}: m \times m \rightarrow \underline{C} \text{ und } \underline{M}: n \times m.$$

Mit MATLAB können solche Ljapunow-Gleichungen mittels $\mathbf{M} = \mathbf{lyap}(\mathbf{A}, \mathbf{B}, -\mathbf{C})$ gelöst werden (im Englischen wird der russische Name mit "Lyapunov" übersetzt!).



1.3 Eigenschaften der Zustandsgleichungen

Beispiel: Lösen einer 2 x 2-Ljapunow-Gleichung

$$\begin{aligned} \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix} &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} + \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \\ &= \begin{pmatrix} a_{11}m_{11} + a_{12}m_{21} & a_{11}m_{12} + a_{12}m_{22} \\ a_{21}m_{11} + a_{22}m_{21} & a_{21}m_{12} + a_{22}m_{22} \end{pmatrix} + \begin{pmatrix} m_{11}b_{11} + m_{12}b_{21} & m_{11}b_{12} + m_{12}b_{22} \\ m_{21}b_{11} + m_{22}b_{21} & m_{21}b_{12} + m_{22}b_{22} \end{pmatrix} \\ \underbrace{\begin{pmatrix} c_{11} \\ c_{12} \\ c_{21} \\ c_{22} \end{pmatrix}}_{\underline{c}} &= \underbrace{\begin{pmatrix} a_{11} + b_{11} & b_{21} & a_{12} & 0 \\ b_{12} & a_{11} + b_{22} & 0 & a_{12} \\ a_{21} & 0 & a_{22} + b_{11} & b_{21} \\ 0 & a_{21} & b_{12} & a_{22} + b_{22} \end{pmatrix}}_{\underline{L}} \underbrace{\begin{pmatrix} m_{11} \\ m_{12} \\ m_{21} \\ m_{22} \end{pmatrix}}_{\underline{m}} \end{aligned}$$

$$\underline{c} = \underline{L} \cdot \underline{m} \rightarrow \underline{m} = \underline{L}^{-1} \cdot \underline{c}$$

Dieses Gleichungssystem mit 4 (2 x 2) Gleichungen und 4 Unbekannten ($m_{11}, m_{12}, m_{21}, m_{22}$) kann leicht gelöst werden!

1.3 Eigenschaften der Zustandsgleichungen

Steuerbarkeit

Ein System heißt *steuerbar*, wenn es in endlicher Zeit t_e von jedem beliebigen Anfangszustand \underline{x}_0 durch einen geeignet gewählten Verlauf der Eingangsgröße in jeden beliebigen Endzustand $\underline{x}(t_e)$ überführt werden kann.

Der Zusammenhang zwischen dem Zustand \underline{x} und der Ausgangsgröße y spielt hierbei gar keine Rolle. Es ist nur von Belang, ob man mit dem Eingang u alle Zustände \underline{x} beliebig beeinflussen kann. Deshalb ist klar, dass die Steuerbarkeit alleine anhand von \underline{A} und \underline{b} nachgeprüft werden kann.

Der Endzustand $\underline{x}(t_e)$ berechnet sich aus dem Anfangszustand \underline{x}_0 , wie wir schon wissen, mit:

$$\underline{x}(t_e) = e^{\underline{A}t_e} \underline{x}_0 + \int_0^{t_e} e^{\underline{A}(t_e-\tau)} \underline{b} u(\tau) d\tau$$

Wählen wir z.B. $\underline{x}(t_e) = \underline{0}$ so erhalten wir:

$$-e^{\underline{A}t_e} \underline{x}_0 = \int_0^{t_e} e^{\underline{A}(t_e-\tau)} \underline{b} u(\tau) d\tau = e^{\underline{A}t_e} \int_0^{t_e} e^{-\underline{A}\tau} \underline{b} u(\tau) d\tau$$

1.3 Eigenschaften der Zustandsgleichungen

Kürzen führt auf:

$$-\underline{x}_0 = \int_0^{t_e} e^{-\underline{A}\tau} \underline{b} u(\tau) d\tau$$

Einsetzen der Reihenentwicklung der Matrixexponentialfunktion liefert:

$$-\underline{x}_0 = \underbrace{\underline{b} \int_0^{t_e} u(\tau) d\tau}_{u_0} - \underbrace{\underline{A} \underline{b} \int_0^{t_e} \tau u(\tau) d\tau}_{-u_1} + \underbrace{\underline{A}^2 \underline{b} \int_0^{t_e} \frac{\tau^2}{2!} u(\tau) d\tau}_{u_2} - \underbrace{\underline{A}^3 \underline{b} \int_0^{t_e} \frac{\tau^3}{3!} u(\tau) d\tau}_{-u_3} + \dots$$

$$-\underline{x}_0 = \underline{b} u_0 + \underline{A} \underline{b} u_1 + \underline{A}^2 \underline{b} u_2 + \underline{A}^3 \underline{b} u_3 + \dots$$

D.h. \underline{x}_0 setzt sich aus (unendlich vielen) n -dimensionalen Spaltenvektoren zusammen.

Jeder beliebige Anfangszustand \underline{x}_0 kann nur dann durch einen geeigneten Verlauf der Eingangsgröße erzeugt werden, wenn die Vektoren auf der rechten Seite den gesamten n -dimensionalen Raum aufspannen. Das bedeutet, folgende Matrix (mit unendlich vielen Spalten) muss Rang n aufweisen:

$$(\underline{b} \quad \underline{A} \underline{b} \quad \underline{A}^2 \underline{b} \quad \underline{A}^3 \underline{b} \quad \dots)$$

1.3 Eigenschaften der Zustandsgleichungen

Aus dem Theorem von *Cayley-Hamilton* wissen wir, dass \underline{A}^n linear abhängig von den niedrigeren Potenzen von \underline{A} ist, d.h. \underline{A}^n lässt sich als Linearkombination von \underline{I} , \underline{A} , \underline{A}^2 , ... \underline{A}^{n-1} schreiben. Gleiches gilt für alle höheren Potenzen von \underline{A} . Daher tragen nur die ersten n Spalten obiger Matrix zu ihrem Rang bei. Es reicht also aus zu prüfen, ob die Matrix, bestehend aus den ersten n Spalten, Rang n hat.

Steuerbarkeitskriterium

Folgende $n \times n$ -Matrix heißt **Steuerbarkeitsmatrix**:

$$\underline{S}_S = (\underline{b} \quad \underline{A}\underline{b} \quad \underline{A}^2\underline{b} \quad \dots \quad \underline{A}^{n-1}\underline{b})$$

Wenn die Steuerbarkeitsmatrix vollen Rang hat, dann ist das System $(\underline{A}, \underline{b})$ steuerbar:

$$\text{Rang } \underline{S}_S = n$$

Bemerkung: Bei der numerischen Berechnung der Steuerbarkeit ist es wegen begrenzter Rechengenauigkeit notwendig eine Toleranzgrenze anzugeben, um zu definieren, ab wo ein Rangabfall (Singularität) festgestellt wird. Bei schlechter Konditionierung von \underline{S}_S ist es durchaus sinnvoll, von "*schlecht steuerbar*" oder "*schwer steuerbar*" o.ä. zu sprechen.

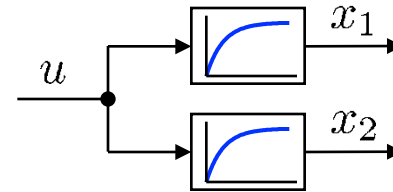
1.3 Eigenschaften der Zustandsgleichungen

Welche Art von Systemen sind nicht steuerbar?

- Eigenvorgänge, die nicht mit dem Eingang verbunden sind, sind nicht steuerbar. Das lässt sich leicht in der Diagonalform erkennen. Ist dort ein $b_i = 0$, dann kann der Zustand x_i nicht vom Eingang beeinflusst werden. Dieser Zustand und damit das System sind also nicht steuerbar.
- 2 parallele Teilsysteme mit denselben dynamischen Eigenschaften sind nicht steuerbar.

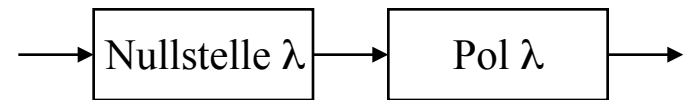
Beispiel:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} u$$



Da beide Teilsysteme identische Dynamik haben, lassen sich die beiden Zustände nur linear abhängig beeinflussen. Es wird z.B. immer gelten: $x_1 > 0 \rightarrow x_2 > 0$ (falls b_1 und b_2 gleiche Vorzeichen haben) bzw. $x_1 > 0 \rightarrow x_2 < 0$ (falls b_1 und b_2 unterschiedliche Vorzeichen haben).

- Jede Nullstellen/Pol-Kürzung (in dieser Reihenfolge) macht je einen Zustand nicht steuerbar.



1.3 Eigenschaften der Zustandsgleichungen

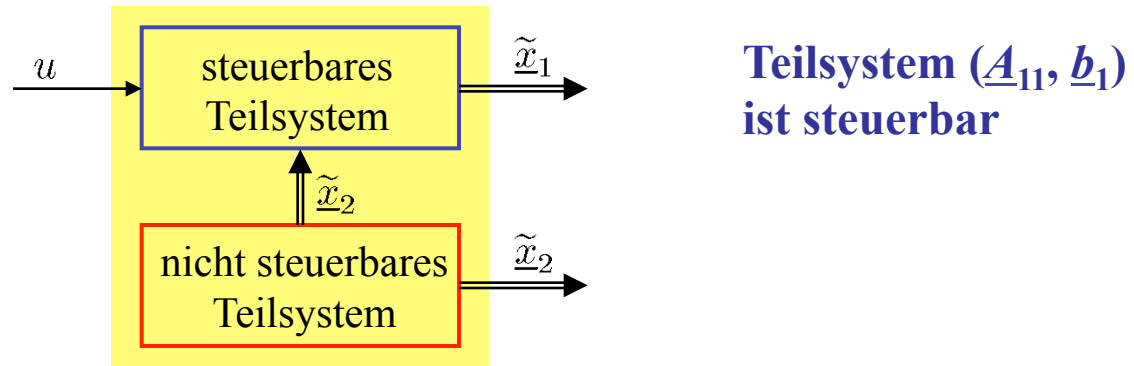
Zerlegung eines Systems in einen steuerbaren und einen nicht steuerbaren Teil

Jedes System lässt sich in einen steuerbaren und nicht steuerbaren Teil zerlegen.

Originalsystem: $\dot{\underline{x}} = \underline{A} \underline{x} + \underline{b} u$
 $y = \underline{c}^T \underline{x} + d u$

Zerlegtes System: $\begin{pmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \end{pmatrix} = \begin{pmatrix} \underline{A}_{11} & \underline{A}_{12} \\ 0 & \underline{A}_{22} \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} + \begin{pmatrix} \underline{b}_1 \\ \underline{0} \end{pmatrix} u$
 $y = (\underline{c}_1^T \quad \underline{c}_2^T) \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} + d u$

steuerbar
nicht steuerbar



Bemerkung: Ist das Originalsystem schon steuerbar, so gilt: $\tilde{x}_1 = \underline{x}$ und $\tilde{x}_2 = \text{leer}$.

1.3 Eigenschaften der Zustandsgleichungen

Beobachtbarkeit

Ein System heißt *beobachtbar*, wenn der Anfangszustand \underline{x}_0 aus dem über einem endlichen Intervall $[0, t_e]$ bekannten Verlauf der Eingangsgröße und Ausgangsgröße bestimmt werden kann.

Zur Bestimmung der Beobachtbarkeit ist es nur wichtig zu wissen, wie der Zustand \underline{x} auf den Ausgang y wirkt. Deshalb ist klar, dass die Beobachtbarkeit alleine anhand von \underline{A} und \underline{c}^T nachgeprüft werden kann. (Die Kenntnis des Eingangsverlauf ist zwar notwendig um den Anfangszustand zu rekonstruieren; dies ist aber mit jedem beliebigen \underline{b} möglich, so dass in \underline{b} keine Information über die Beobachtbarkeit steckt!)

Beobachtbarkeitskriterium

Folgende $n \times n$ -Matrix heißt **Beobachtbarkeitsmatrix**: $\underline{S}_B = \begin{pmatrix} \underline{c}^T \\ \underline{c}^T \underline{A} \\ \underline{c}^T \underline{A}^2 \\ \vdots \\ \underline{c}^T \underline{A}^{n-1} \end{pmatrix}$

Wenn die Beobachtbarkeitsmatrix vollen Rang hat, dann ist das System $(\underline{A}, \underline{c}^T)$ beobachtbar: $\text{Rang } \underline{S}_B = n$

1.3 Eigenschaften der Zustandsgleichungen

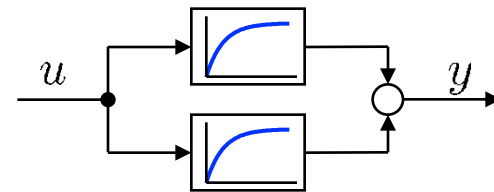
Welche Art von Systemen sind nicht beobachtbar?

- Eigenvorgänge, die nicht mit dem Ausgang verbunden sind, sind nicht beobachtbar. Das lässt sich leicht in der Diagonalform erkennen. Ist dort ein $c_i = 0$, dann kann der Zustand x_i nicht den Ausgang beeinflussen. Dieser Zustand und damit das System sind also nicht beobachtbar.
- 2 parallele Teilsysteme mit denselben dynamischen Eigenschaften sind nicht beobachtbar.

Beispiel:

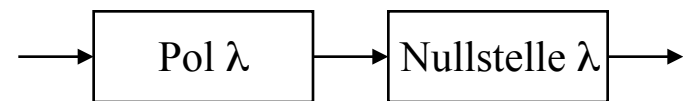
$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} u$$

$$y = (c_1 \quad c_2) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



Zwar beeinflussen beide Zustände den Ausgang, aber deren Einfluss lässt sich nicht mehr trennen, da beide Teilsysteme dieselbe Dynamik aufweisen. Es lässt sich im Nachhinein also nicht mehr bestimmen, welcher Anteil von welchem Zustand stammt.

- Jede Pol/Nullstellen-Kürzung (in dieser Reihenfolge) macht je einen Zustand nicht beobachtbar.



1.3 Eigenschaften der Zustandsgleichungen

Zerlegung eines Systems in einen beobachtbaren und einen nicht beobachtbaren Teil

Jedes System lässt sich in einen beobachtbaren und nicht beobachtbaren Teil zerlegen.

Originalsystem: $\dot{\underline{x}} = \underline{A} \underline{x} + \underline{b} u$

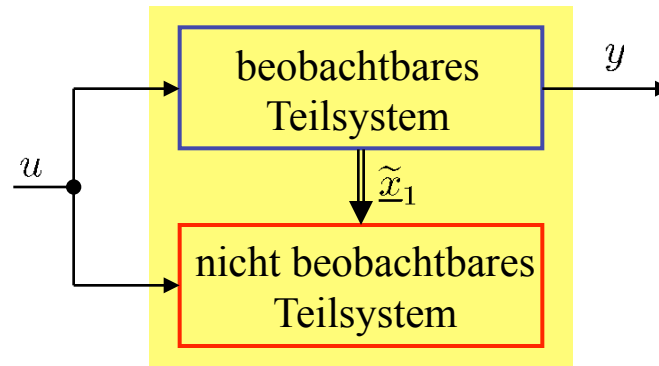
$$y = \underline{c}^T \underline{x} + d u$$

Zerlegtes System: $\begin{pmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \end{pmatrix} = \begin{pmatrix} \underline{A}_{11} & 0 \\ \underline{A}_{21} & \underline{A}_{22} \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} + \begin{pmatrix} \underline{b}_1 \\ \underline{b}_2 \end{pmatrix} u$

$$y = (\underline{c}_1^T \quad \underline{0}) \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} + d u$$

beobachtbar

nicht beobachtbar



Teilsystem $(\underline{A}_{11}, \underline{c}_1^T)$ ist beobachtbar

Bemerkung: Ist das Originalsystem schon beobachtbar, so gilt: $\tilde{x}_1 = \underline{x}$ und $\tilde{x}_2 = \text{leer}$.

1.3 Eigenschaften der Zustandsgleichungen

Kalman-Zerlegung

Wenn wir die Zerlegungen bezüglich Steuerbarkeit und Beobachtbarkeit zusammenfassen, können wir ein System in insgesamt 4 Teile zerlegen:

1. Teil: steuerbar und beobachtbar

1. Teil: steuerbar und nicht beobachtbar

3. Teil: nicht steuerbar und beobachtbar

2. Teil: nicht steuerbar und nicht beobachtbar

$$\begin{pmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \\ \dot{\tilde{x}}_3 \\ \dot{\tilde{x}}_4 \end{pmatrix} = \begin{pmatrix} \underline{A}_{11} & \underline{0} & \underline{A}_{13} & \underline{0} \\ \underline{A}_{21} & \underline{A}_{22} & \underline{A}_{23} & \underline{A}_{24} \\ \underline{0} & \underline{0} & \underline{A}_{33} & \underline{0} \\ \underline{0} & \underline{0} & \underline{A}_{43} & \underline{A}_{44} \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \end{pmatrix} + \begin{pmatrix} \underline{b}_1 \\ \underline{b}_2 \\ \underline{0} \\ \underline{0} \end{pmatrix} u$$

$$y = \begin{pmatrix} \underline{c}_1^T & \underline{0}^T & \underline{c}_3^T & \underline{0}^T \end{pmatrix} \begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \\ \tilde{x}_4 \end{pmatrix} + d u$$

1.3 Eigenschaften der Zustandsgleichungen

Dualität zwischen Steuerbarkeit und Beobachtbarkeit

Die Kriterien zur Überprüfung der Steuerbarkeit und Beobachtbarkeit sind sehr eng miteinander verwandt. Wir stellen fest, dass das Steuerbarkeitskriterium in das Beobachtbarkeitskriterium übergeht und umgekehrt, wenn

$$\underline{A} \leftrightarrow \underline{A}^T \quad \text{und} \quad \underline{b} \leftrightarrow \underline{c}.$$

Deshalb sagt man, die beiden Eigenschaften (Steuerbarkeit und Beobachtbarkeit) sind **dual** zueinander. Man kann neben dem "normalen" (primären) System folgendes duale System betrachten:

Primäres System:

$$\dot{\underline{x}}(t) = \underline{A} \underline{x}(t) + \underline{b} u(t)$$

$$y(t) = \underline{c}^T \underline{x}(t) + d u(t)$$

Duales System:

$$\dot{\underline{x}}_D(t) = \underline{A}^T \underline{x}_D(t) + \underline{c} u_D(t)$$

$$y_D(t) = \underline{b}^T \underline{x}_D(t) + d u_D(t)$$

Wenn das *primäre* System $(\underline{A}, \underline{b}, \underline{c}^T)$ *steuerbar* ist, dann ist das *duale* System $(\underline{A}^T, \underline{c}, \underline{b}^T)$ *beobachtbar*.

Wenn das *primäre* System $(\underline{A}, \underline{b}, \underline{c}^T)$ *beobachtbar* ist, dann ist das *duale* System $(\underline{A}^T, \underline{c}, \underline{b}^T)$ *steuerbar*.

1.3 Eigenschaften der Zustandsgleichungen

Transformation der Zustandsgleichungen

Wir wissen bereits, dass es im Zustandsraum unendlich viele mögliche Realisationen einer Übertragungsfunktion gibt, selbst wenn wir uns auf die minimalen Realisationen beschränken, d.h. diejenigen, die nur die minimale Anzahl von Zuständen enthalten. Die verschiedenen Realisation unterscheiden sich nur in der Wahl des Zustandsvektors, sie sind identisch bezüglich ihres Ein-/Ausgangsverhaltens.

Wie kann man Zustandsgleichungen in andere Realisationen transformieren?

Es wird zunächst eine reguläre (d.h. sie hat vollen Rang) $n \times n$ -Transformationsmatrix \underline{T} gewählt. Zwischen dem alten Zustandsvektor \underline{x} und dem neuen (transformierten) Zustandsvektor \underline{x}_T wird so umgerechnet:

$$\underline{x}(t) = \underline{T} \underline{x}_T(t)$$

$$\underline{x}_T(t) = \underline{T}^{-1} \underline{x}(t)$$

Wie die Transformationsmatrix \underline{T} genau gewählt werden muss, um in die wichtigsten kanonischen Formen transformieren zu können, wird weiter hinten behandelt.

1.3 Eigenschaften der Zustandsgleichungen

Mit dieser Transformation ergeben sich folgende neue Zustandsgleichungen:

$$\begin{aligned} \underline{T} \dot{\underline{x}}_T(t) &= \underline{A} \underline{T} \underline{x}_T(t) + \underline{b} u(t) & \rightarrow & \quad \dot{\underline{x}}_T(t) = \underline{T}^{-1} \underline{A} \underline{T} \underline{x}_T(t) + \underline{T}^{-1} \underline{b} u(t) \\ y(t) &= \underline{c}^T \underline{T} \underline{x}_T(t) + d u(t) & & \quad y(t) = \underline{c}^T \underline{T} \underline{x}_T(t) + d u(t) \end{aligned}$$

Mit der Definition einer neuen (transformierten) Systemmatrix und Eingangs- und Ausgangsvektoren ergibt sich für die transformierten Zustandsgleichungen:

$$\begin{aligned} \dot{\underline{x}}_T(t) &= \underline{A}_T \underline{x}_T(t) + \underline{b}_T u(t) \\ y(t) &= \underline{c}_T^T \underline{x}_T(t) + d_T u(t) \end{aligned}$$

mit

$$\underline{A}_T = \underline{T}^{-1} \underline{A} \underline{T}$$

$$\underline{b}_T = \underline{T}^{-1} \underline{b}$$

$$\underline{c}_T^T = \underline{c}^T \underline{T}$$

$$d_T = d$$

1.3 Eigenschaften der Zustandsgleichungen

Transformation in die Diagonalform

Um die Zustandsgleichungen in die Diagonalform zu transformieren, muss die Transformationsmatrix aus den Eigenvektoren \underline{v}_i der Systemmatrix \underline{A} aufgebaut sein:

$$\underline{T} = (\underline{v}_1 \quad \underline{v}_2 \quad \cdots \quad \underline{v}_n)$$

Dies lässt sich leicht zeigen, indem man die transformierte Systemmatrix ausrechnet:

$$\underline{A}_T = \underline{T}^{-1} \underline{A} \underline{T} = \underline{T}^{-1} \underline{A} (\underline{v}_1 \quad \underline{v}_2 \quad \cdots \quad \underline{v}_n) = \underline{T}^{-1} (\underline{A} \underline{v}_1 \quad \underline{A} \underline{v}_2 \quad \cdots \quad \underline{A} \underline{v}_n)$$

Da \underline{v}_i die Eigenvektoren von \underline{A} sind gilt $\underline{A} \underline{v}_i = \lambda_i \underline{v}_i$ und damit:

$$\underline{A}_T = \underline{T}^{-1} (\lambda_1 \underline{v}_1 \quad \lambda_2 \underline{v}_2 \quad \cdots \quad \lambda_n \underline{v}_n) = \underline{T}^{-1} \underline{T} \text{diag } \lambda_i = \text{diag } \lambda_i = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_n \end{pmatrix}$$

1.3 Eigenschaften der Zustandsgleichungen

Transformation in die Regelungsnormalform

Um die Zustandsgleichungen in die Regelungsnormalform zu transformieren, muss eine Transformationsmatrix aufgebaut werden, die von der Inversen der Steuerbarkeitsmatrix \underline{S}_S abhängt. Diese Inverse existiert nur, wenn \underline{S}_S vollen Rang hat, d.h. das System steuerbar ist. Für nicht steuerbare Systeme existiert die Regelungsnormalform nicht.

Die Steuerbarkeitsmatrix ist wie folgt definiert:

$$\underline{S}_S = (\underline{b} \quad \underline{A}\underline{b} \quad \underline{A}^2\underline{b} \quad \dots \quad \underline{A}^{n-1}\underline{b})$$

Von ihrer Inversen wird die letzte Zeile benötigt:

$$\underline{s}_S^T = (0 \quad \dots \quad 0 \quad 1) \underline{S}_S^{-1}$$

Daraus berechnet sich die Transformationsmatrix zu:

$$\underline{T} = \begin{pmatrix} \underline{s}_S^T \\ \underline{s}_S^T \underline{A} \\ \underline{s}_S^T \underline{A}^2 \\ \vdots \\ \underline{s}_S^T \underline{A}^{n-1} \end{pmatrix}^{-1}$$

1.3 Eigenschaften der Zustandsgleichungen

Transformation in die Beobachtungsnormalform

Um die Zustandsgleichungen in die Beobachtungsnormalform zu transformieren, muss eine Transformationsmatrix aufgebaut werden, die von der Inversen der Beobachtbarkeitsmatrix \underline{S}_B abhängt. Diese Inverse existiert nur, wenn \underline{S}_B vollen Rang hat, d.h. das System beobachtbar ist. Für nicht beobachtbare Systeme existiert die Beobachtungsnormalform nicht.

Die Beobachtbarkeitsmatrix ist wie folgt definiert:

$$\underline{S}_B = \begin{pmatrix} \underline{c}^T \\ \underline{c}^T \underline{A} \\ \underline{c}^T \underline{A}^2 \\ \vdots \\ \underline{c}^T \underline{A}^{n-1} \end{pmatrix}$$

Von ihrer Inversen wird die letzte Spalte benötigt:

$$\underline{s}_B = \underline{S}_B^{-1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

Daraus berechnet sich die Transformationsmatrix zu:

$$\underline{T} = (\underline{s}_B \quad \underline{A} \underline{s}_B \quad \underline{A}^2 \underline{s}_B \quad \cdots \quad \underline{A}^{n-1} \underline{s}_B)$$

1.3 Eigenschaften der Zustandsgleichungen

Die Beobachtungsnormalform kann wie die Regelungsnormalform direkt aus den Zähler- und Nennerkoeffizienten der zugehörigen Übertragungsfunktion bestimmt werden. Für sprungfähige Systeme ergibt sie sich aus folgenden Gleichungen:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{n-1}(t) \\ \dot{x}_n(t) \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{pmatrix} + \begin{pmatrix} b_0 - b_n a_0 \\ b_1 - b_n a_1 \\ b_2 - b_n a_2 \\ \vdots \\ b_{n-1} - b_n a_{n-1} \end{pmatrix} u(t)$$

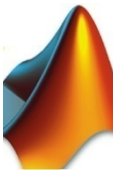
$$y(t) = (0 \quad 0 \quad \cdots \quad 0 \quad 1) \underline{x}(t) + b_n u(t)$$

Für nicht sprungfähige Systeme vereinfachen sich die Gleichungen, weil $b_n = 0$ ist. Der Eingangsvektor \underline{b} hat dann nur noch die Elemente b_i ($i = 0, \dots, n-1$) und der Durchgriff fällt weg.

Es existiert eine **Dualität** zwischen der Regelungs- und der Beobachtungsnormalform. Es gilt (die Indizes R und B stehen für die jeweiligen Normalformen):

$$\underline{A}_R = \underline{A}_B^T \quad \underline{b}_R = \underline{c}_B \quad \underline{c}_R^T = \underline{b}_B^T$$

1.3 Eigenschaften der Zustandsgleichungen



Berechnung von Eigenvektoren und Eigenwerten:

```
A = diag([2 3 4]);           % Erzeuge eine Diagonalmatrix
[EV, EW] = eig(A);          % Berechnet die Eigenvektoren und -werte von A.
                             % Die Eigenvektoren stehen in den Spalten der
                             % Matrix EV. Die Eigenwerte stehen in der
                             % Diagonale der Matrix EW.
```

Berechnung von Steuerbarkeit und Beobachtbarkeit:

```
System = ss(A,b,c,d);      % Erzeugt Zustandsgleichungen
S_S = ctrb(System);        % Erzeugt die Steuerbarkeitsmatrix
S_B = obsv(System);       % Erzeugt die Beobachtbarkeitsmatrix
r = rank(S_S);            % Berechnet den Rang einer Matrix
```

Transformation in Diagonal- oder Beobachtungsnormalform:

```
[System_T,T] = canon(System,'modal'); % Transformiert in die
                                       % Diagonalform (=Modalform).
                                       % T = Transformationsmatrix
[System_T,T] = canon(System,'companion'); % Transformiert in die
                                       % Beobachtungsnormalform
```

1.4 Zustandsregler durch Polvorgabe

Grundidee eines *Zustandsreglers* ist es, die *Zustände* – also *innere Informationen* aus dem Prozess – *rückzukoppeln*. Dies steht im Gegensatz zum Standardregelkreis, wo nur die Regelgröße – also der Ausgang des Prozesses – rückgekoppelt wird. Damit kann der Zustandsregler als eine Art systematische Erweiterung des Gedankens der Kaskadenregelung verstanden werden.

Vorteile des Zustandsreglers sind:

- Kleine Abweichungen werden sehr schnell erkannt und korrigiert, weil viel Information aus dem Inneren des Prozesses verwendet wird (gleicher Gedanke wie bei der Kaskadenregelung).
- Es existieren sehr elegante und ausgereifte Verfahren für den Reglerentwurf.
- Die erreichbare Regelgüte ist meist sehr hoch.

Nachteile des Zustandsreglers sind:

- Zustandsraumverfahren benötigen gute Modelle und Rechnerunterstützung.
- Einfache Interpretierbarkeit der Ergebnisse (z.B. der Reglerparameter) nicht gegeben.
- Fehlende Erfahrung von Anlagenpersonal; keine direkte Verbindung zum PID-Regler.

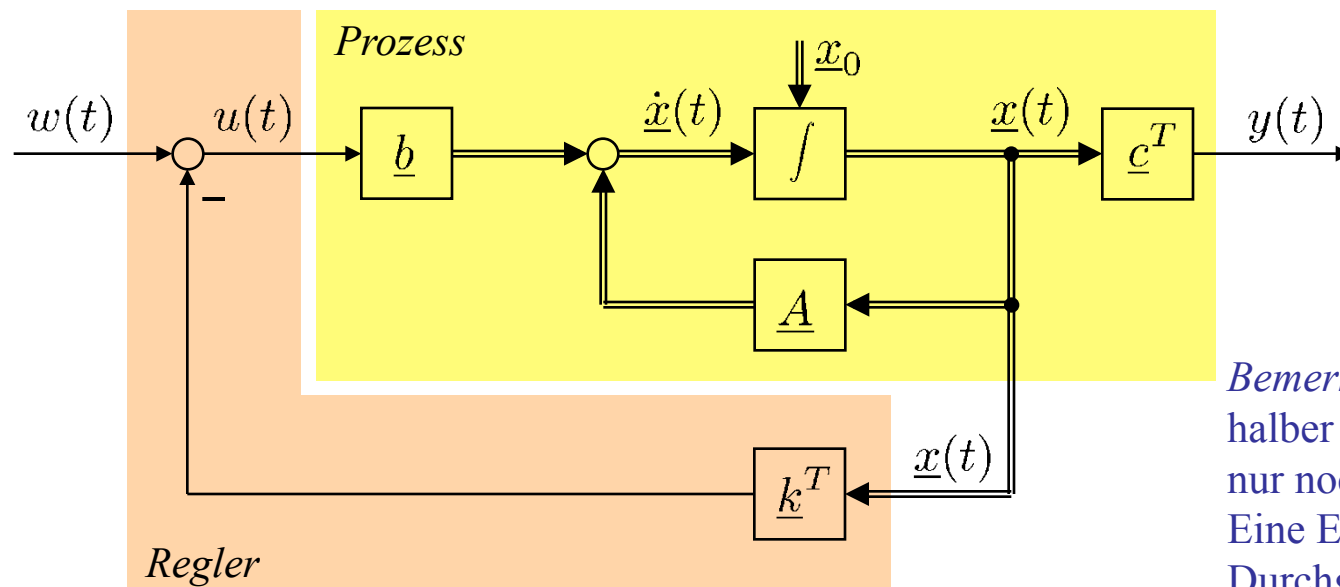
1.4 Zustandsregler durch Polvorgabe

Das Blockschaltbild zeigt, dass nicht der Ausgang $y(t)$ rückgekoppelt wird, sondern der gesamte n -dimensionale Zustandsvektor $\underline{x}(t)$. Der Regler summiert die einzelnen Zustände gewichtet auf (Skalarprodukt) und führt das (skalare) Ergebnis zum Soll/Istwert-Vergleich.

Die Stellgröße berechnet sich also aus:

$$u(t) = w(t) - \underline{k}^T \underline{x}(t) = w(t) - k_1 x_1(t) - k_2 x_2(t) - \dots - k_n x_n(t)$$

Viele Bücher konzentrieren sich zunächst auf den Fall der Ausregelung der Anfangswerte \underline{x}_0 und setzen daher $w(t) = 0$. Wir rechnen weiterhin mit $w(t)$, wegen des direkteren Bezugs zum Standardregelkreis.



Bemerkung: Der besseren Übersicht halber behandeln wir im Weiteren nur noch Systeme ohne Durchgriff. Eine Erweiterung auf Systeme mit Durchgriff ist aber leicht möglich.

1.4 Zustandsregler durch Polvorgabe

Der Zustandsregler wirkt also proportional mit den Faktoren k_i auf die Zustände. Ist es deshalb auch ein P-Regler?

Nein! Um die Zusammenhänge möglichst klar zu machen gehen wir von folgendem Prozess mit Zählerpolynom = 1 aus:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0}$$

Im Zustandsraum lautet die entsprechende Regelungsnormform:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{n-1}(t) \\ \dot{x}_n(t) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u(t)$$

$$y(t) = (1 \ 0 \ \dots \ 0) \underline{x}(t)$$

D.h. $y(t) = x_1(t)$.

1.4 Zustandsregler durch Polvorgabe

Im Laplace-Bereich geschrieben wird die Ausgangsgleichung und die ersten $n-1$ Zustandsdifferentialgleichungen nach einem Tausch der rechten mit der linken Seite zu:

$$X_1(s) = Y(s)$$

$$X_2(s) = sX_1(s) = sY(s)$$

$$X_3(s) = sX_2(s) = s^2X_1(s) = s^2Y(s)$$

⋮

$$X_n(s) = sX_{n-1}(s) = s^{n-1}X_1(s) = s^{n-1}Y(s)$$

Der Zustandsregler führt ja die Zustände $x_1(t)$, $x_2(t)$, ..., $x_n(t)$ zurück. Damit führt er (implizit) den Ausgang, den abgeleiteten Ausgang, den zweimal abgeleiteten Ausgang, ... zurück:

$$U(s) = W(s) - k_1Y(s) - k_2sY(s) - k_3s^2Y(s) - \dots - k_ns^{n-1}Y(s)$$

P

D

D_2

D_{n-1}

Es handelt sich also um einen **PD_{n-1}-Regler**. Allerdings mit einem entscheidenden *Unterschied* zum Standardregler: Die Ableitungen werden *nicht durch Differentiation* des Ausgangssignals gewonnen. Statt dessen werden die *Zustände gemessen*, die schon diesen differenzierten Größen entsprechen! Es gibt hier also keine Verstärkung des Rauschens!

1.4 Zustandsregler durch Polvorgabe

Bemerkung: Wer genau aufgepasst hat, wird einen weiteren Unterschied zum Standardregler erkannt haben: Die Ableitungen beziehen sich ausschließlich auf das Ausgangssignal $y(t)$. Beim Standardregelkreis würden sie sich auf die Regelabweichung $e(t) = w(t) - y(t)$ beziehen. D.h. der PD_{n-1} -Regler im Standardregelkreis würde im Laplace-Bereich so lauten:

$$U(s) = \tilde{k}_1[W(s) - Y(s)] + \tilde{k}_2s[W(s) - Y(s)] + \dots + \tilde{k}_ns^{n-1}[W(s) - Y(s)]$$

Für konstante Führungsgrößen $w(t) = \text{konst.}$ sind aber sowieso alle Ableitung von $w(t)$ gleich Null: $sW(s) = 0, s^2W(s) = 0, \dots$ und dann sind beide Gleichungen (fast) identisch.

Für den Entwurf eines Zustandsreglers, also die systematische Bestimmung der k_i , gibt es zwei Hauptmethoden:

- *Polvorgabe:* Es stehen n Reglerparameter zur Verfügung, mit denen es möglich ist, die n Pole des charakteristischen Polynoms an gewünschte Stellen zu verschieben.
- *Optimierung:* Die Minimierung einer quadratischen Verlustfunktion führt beim Zustandsregler (im Gegensatz zum Standardregler) zu einer global optimalen, einfachen, leicht berechenbaren Lösung. Das wird in Kapitel 1.5 behandelt.

1.4 Zustandsregler durch Polvorgabe

Unser Ziel ist nun eine Gleichung für die Bestimmung der Reglerparameter herzuleiten. Dazu müssen wir die Beziehung zwischen den Reglerparametern und der charakteristischen Gleichung finden. Die Zustandsgleichungen mit Regler lauten:

$$\begin{aligned}\dot{\underline{x}} &= \underline{A}\underline{x} + \underline{b}u = \underline{A}\underline{x} + \underline{b}(w - \underline{k}^T \underline{x}) \\ y &= \underline{c}^T \underline{x}\end{aligned}$$

$$\begin{aligned}\dot{\underline{x}} &= \left(\underline{A} - \underline{b}\underline{k}^T\right) \underline{x} + \underline{b}w \\ y &= \underline{c}^T \underline{x}\end{aligned}$$

D.h. die Systemmatrix \underline{A} wird durch die Zustandsrückkopplung verändert und die Führungsgröße w tritt an die Stelle der Stellgröße u . Alles andere bleibt wie es war.

Da \underline{A} alle Informationen über die Pole des Systems enthält (nämlich alle Nennerkoeffizienten a_i der Übertragungsfunktion), können wir erwarten mit Hilfe der Zustandsrückkopplung diese Pole gezielt ändern zu können; aber auch nur diese (also keine Nullstellen).

Bemerkung: Das *äußere Produkt* $\underline{b}\underline{k}^T$ wird wie folgt berechnet:

$$\underline{b}\underline{k}^T = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} (k_1 \quad k_2 \quad \cdots \quad k_n) = \begin{pmatrix} b_1 k_1 & b_1 k_2 & \cdots & b_1 k_n \\ b_2 k_1 & b_2 k_2 & \cdots & b_2 k_n \\ \vdots & \vdots & \ddots & \vdots \\ b_n k_1 & b_n k_2 & \cdots & b_n k_n \end{pmatrix}$$

1.4 Zustandsregler durch Polvorgabe

Möchten wir nun die n Pole $\lambda_1, \lambda_2, \dots, \lambda_n$ vorgeben, so wünschen wir uns folgende charakteristische Gleichung:

$$\text{Wunsch: } N(s) = (s - \lambda_1)(s - \lambda_2) \cdot \dots \cdot (s - \lambda_n) = s^n + n_{n-1}s^{n-1} + \dots + n_1s + n_0$$

Für die charakteristische Gleichung des geschlossenen Regelkreises müssen wir die Determinante von $s\underline{I} - (\underline{A} - \underline{b}\underline{k}^T)$ berechnen. Gleichsetzen der charakteristischen Gleichung des geschlossenen Regelkreises mit dem Wunschpolynom $N(s)$ führt zu folgender Bedingung:

$$\det \left(s\underline{I} - (\underline{A} - \underline{b}\underline{k}^T) \right) = s^n + n_{n-1}s^{n-1} + \dots + n_1s + n_0$$

Ausrechnen der Determinante führt ebenfalls auf ein Polynom n -ten Grades und Koeffizientenvergleich liefert dann die gesuchten Reglerparameter k_i .

Allerdings ist der Zustandsregler nur dann in der Lage *alle* Pole beliebig zu verschieben, wenn das System $(\underline{A}, \underline{b})$ steuerbar ist.

Denn wir wissen: Für nicht steuerbare Systeme hat die Stellgröße auf bestimmte Zustände keinen Einfluss und kein Regler kann diesen Zuständen eine gewünschte Dynamik aufzwingen.

1.4 Zustandsregler durch Polvorgabe

Die Berechnung der Determinante in der Polvorgabegleichung gestaltet sich recht aufwändig, insbesondere für Systeme hoher Ordnung (großes n). Diese Berechnung vereinfacht sich extrem, wenn die Zustandsgleichungen in *Regelungsnormalform* vorliegen:

$$\underline{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{pmatrix} \quad \underline{b} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

$$\rightarrow \underline{b} \underline{k}^T = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} (k_1 \quad k_2 \quad \cdots \quad k_n) = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ k_1 & k_2 & \cdots & k_n \end{pmatrix}$$

1.4 Zustandsregler durch Polvorgabe

$$\rightarrow \underline{A} - \underline{b} \underline{k}^T = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -(a_0 + k_1) & -(a_1 + k_2) & -(a_2 + k_3) & \dots & -(a_{n-1} + k_n) \end{pmatrix}$$

Das bedeutet: Auch der *geschlossene* Regelkreis lässt sich in Regelungsnormalform schreiben. Wir kommen also durch folgende einfache Operation von der Systemmatrix des offenen zu der des geschlossenen Regelkreises:

$$a_i \rightarrow a_i + k_{i+1}, \quad i = 0, 1, \dots, n - 1$$

Weil die Systemmatrix des geschlossenen Regelkreises wieder in Regelungsnormalform auftritt, können wir das charakteristische Polynom direkt ablesen:

$$\det \left(s \underline{I} - \left(\underline{A} - \underline{b} \underline{k}^T \right) \right) = s^n + (a_{n-1} + k_n) s^{n-1} + \dots + (a_1 + k_2) s + (a_0 + k_1)$$

Gleichsetzen mit dem Wunschpolynom liefert:

$$s^n + (a_{n-1} + k_n) s^{n-1} + \dots + (a_1 + k_2) s + (a_0 + k_1) = s^n + n_{n-1} s^{n-1} + \dots + n_1 s + n_0$$

1.4 Zustandsregler durch Polvorgabe

Daraus ergeben sich direkt die Reglerparameter:

$$k_{i+1} = n_i - a_i, \quad i = 0, 1, \dots, n - 1$$

Oder in vektorieller Form:

$$\underline{k}^T = (n_0 \quad n_1 \quad \dots \quad n_{n-1}) - (a_0 \quad a_1 \quad \dots \quad a_{n-1})$$

Die Reglerparameter entsprechen also gerade den Differenzen zwischen den Koeffizienten des gewünschten charakteristischen Polynoms und denen des offenen Kreises. Die Rückführung wird umso mehr verstärkt (k_i größer), je weiter diese voneinander abweichen.

In Regelungsnormalform ist der Reglerentwurf durch Polvorgabe also nahezu trivial. Liegt ein System *nicht* in Regelungsnormalform vor (das ist der Normalfall), dann haben wir zwei Alternativen:

1. Berechnung der Determinante von $sI - (\underline{A} - \underline{b}\underline{k}^T)$ und anschließender Koeffizientenvergleich mit dem Wunschpolynom.
2. a) Transformation des Systems in Regelungsnormalform ($\underline{x} \rightarrow \underline{x}_T$).
b) Reglerentwurf in Regelungsnormalform mittels $k_{i+1} = n_i - a_i$.
c) Rücktransformation des Reglers auf die ursprünglichen Zustände des Systems ($\underline{x}_T \rightarrow \underline{x}$).

1.4 Zustandsregler durch Polvorgabe

Es fehlt noch die Information, wie der Regler \underline{k}^T transformiert werden muss. Das geregelte Originalsystem folgt den Zustandsdifferentialgleichungen:

$$\dot{\underline{x}}(t) = \left(\underline{A} - \underline{b} \underline{k}^T \right) \underline{x}(t) + \underline{b} w(t)$$

Das transformierte System wird dann:

$$\begin{aligned} \underline{T} \dot{\underline{x}}_T(t) &= \left(\underline{A} - \underline{b} \underline{k}^T \right) \underline{T} \underline{x}_T(t) + \underline{b} w(t) \\ &\rightarrow \dot{\underline{x}}_T(t) = \left(\underline{T}^{-1} \underline{A} \underline{T} - \underline{T}^{-1} \underline{b} \underline{k}^T \underline{T} \right) \underline{x}_T(t) + \underline{T}^{-1} \underline{b} w(t) \end{aligned}$$

Daraus folgt:

$$\underline{k}^T = \underline{k}_T^T \underline{T}^{-1}$$

$$\underline{k}_T^T = \underline{k}^T \underline{T}$$

Regelungsnormalform

Originalsystem

1.4 Zustandsregler durch Polvorgabe

Statt obige Transformationen der Reihe nach durchzuführen, ist es möglich, die Gleichungen zuerst zu vereinfachen. Ist der Regler in der Regelungsnormalform entworfen, soll er ins Originalsystem zurück transformiert werden:

$$\underline{k}^T = \underline{k}_T^T \underline{T}^{-1} = [(n_0 \quad n_1 \quad \cdots \quad n_{n-1}) - (a_0 \quad a_1 \quad \cdots \quad a_{n-1})] \underline{T}^{-1}$$

Einsetzen der Transformationsmatrix liefert:

$$\underline{k}^T = (n_0 \quad n_1 \quad \cdots \quad n_{n-1}) \begin{pmatrix} \underline{s}_S^T \\ \underline{s}_S^T \underline{A} \\ \underline{s}_S^T \underline{A}^2 \\ \vdots \\ \underline{s}_S^T \underline{A}^{n-1} \end{pmatrix} - (a_0 \quad a_1 \quad \cdots \quad a_{n-1}) \begin{pmatrix} \underline{s}_S^T \\ \underline{s}_S^T \underline{A} \\ \underline{s}_S^T \underline{A}^2 \\ \vdots \\ \underline{s}_S^T \underline{A}^{n-1} \end{pmatrix}$$

Ausmultiplizieren des rechten Produkts ergibt:

$$\underline{s}_S^T (a_0 \underline{I} + a_1 \underline{A} + a_2 \underline{A}^2 + \dots + a_{n-1} \underline{A}^{n-1})$$

Nach dem Cayley-Hamilton-Theorem ist der Ausdruck in der Klammer gleich $-\underline{A}^n$. Setzt man dies ein, so erhält man eine vereinfachte Gleichung für den Reglerentwurf durch Polvorgabe für Systeme in beliebiger Zustandsraumform:

1.4 Zustandsregler durch Polvorgabe

$$\underline{k}^T = (n_0 \quad n_1 \quad \cdots \quad n_{n-1}) \begin{pmatrix} \underline{s}_S^T \\ \underline{s}_S^T \underline{A} \\ \underline{s}_S^T \underline{A}^2 \\ \vdots \\ \underline{s}_S^T \underline{A}^{n-1} \end{pmatrix} + \underline{s}_S^T \underline{A}^n$$

Formel nach Ackermann

Der Zustandsregler durch Polvorgabe für beliebige Zustandsraumformen berechnet sich mit:

$$\underline{k}^T = \underline{s}_S^T (n_0 \underline{I} + n_1 \underline{A} + n_2 \underline{A}^2 + \dots + n_{n-1} \underline{A}^{n-1} + \underline{A}^n)$$

$$= (n_0 \quad n_1 \quad \cdots \quad n_{n-1} \quad 1) \begin{pmatrix} \underline{s}_S^T \\ \underline{s}_S^T \underline{A} \\ \underline{s}_S^T \underline{A}^2 \\ \vdots \\ \underline{s}_S^T \underline{A}^{n-1} \\ \underline{s}_S^T \underline{A}^n \end{pmatrix}$$

\underline{s}_S^T ist die letzte Zeile der inversen Steuerbarkeitsmatrix des Originalsystems.

1.4 Zustandsregler durch Polvorgabe

Entwurf eines Zustandsreglers mittels Ljapunow-Gleichung

Eine sehr trickreiche und einfach umzusetzende Methode der Polvorgabe ist über die Lösung einer Ljapunow-Gleichung möglich. Die Methode läuft so ab:

1. Wähle eine Matrix \underline{N} , die als Eigenwerte die gewünschten Pole des geschlossenen Regelkreises hat. Einschränkung: Diese Eigenwerte müssen sich alle von den Eigenwerten von \underline{A} unterscheiden.

Wir können also entweder direkt die Wunschpole vorgeben, indem wir eine *Diagonalform* wählen (da sind die Eigenwerte direkt gleich den Diagonalelementen) oder wir geben ein *charakteristisches Polynom* vor,

$$N(s) = (s - \lambda_1)(s - \lambda_2) \cdot \dots \cdot (s - \lambda_n) = s^n + n_{n-1}s^{n-1} + \dots + n_1s + n_0$$

indem wir eine Systemmatrix vom Typ der *Regelungsnormalform* wählen:

$$\underline{N} = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & & 0 \\ \vdots & & \ddots & \\ 0 & \dots & 0 & \lambda_n \end{pmatrix} \quad \text{oder} \quad \underline{N} = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -n_0 & -n_1 & -n_2 & \dots & -n_{n-1} \end{pmatrix}$$

1.4 Zustandsregler durch Polvorgabe

- Wähle einen $1 \times n$ -Vektor $\tilde{\underline{k}}^T$, so dass $(\underline{N}, \tilde{\underline{k}}^T)$ beobachtbar ist. Dieser Vektor kann als transformierter Rückführvektor/Reglervektor interpretiert werden. Eine mögliche Wahl, welche die Beobachtbarkeit sicherstellt ist:

$$\tilde{\underline{k}}^T = (1 \quad 1 \quad \dots \quad 1)$$

- Löse folgende Ljapunow-Gleichung nach \underline{T} auf:

$$\underline{A}\underline{T} - \underline{T}\underline{N} = \underline{b}\tilde{\underline{k}}^T$$

- Der Rückführvektor des Zustandsreglers berechnet sich aus:

$$\underline{k}^T = \tilde{\underline{k}}^T \underline{T}^{-1}$$

Die Funktionsweise dieser Methode kann man wie folgt nachprüfen: verschiedene Realisationen des selben Systems

$$\tilde{\underline{k}}^T = \underline{k}^T \underline{T} \quad (\text{wenn } \underline{T} \text{ nicht singulär ist})$$

$$\rightarrow \underline{A}\underline{T} - \underline{T}\underline{N} = \underline{b}\underline{k}^T \underline{T} \quad \rightarrow \quad (\underline{A} - \underline{b}\underline{k}^T)\underline{T} = \underline{T}\underline{N} \quad \rightarrow \quad \underline{A} - \underline{b}\underline{k}^T = \underline{T}\underline{N}\underline{T}^{-1}$$

Wenn wir \underline{T} als Transformationsmatrix interpretieren, dann sind $\underline{A} - \underline{b}\underline{k}^T$ und \underline{N} Systemmatrizen unterschiedlicher Realisationen des selben Systems. Damit haben beide auch die selben Eigenwerte! Der geschlossene Regelkreis hat also die selben Eigenwerte wie \underline{N} .

1.4 Zustandsregler durch Polvorgabe

Regelstrecke, offener und geschlossener Regelkreis im *Frequenzbereich*

Es ist oft nützlich und auch dem Verständnis förderlich, wichtiges Ein-/Ausgangsverhalten nicht nur im Zustandsraum sondern auch als Übertragungsfunktion zu beschreiben.

Die Übertragungsfunktion der *Regelstrecke* $u \rightarrow y$ ist:

$$G_S(s) = \underline{c}^T (s\underline{I} - A)^{-1} \underline{b}$$

Die Übertragungsfunktion des *offenen Regelkreises* läuft über \underline{k}^T zurück statt über \underline{c}^T zum Ausgang und ist daher:

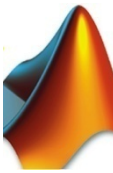
$$G_0(s) = \underline{k}^T (s\underline{I} - A)^{-1} \underline{b}$$

Die Übertragungsfunktion des geschlossenen Regelkreises $w \rightarrow y$ muss die Zustandsrückkopplung mit berücksichtigen:

$$G_w(s) = \underline{c}^T (s\underline{I} - A + \underline{b}\underline{k}^T)^{-1} \underline{b}$$

1.4 Zustandsregler durch Polvorgabe

MATLAB



Reglerentwurf durch Polvorgabe:

```
System = ss(A,b,c,d); % Definiere ein System im Zustandsraum
rank(ctrb(System))    % Prüfe dessen Steuerbarkeit (Ergebnis == n ?)
EW = [-3 -2+i -2-i]; % Gewünschte Eigenwerte
k = place(A,b,EW);    % Reglerentwurf mittels Polvorgabe
```

1.5 Zustandsregler durch Optimierung (LQ)

Aus Kapitel 11.1 in MRT I kennen wir folgende quadratische Gütefunktion:

$$J = \int_0^{t_e} [e^2(t) + \alpha u^2(t)] dt$$

Sie bestraft die Regelabweichung und den Stellaufwand. Gehen wir zunächst von einer konstanten Führungsgröße aus, also von einer Festwertregelung (*regulation problem*). Dann können wir vereinfachend die Führungsgröße $w = 0$ setzen (eine andere Konstante lässt sich durch Verschiebung des Arbeitspunktes von y leicht erzeugen). Dann vereinfacht sich die Verlustfunktion zu:

$$J = \int_0^{t_e} [y^2(t) + \alpha u^2(t)] dt$$

Wir wollen also eine Störung innerhalb einer Zeit t_e "möglichst gut" ausregeln (y soll auf 0 zurück gehen), wobei obige Verlustfunktion angibt, was mit "möglichst gut" genau gemeint ist. Wird die Endzeit t_e klein gewählt, so kann es sein, dass y noch weit von 0 entfernt ist. Daher ist es sinnvoll noch einen Term für die Abweichung von y vom gewünschten Endwert (= 0) zum Zeitpunkt t_e in die Verlustfunktion aufzunehmen.

1.5 Zustandsregler durch Optimierung (LQ)

Mit den im Zustandsraum üblichen Variablennamen lautet dann die vollständige Verlustfunktion:

$$J = sy^2(t_e) + \int_0^{t_e} [q_y y^2(t) + ru^2(t)] dt$$

Für mehrere Eingänge und Ausgänge lautet die entsprechende Schreibweise:

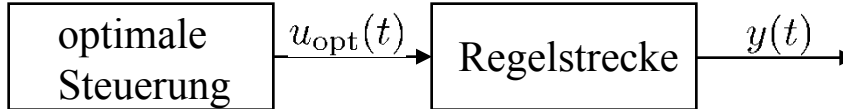
$$J = \underline{y}(t_e)^T \underline{S} \underline{y}(t_e) + \int_0^{t_e} [\underline{y}(t)^T \underline{Q}_y \underline{y}(t) + \underline{u}(t)^T \underline{R} \underline{u}(t)] dt$$

Oft werden die Wichtungsmatrizen \underline{S} , \underline{Q}_y und \underline{R} als Diagonalmatrizen gewählt. Z.B.:

$$\begin{pmatrix} u_1 & u_2 & \cdots & u_{n_u} \end{pmatrix} \begin{pmatrix} r_{11} & 0 & \cdots & 0 \\ 0 & r_{22} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_{n_u n_u} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n_u} \end{pmatrix}$$
$$= r_{11}u_1^2 + r_{22}u_2^2 + \dots + r_{n_u n_u}u_{n_u}^2$$

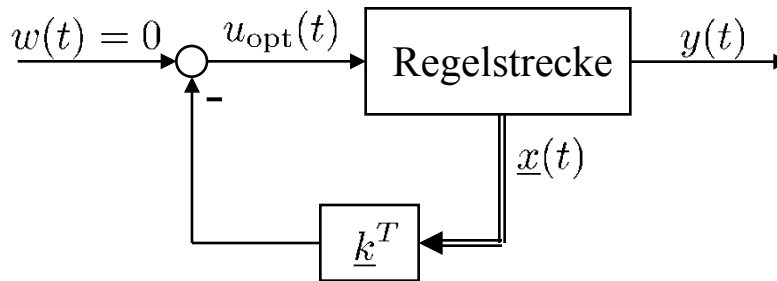
1.5 Zustandsregler durch Optimierung (LQ)

Wenn wir diese Verlustfunktion bezüglich $u(t)$, $0 \leq t \leq t_e$, minimieren würden, bekämen wir als Ergebnis die *optimale Stellgrößenfolge* heraus.



Wir sind aber *nicht* an einer *Steuerung* interessiert, sondern an einer *Regelung*. Wir suchen die optimale Zustandsregelung (zunächst für $w = 0$):

$$u(t) = -\underline{k}^T \underline{x}(t)$$



Wenn wir die Zustandsrückkopplung in die Verlustfunktion einsetzen und dann über \underline{k}^T (statt über $u(t)$, $0 \leq t \leq t_e$) minimieren würden, erhielten wir den *optimalen Zustandsregler*.

1.5 Zustandsregler durch Optimierung (LQ)

Allerdings wäre dieser Regler *zeitvariant!* D.h.

$$\underline{k}^T = \underline{k}^T(t)$$

Mathematisch ergibt sich diese Zeitvarianz als Folge des endlichen Betrachtungszeitraums t_e .

Einen *zeitinvarianten* Regler erhalten wir, wenn wir "nur" einen asymptotisch optimalen Regler fordern, d.h. für $t_e \rightarrow \infty$. Wir können dann annehmen, dass $y(t_e \rightarrow \infty) = 0$; ansonsten würde das Integral nicht konvergieren und wir könnten gar keinen optimalen Regler berechnen. Weil $y(t_e \rightarrow \infty) = 0$, fällt der erste Term in der Verlustfunktion weg und sie vereinfacht sich zu:

$$J = \int_0^\infty [q_y y^2(t) + r u^2(t)] dt \quad \text{bzw.} \quad J = \int_0^\infty [\underline{y}(t)^T \underline{Q}_y \underline{y}(t) + \underline{u}(t)^T \underline{R} \underline{u}(t)] dt$$

Noch etwas flexibler sind wir, wenn wir die Zustände (und nicht den Ausgang/die Ausgänge) einzeln gewichten können. Deshalb geht man i.a. von folgender Verlustfunktion aus, die für $\underline{Q} = \underline{c} \underline{c}^T q_y$ bzw. $\underline{Q} = \underline{c} \underline{Q}_y \underline{c}^T$ obige Verlustfunktion als Spezialfall enthält:

$$J = \int_0^\infty [\underline{x}(t)^T \underline{Q} \underline{x}(t) + r u^2(t)] dt$$

$$\text{bzw.} \quad J = \int_0^\infty [\underline{x}(t)^T \underline{Q} \underline{x}(t) + \underline{u}(t)^T \underline{R} \underline{u}(t)] dt$$

1.5 Zustandsregler durch Optimierung (LQ)

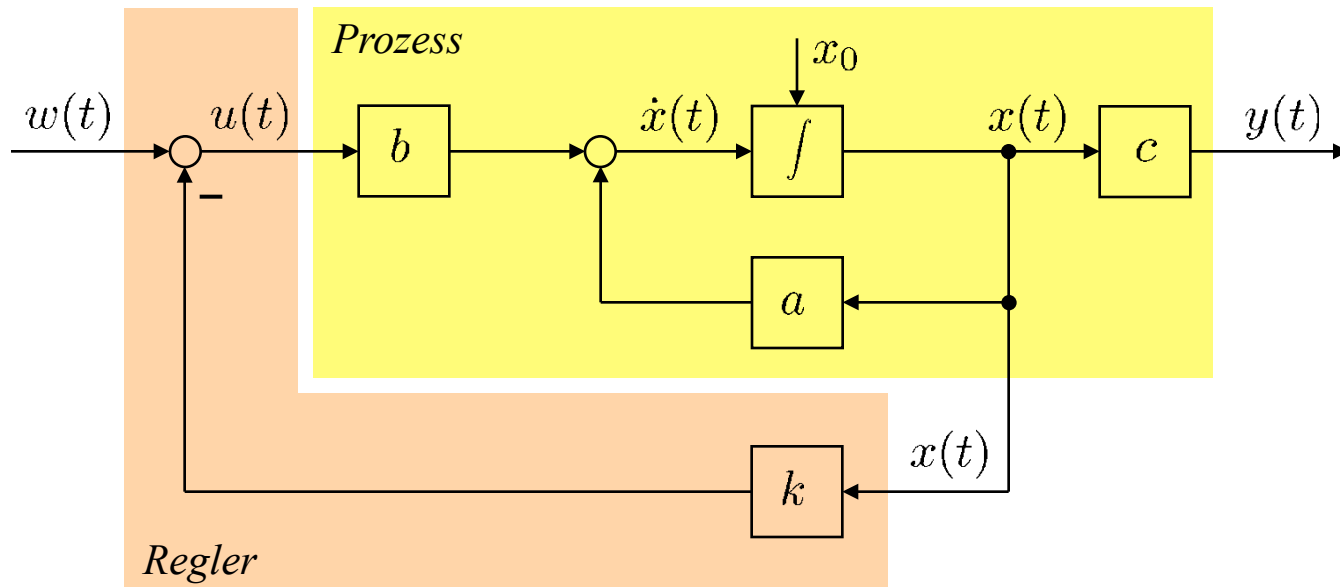
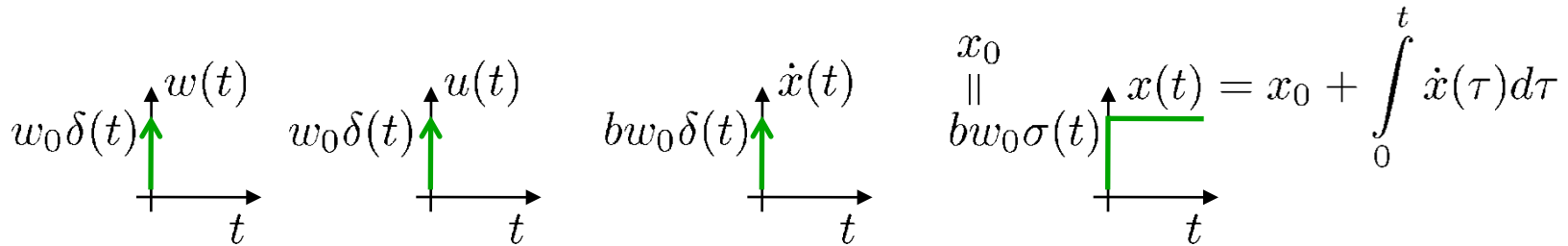
Die Minimierung dieser Verlustfunktion bzgl. einer linearen Zustandsrückführung \underline{k}^T führt auf einen zeitinvarianten, optimalen Regler. Dieser Regler wird oft **LQ-Regler** bzw. **linear quadratischer Regler** genannt, weil er auf der Optimierung einer quadratischen Verlustfunktion bzgl. eines linearen Reglers basiert.

Weitere Bemerkungen zur Verlustfunktion:

- Der Term $\underline{x}(t)^T \underline{Q} \underline{x}(t)$ bestraft Abweichungen der Zustände von 0. Für $t \rightarrow \infty$ müssen alle Zustände gegen 0 streben, ansonsten divergiert das Integral gegen unendlich.
- Liegt der Arbeitspunkt nicht bei $\underline{x} = \underline{0}$, so kann man diese Vorgehensweise trotzdem weiter verfolgen. Zuvor ist nur eine Arbeitspunktverschiebung der Variablen notwendig. Das wird im nächsten Beispiel veranschaulicht.
- Die Verlustfunktion beinhaltet nur das Ausregeln einer Anfangsabweichung $\underline{x}(0) = \underline{x}_0$. Dies bedeutet: Der sich ergebende optimale Regler ist auf eine plötzlich auftretende Verschiebungen des Zustandsvektors ausgelegt. Das entspricht einer impulsförmigen Führungsgröße $w(t) = \delta(t)$, denn nur ein Delta-Impuls kann eine ruckartige Verschiebung der Zustände bewirken. Sprungförmige o.a. Führungsgrößen werden bei der Optimierung nicht berücksichtigt. Kapitel 1.7 behandelt die dafür nötigen Erweiterungen.

1.5 Zustandsregler durch Optimierung (LQ)

Ausregeln eines Dirac-Impulses $w_0\delta(t)$ = Ausregeln einer Anfangsabweichung x_0



Ein Dirac-Impuls als Führungsgröße ist identisch mit einer Anfangsabweichung des Zustandes $x_0 \neq 0$. Es werden alle Frequenzen gleich stark angeregt. Im Gegensatz zu einem Sprung als Führungsgröße, entsteht dabei **keine** bleibende Regelabweichung, auch ohne I-Anteil!

1.5 Zustandsregler durch Optimierung (LQ)

Beispiel: Arbeitspunktverschiebung zur Nullpunktänderung

$$\dot{\underline{x}}(t) = \begin{pmatrix} 0 & 1 \\ -3 & -2 \end{pmatrix} \underline{x}(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t)$$

$$y(t) = (4 \ 0) \underline{x}(t)$$

Gewünschte (konstante) Führungsgröße $w = 7$ (statt $w = 0$).

Daraus folgen die stationären Werte ($\dot{\underline{x}}(t) = \underline{0}$):

$$y = w \quad \rightarrow \quad y = 7 \quad \text{(Sollwert = Istwert)}$$

$$y = 4x_1 \quad \rightarrow \quad x_1 = \frac{7}{4} \quad \text{(Ausgangsgleichung)}$$

$$0 = x_2 \quad \rightarrow \quad x_2 = 0 \quad \text{(1. Zustands-DGL)}$$

$$0 = -3x_1 - 2x_2 + u \quad \rightarrow \quad u = 3x_1 = \frac{21}{4} \quad \text{(2. Zustands-DGL)}$$

Also können wir mit folgenden Transformationen den Arbeitspunkt auf 0 verschieben:

$$\tilde{y}(t) = y(t) - 7 \quad \tilde{x}_1(t) = x_1(t) - \frac{7}{4} \quad \tilde{x}_2(t) = x_2(t) \quad \tilde{u}(t) = u(t) - \frac{21}{4}$$

1.5 Zustandsregler durch Optimierung (LQ)

Der Reglerentwurf findet dann im transformierten System mit $\tilde{w} = 0$ statt:

$$\dot{\tilde{x}}(t) = \begin{pmatrix} 0 & 1 \\ -3 & -2 \end{pmatrix} \tilde{x}(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tilde{u}(t)$$

$$\tilde{y}(t) = (4 \ 0) \tilde{x}(t)$$

Dieser Reglerentwurf liefert einen Regler $\tilde{k}^T = (\tilde{k}_1 \ \tilde{k}_2)$.

Die Regelung läuft dann so ab:

1. Messung der Zustände $x_1(t)$ und $x_2(t)$.
2. Berechnung der transformierten Zustände: $\tilde{x}_1(t) = x_1(t) - \frac{7}{4}$, $\tilde{x}_2(t) = x_2(t)$.
3. Berechnung der transformierten Stellgröße:

$$\tilde{u}(t) = \tilde{w}(t) - \tilde{k}_1 \tilde{x}_1(t) - \tilde{k}_2 \tilde{x}_2(t) = 0 - \tilde{k}_1 \left(x_1(t) - \frac{7}{4} \right) - \tilde{k}_2 x_2(t)$$

4. Rücktransformation auf die Original-Stellgröße:

$$u(t) = \frac{21}{4} + \tilde{u}(t) = \frac{21}{4} - \tilde{k}_1 \left(x_1(t) - \frac{7}{4} \right) - \tilde{k}_2 x_2(t)$$

1.5 Zustandsregler durch Optimierung (LQ)

Einschub: Quadratische Matrix-Formen

Eine quadratische Gleichung in einer Variablen x hat folgende Form:

$$y = ax^2 + bx + c = xax + bx + c$$

Dabei bestimmen a , b und c die Größe des quadratischen, linearen und konstanten Anteils. Die Steigung und Krümmung dieser Parabel lassen sich aus der 1. und 2. Ableitung berechnen:

$$\frac{dy}{dx} = 2ax + b, \quad \frac{d^2y}{dx^2} = 2a$$

Im höher dimensionalen Fall ist die Variable ein n -dimensionaler Vektor \underline{x} , und eine quadratische Gleichung lässt sich wie folgt beschreiben:

$$y = \underline{x}^T \underline{A} \underline{x} + \underline{b}^T \underline{x} + c$$

Die Steigung existiert in n Raumrichtungen. Daher ist die 1. Ableitung ein n -dimensionaler Vektor, der sog. *Gradient*. Die 2. Ableitung kann in allen Richtungskombinationen x_i, x_j erfolgen und ist deshalb eine $n \times n$ -Matrix, die sog. *Hesse-Matrix*:

$$\frac{dy}{d\underline{x}} = 2\underline{A} \underline{x} + \underline{b}, \quad \frac{d^2y}{d\underline{x}^2} = 2\underline{A}$$

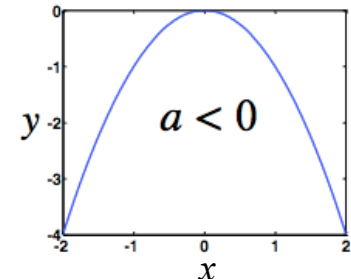
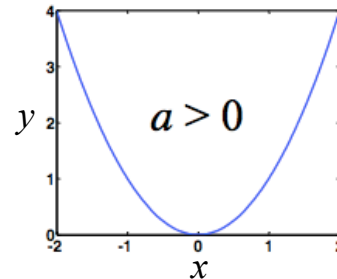
1.5 Zustandsregler durch Optimierung (LQ)

Positive und negative Definitheit

Im einfachsten 1-dimensionalen Fall ist $b = c = 0$ und die quadratische Funktion entspricht einer Parabel durch den Ursprung.

$$y = a x^2$$

Diese Parabel ist nach oben geöffnet für $a > 0$ und nach unten geöffnet für $a < 0$.

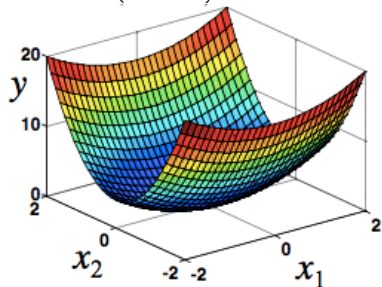


Im einfachsten höher dimensionalen Fall ist $\underline{b} = \underline{0}$, $c = 0$ und die quadratische Funktion entspricht einer Hyper-Parabel durch den Ursprung.

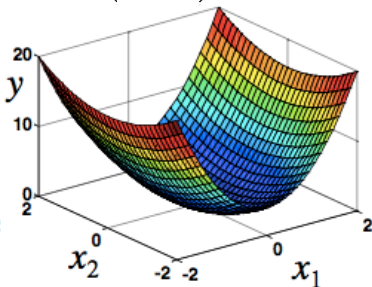
$$y = \underline{x}^T \underline{A} \underline{x}$$

Diese Hyper-Parabel ist nach oben geöffnet und hat ein Minimum falls $\underline{A} \succ 0$, d.h. positiv definit, ist. Sie ist nach unten geöffnet und hat ein Maximum falls $\underline{A} \prec 0$ (negativ definit).

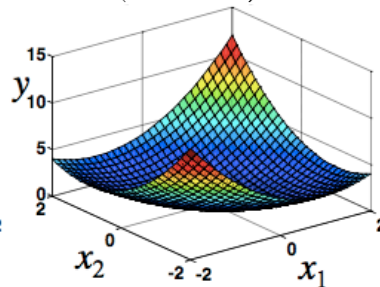
$$\underline{A} = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix} \succ 0$$



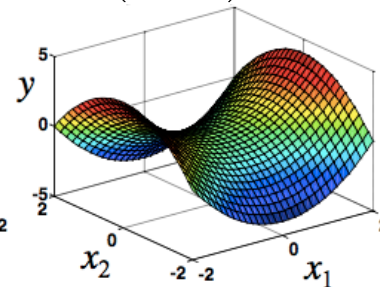
$$\underline{A} = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix} \succ 0$$



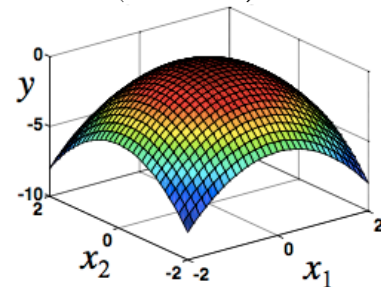
$$\underline{A} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \succ 0$$



$$\underline{A} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$



$$\underline{A} = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \prec 0$$



1.5 Zustandsregler durch Optimierung (LQ)

Eigenschaften positiv definiter (quadratischer) Matrizen \underline{A}

- Die Matrix \underline{A} hat vollen Rang, d.h. linear unabhängige Spalten.
- Alle Eigenwerte von \underline{A} sind positiv.
- Für jeden Vektor $\underline{x} \neq \underline{0}$ gilt $\underline{x}^T \underline{A} \underline{x} > 0$.
- Symbolisch schreibt man „ \underline{A} ist positiv definit“ als: $\underline{A} \succ 0$.
- Für $\underline{A} - \underline{B} \succ 0$ kann man auch schreiben: $\underline{A} \succ \underline{B}$.
Das bedeutet dann: $\underline{x}^T \underline{A} \underline{x} > \underline{x}^T \underline{B} \underline{x}$ für alle $\underline{x} \neq \underline{0}$.
- Alle Matrizen der Form $\underline{A} = \underline{B}^T \underline{B}$ mit vollem Rang sind symmetrisch und positiv definit.
Sie entstehen z.B. als Kovarianz-Matrizen wie sie auch in der mehrdimensionalen Gaußverteilung vorkommen:
$$\underline{A} = \sum_{i=1}^N \underbrace{(\underline{x}(i) - \bar{\underline{x}})}_{\text{Spalte}} \underbrace{(\underline{x}(i) - \bar{\underline{x}})^T}_{\text{Zeile}}$$

Symmetrie in quadratischen Matrix-Formen

Jede quadratische Matrix-Form kann und sollte konventionsgemäß mittels einer symmetrischen Matrix geschrieben werden:

$$(\underline{x}_1 \ \underline{x}_2) \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \end{pmatrix} = a\underline{x}_1^2 + (b+c)\underline{x}_1\underline{x}_2 + d\underline{x}_2^2 = (\underline{x}_1 \ \underline{x}_2) \begin{pmatrix} a & \frac{b+c}{2} \\ \frac{b+c}{2} & d \end{pmatrix} \begin{pmatrix} \underline{x}_1 \\ \underline{x}_2 \end{pmatrix}$$

1.5 Zustandsregler durch Optimierung (LQ)

Herleitung: Optimaler linearer Zustandsregler

Folgende Verlustfunktion soll minimiert werden:

$$J = \int_0^{\infty} [\underline{x}(t)^T \underline{Q} \underline{x}(t) + ru^2(t)] dt \quad \text{mit } u(t) = -\underline{k}^T \underline{x}(t)$$

Zunächst berechnen wir den Wert der Verlustfunktion für das *ungeregelte* System, d.h. für $u(t) = 0$:

$$J = \int_0^{\infty} \underline{x}(t)^T \underline{Q} \underline{x}(t) dt = \int_0^{\infty} \underline{x}_0^T e^{\underline{A}^T t} \underline{Q} e^{\underline{A} t} \underline{x}_0 dt = \underline{x}_0^T \underline{P} \underline{x}_0 \quad \text{mit } \underline{P} = \int_0^{\infty} e^{\underline{A}^T t} \underline{Q} e^{\underline{A} t} dt$$

Partielle Integration liefert:

$$\underline{P} = \int_0^{\infty} \underbrace{e^{\underline{A}^T t}}_u \underbrace{\underline{Q} e^{\underline{A} t}}_{v'} dt = \underbrace{e^{\underline{A}^T t} \underline{Q}}_u \underbrace{\underline{A}^{-1} e^{\underline{A} t}}_v \Big|_0^{\infty} - \int_0^{\infty} \underbrace{\underline{A}^T e^{\underline{A}^T t}}_{u'} \underbrace{\underline{Q} \underline{A}^{-1} e^{\underline{A} t}}_v dt$$

Vertauschen der Multiplikationsreihenfolge von \underline{A}^{-1} und $e^{\underline{A} t}$ liefert:

$$\underline{P} = -\underline{Q} \underline{A}^{-1} - \underline{A}^T \int_0^{\infty} e^{\underline{A}^T t} \underline{Q} e^{\underline{A} t} dt \underline{A}^{-1} = -\underline{Q} \underline{A}^{-1} - \underline{A}^T \underline{P} \underline{A}^{-1}$$

Partielle Integration:

$$\int_a^b uv' dt = uv \Big|_a^b - \int_a^b u'v dt$$

$$\text{mit } u' = \frac{d}{dt}u, \quad v' = \frac{d}{dt}v$$

1.5 Zustandsregler durch Optimierung (LQ)

Rechtsmultiplikation mit \underline{A} führt auf die:

$$\text{Ljapunow-Gleichung: } \underline{A}^T \underline{P} + \underline{P} \underline{A} = -\underline{Q}$$

Die Ljapunow-Gleichung kann auch für die Überprüfung der Stabilität verwendet werden. Es gilt nämlich folgender Satz:

Stabilität linearer Systeme nach Ljapunow

Ist \underline{Q} eine beliebige, symmetrische, positiv definite Matrix, dann gibt es für die Ljapunow-Gleichung genau dann eine symmetrische, positiv definite Lösung \underline{P} , wenn die Systemmatrix \underline{A} stabil ist, d.h. alle ihre Eigenwerte einen negativen Realteil haben.

Bemerkung: Eine quadratische Matrix \underline{P} ist dann *positiv definit*, wenn für jeden beliebigen Vektor \underline{x} (ungleich $\underline{0}$) gilt:

$$\underline{x}^T \underline{P} \underline{x} > 0$$

Eine positiv definite Matrix hat nur positive Diagonalelemente und positive Eigenwerte. Die Gleichung $\underline{x}^T \underline{P} \underline{x}$ beschreibt eine (Hyper)-Parabel mit dem Minimum im Ursprung.

1.5 Zustandsregler durch Optimierung (LQ)

Jetzt berechnen wir den Wert der Verlustfunktion für das *geregelte* System, d.h. für $u(t) = -\underline{k}^T \underline{x}(t) = -\underline{x}(t)^T \underline{k}$:

$$\begin{aligned} J &= \int_0^{\infty} [\underline{x}(t)^T \underline{Q} \underline{x}(t) + r u^2(t)] dt = \int_0^{\infty} [\underline{x}(t)^T \underline{Q} \underline{x}(t) + \underline{x}(t)^T \underline{k} r \underline{k}^T \underline{x}(t)] dt \\ &= \int_0^{\infty} \underline{x}(t)^T (\underline{Q} + r \underline{k} \underline{k}^T) \underline{x}(t) dt = \int_0^{\infty} \underline{x}(t)^T \underline{\bar{Q}} \underline{x}(t) dt \quad \text{mit} \quad \underline{\bar{Q}} = \underline{Q} + r \underline{k} \underline{k}^T \end{aligned}$$

Mit den gleichen Umformungen wie zuvor beim unregulierten System erhält man

$$J = \underline{x}_0^T \underline{P} \underline{x}_0$$

und die Ljapunow-Gleichung Systemmatrix des *geregelten* Systems

$$\underline{\bar{A}}^T \underline{P} + \underline{P} \underline{\bar{A}} = -\underline{\bar{Q}} \quad \text{mit} \quad \underline{\bar{A}} = \underline{A} - \underline{b} \underline{k}^T$$

1.5 Zustandsregler durch Optimierung (LQ)

Um den optimalen Regler zu finden, müssen wir jetzt die Verlustfunktion des geregelten Systems nach den Reglerparametern k_i (Elementen von \underline{k}^T) ableiten und nullsetzen:

$$\frac{\partial J}{\partial k_i} = \underline{x}_0^T \frac{\partial \underline{P}}{\partial k_i} \underline{x}_0 = 0, \quad i = 1, 2, \dots, n$$

Da diese Gleichung für beliebige \underline{x}_0 gelten soll, muss die Ableitung von \underline{P} gleich Null sein:

$$\frac{\partial \underline{P}}{\partial k_i} = \underline{0}, \quad i = 1, 2, \dots, n$$

Um diese Optimalitätsbedingungen verwenden zu können, leiten wir die Ljapunow-Gleichung nach den Reglerparametern ab:

$$\frac{\partial \bar{\underline{A}}^T}{\partial k_i} \underline{P} + \bar{\underline{A}}^T \overbrace{\frac{\partial \underline{P}}{\partial k_i}}^{=0} + \overbrace{\frac{\partial \underline{P}}{\partial k_i}}^{=0} \bar{\underline{A}} + \underline{P} \frac{\partial \bar{\underline{A}}}{\partial k_i} = -\frac{\partial \bar{\underline{Q}}}{\partial k_i}$$

$$\rightarrow -\frac{\partial \underline{k}}{\partial k_i} \underline{b}^T \underline{P} - \underline{P} \underline{b} \frac{\partial \underline{k}^T}{\partial k_i} = -r \frac{\partial \underline{k}}{\partial k_i} \underline{k}^T - r \underline{k} \frac{\partial \underline{k}^T}{\partial k_i}$$

$$\rightarrow \frac{\partial \underline{k}}{\partial k_i} \left(r \underline{k}^T - \underline{b}^T \underline{P} \right) + \left(r \underline{k} - \underline{P} \underline{b} \right) \frac{\partial \underline{k}^T}{\partial k_i} = 0$$

Merker:

$$\bar{\underline{A}} = \underline{A} - \underline{b} \underline{k}^T$$

$$\bar{\underline{A}}^T = \underline{A}^T - \underline{k} \underline{b}^T$$

$$\bar{\underline{Q}} = \underline{Q} + r \underline{k} \underline{k}^T$$

1.5 Zustandsregler durch Optimierung (LQ)

Die letzte Gleichung ist erfüllt, wenn $r\underline{k}^T = \underline{b}^T \underline{P}$, da dann der erste Term zu Null wird und der zweite Term ebenfalls wegen (\underline{P} ist symmetrisch!):

$$\left(r\underline{k}^T\right)^T = r\underline{k} = \left(\underline{b}^T \underline{P}\right)^T = \underline{P}^T \underline{b} = \underline{P} \underline{b} \quad \rightarrow \quad r\underline{k} = \underline{P} \underline{b}$$

Somit berechnet sich der optimale Regler mittels:

$$\text{Optimaler Zustandsregler: } \underline{k}^T = \frac{1}{r} \underline{b}^T \underline{P}$$

Setzt man den optimalen Regler in die Ljapunow-Gleichung ein, so ergibt sich die

$$\text{Matrix-Riccati-Gleichung: } \underline{A}^T \underline{P} + \underline{P} \underline{A} - \frac{1}{r} \underline{P} \underline{b} \underline{b}^T \underline{P} + \underline{Q} = \underline{0}$$

Die symmetrische, positiv definite Lösung \underline{P} dieser *algebraischen* Matrix-Riccati-Gleichung wird in obige Gleichung für den optimalen Regler eingesetzt.

Wir erkennen schon, dass für den Spezialfall $r \rightarrow \infty$ die Matrix-Riccati-Gleichung in die Ljapunow-Gleichung des *ungeregelten* Systems übergeht. D.h. für unendlich hohe Stellgrößenbestrafung ist der Regler inaktiv mit $\underline{k}^T = \underline{0}$ und daher $u(t) = 0$.

1.5 Zustandsregler durch Optimierung (LQ)

Nachtrag: Herleitung der Matrix-Riccati-Gleichung

Ljapunov-Gleichung: $\underline{\bar{A}}^T \underline{P} + \underline{P} \underline{\bar{A}} = -\underline{\bar{Q}}$ mit $\underline{\bar{A}} = \underline{A} - \underline{b} \underline{k}^T$ und $\underline{\bar{Q}} = \underline{Q} + r \underline{k} \underline{k}^T$

Optimaler Regler: $\underline{k}^T = \frac{1}{r} \underline{b}^T \underline{P}$ bzw. $\underline{k} = \frac{1}{r} \underline{P} \underline{b}$

Es gilt also:

$$(\underline{A}^T - \underline{k} \underline{b}^T) \underline{P} + \underline{P} (\underline{A} - \underline{b} \underline{k}^T) + \underline{Q} + r \underline{k} \underline{k}^T = 0$$

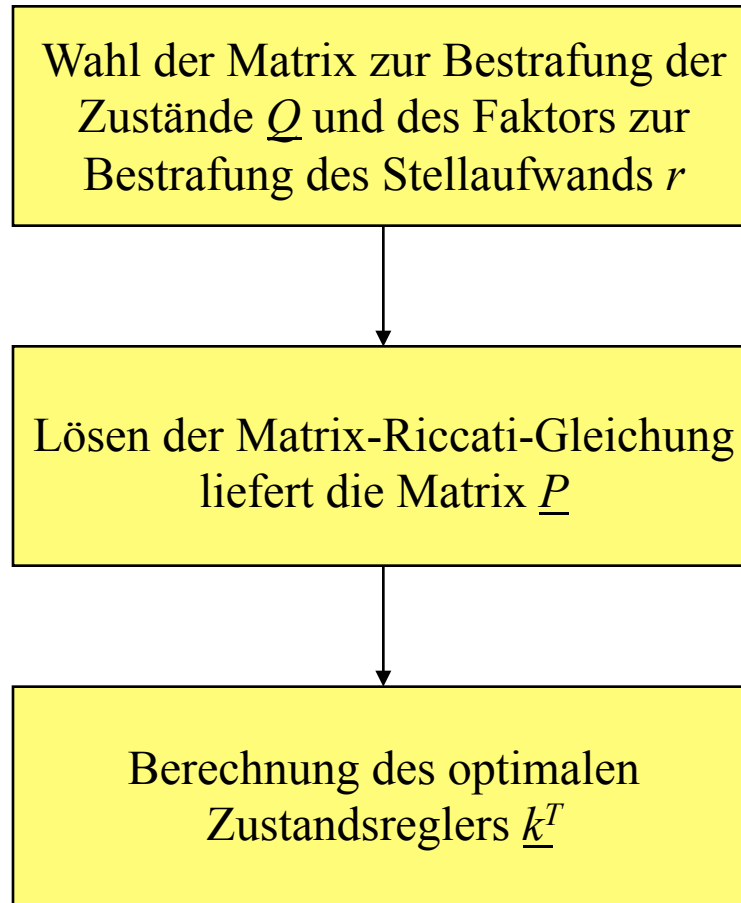
$$(\underline{A}^T - \frac{1}{r} \underline{P} \underline{b} \underline{b}^T) \underline{P} + \underline{P} (\underline{A} - \underline{b} \frac{1}{r} \underline{b}^T \underline{P}) + \underline{Q} + r \frac{1}{r} \underline{P} \underline{b} \frac{1}{r} \underline{b}^T \underline{P} = 0$$

$$\underline{A}^T \underline{P} - \frac{1}{r} \underline{P} \underline{b} \underline{b}^T \underline{P} + \underline{P} \underline{A} - \frac{1}{r} \underline{P} \underline{b} \underline{b}^T \underline{P} + \underline{Q} + \frac{1}{r} \underline{P} \underline{b} \underline{b}^T \underline{P} = 0$$

$$\underline{A}^T \underline{P} + \underline{P} \underline{A} - \frac{1}{r} \underline{P} \underline{b} \underline{b}^T \underline{P} + \underline{Q} = 0$$

1.5 Zustandsregler durch Optimierung (LQ)

LQ-Reglerentwurf



1.5 Zustandsregler durch Optimierung (LQ)

Neben der algebraischen Riccati-Gleichung, im Englischen oft mit **ARE** (*algebraic Riccati equation*) abgekürzt gibt es auch die Riccati-Differentialgleichung, im Englischen oft mit **DRE** (*differential Riccati equation*) abgekürzt. Bei der DRE tritt anstelle der "0" die Ableitung der \underline{P} -Matrix:

$$\underline{A}^T \underline{P}(t) + \underline{P}(t) \underline{A} - \frac{1}{r} \underline{P}(t) \underline{b} \underline{b}^T \underline{P}(t) + \underline{Q} = -\dot{\underline{P}}(t)$$

Die \underline{P} -Matrix folgt also einer DGL. Eine solche DRE entsteht, wenn der Zeithorizont im Integral der Verlustfunktion endlich ist. Dann ergibt sich eine *zeitvariante* \underline{P} -Matrix und damit auch ein zeitvarianter optimaler Regler. Für den stationären Fall ist $\underline{P}(t)$ eingeschwungen und daher $\dot{\underline{P}}(t) = 0$. Dann geht die DRE in die ARE über.

1.5 Zustandsregler durch Optimierung (LQ)

Wahl der Wichtungsmatrix Q und der Stellgrößenbestrafung r

- Das Verhältnis zwischen den Einträgen in Q und r bestimmt die Schnelligkeit der Regelung. Kleine Werte von r machen die Regelung schneller, erfordern aber auch größerer Stellgrößen (die auch physikalisch realisierbar sein müssen!). Größere Werte von r verlangsamen den geschlossenen Regelkreis, schonen aber das Stellglied.
- Das Verhältnis der Einträge in Q untereinander bestimmt, welche Zustände wie schnell gegen "0" gefahren werden. Im Allgemeinen ist es völlig ausreichend, wenn Q als *Diagonalmatrix* gewählt wird; meist hat man sowieso keine tieferen Einsichten für eine sinnvolle Wahl der anderen Matrixelemente.
- Q muss so gewählt werden, dass (\underline{A}, Q) *beobachtbar* ist (hierfür ist eigentlich die Erweiterung des Beobachtbarkeitsbegriffs auf Mehrgrößensysteme notwendig). Das ist leicht zu erreichen, wenn Q als Diagonalmatrix ohne Nullen in der Diagonale gewählt wird. Die Beobachtbarkeit stellt sicher, dass wirklich alle Zustände in die Verlustfunktion eingehen. Andernfalls könnte man nicht garantieren, dass alle Zustände auch gegen 0 konvergieren.

1.5 Zustandsregler durch Optimierung (LQ)

Eigenschaften *optimaler* Zustandsregler

- Der Regler ist *stabil*.
- Der Regler hat einen *Phasenrand* von mindestens 60° :

$$\text{Phasenrand: } \varphi_R \geq 60^\circ$$

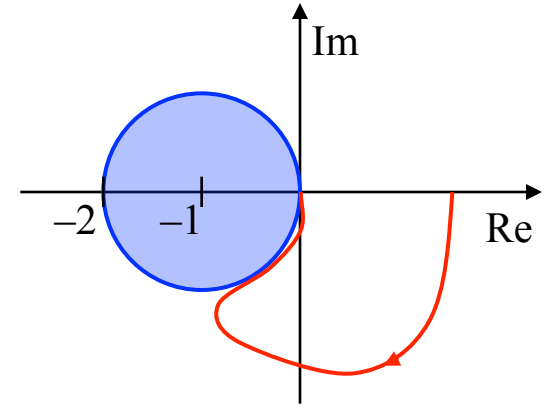
- Die Eigenwerte des geschlossenen Kreises $\bar{\lambda}_i$ sind im Mittel schneller (haben betragsmäßig den größeren Realteil) als die Eigenwerte des offenen Kreises λ_i :

$$\prod_{i=1}^n |\bar{\lambda}_i| \geq \prod_{i=1}^n |\lambda_i|$$

- Der Regler hat PD_{n-1} -Verhalten.
- Die Übertragungsfunktion des offenen Kreises ist gegeben durch

$$G_0(s) = \underline{k}^T (s\underline{I} - \underline{A})^{-1} \underline{b}$$

und weist einen *Polüberschuss* von genau 1 auf! Deshalb gilt hier nicht das Gleichgewichtstheorem nach Bode und der Phasenrand von 60° bleibt auch für hohe Verstärkungen ($r \rightarrow 0$) erhalten.



1.5 Zustandsregler durch Optimierung (LQ)

Erweiterung auf Mehrgrößensysteme

Für mehr als einen Eingang und Ausgang läuft die Berechnungen des optimalen Reglers völlig analog. Die Verlustfunktion und die Reglergleichung lauten dann:

$$J = \int_0^{\infty} [\underline{x}(t)^T \underline{Q} \underline{x}(t) + \underline{u}(t)^T \underline{R} \underline{u}(t)] dt \quad \text{mit} \quad \underline{u}(t) = -\underline{K} \underline{x}(t)$$

Die Ljapunow-Gleichung des unregulierten Systems ändert sich nicht, die des geregelten Systems wird zu:

$$\underline{\bar{A}}^T \underline{P} + \underline{P} \underline{\bar{A}} = -\underline{\bar{Q}} \quad \text{mit} \quad \underline{\bar{A}} = \underline{A} - \underline{B} \underline{K} \quad \text{und} \quad \underline{\bar{Q}} = \underline{Q} + \underline{K}^T \underline{R} \underline{K}$$

Der optimale Regler berechnet sich aus:

$$\underline{K} = \underline{R}^{-1} \underline{B}^T \underline{P}$$

Und die Riccati-Gleichung heißt dann:

$$\underline{A}^T \underline{P} + \underline{P} \underline{A} - \underline{P} \underline{B} \underline{R}^{-1} \underline{B}^T \underline{P} + \underline{Q} = \underline{0}$$

1.5 Zustandsregler durch Optimierung (LQ)

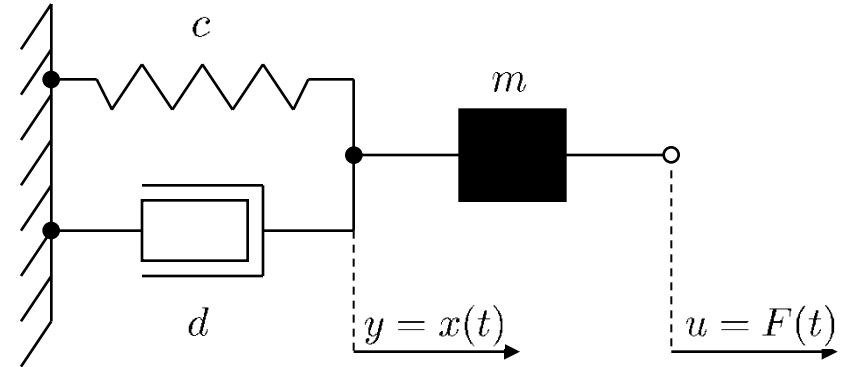
Beispiel: Optimale Regelung eines Feder-Masse-Dämpfer-Systems

Als Zustände für dieses System 1. Ordnung wählen wir Weg $x_1 = x$ und Geschwindigkeit $x_2 = \dot{x}$. Damit ergeben sich folgende Zustandsraumgleichungen:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{c}{m}x_1 - \frac{d}{m}x_2 + \frac{1}{m}u$$

$$y = x_1$$



$$m\ddot{x} + d\dot{x} + cx = F$$

$$m\dot{x}_2 + dx_2 + cx_1 = u$$

Das entspricht folgenden Matrizen/Vektoren:

$$\underline{A} = \begin{pmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{d}{m} \end{pmatrix} \quad \underline{b} = \begin{pmatrix} 0 \\ \frac{1}{m} \end{pmatrix}$$

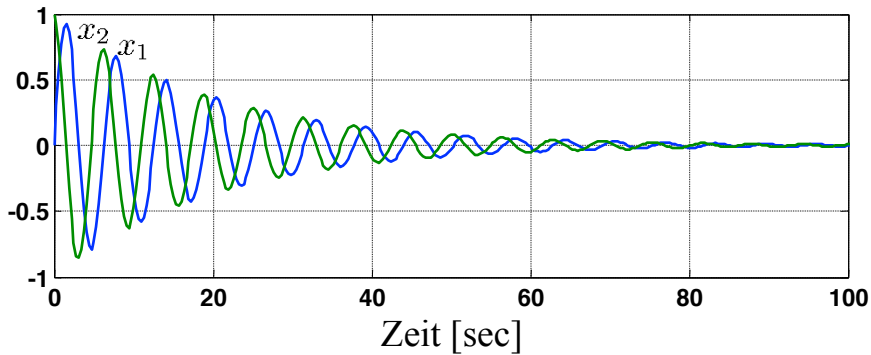
$$\underline{c}^T = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Im Folgenden wählen wir als einfaches Zahlenbeispiel: $m = 1 \text{ kg}$, $c = 1 \text{ N/m}$, $d = 0,1 \text{ Nsec/m}$.

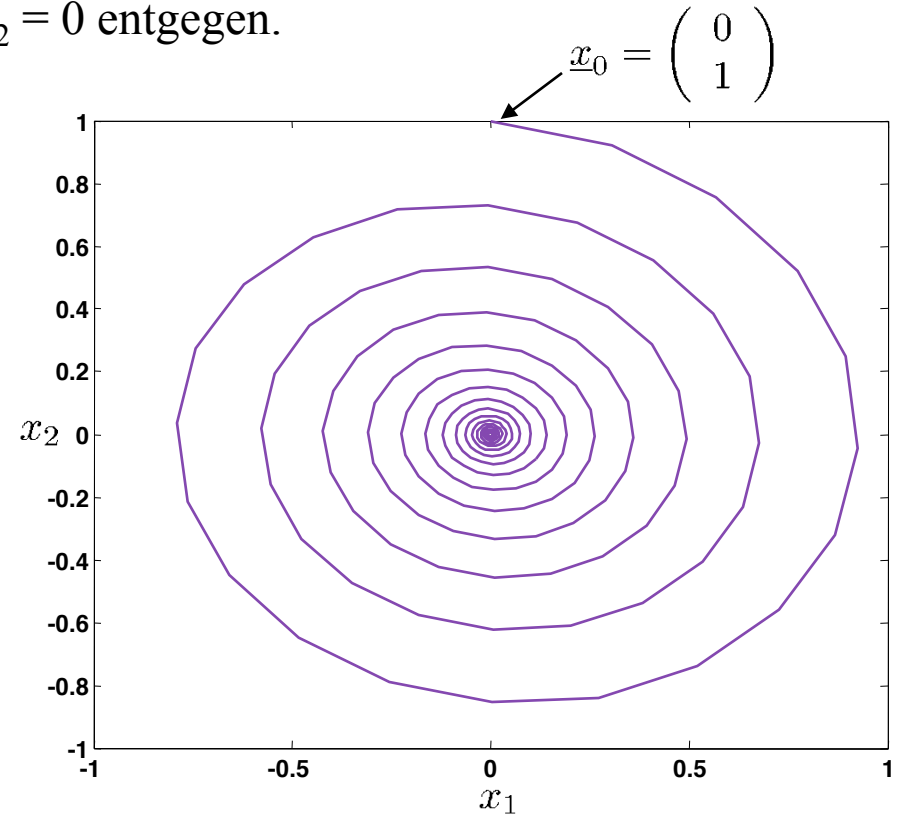
1.5 Zustandsregler durch Optimierung (LQ)

Beispiel: Eigenverhalten des Systems

Ohne äußere Krafteinwirkung, also für $F = 0$, schwingt das System von seinem Anfangszustand $x_1 = 0$ m und $x_2 = 1$ m/sec oszillatorisch, schwach gedämpft seinem Ruhepunkt im Ursprung des Zustandsraums bei $x_1 = 0$ und $x_2 = 0$ entgegen.



Dies ist die Antwort des Systems auf die gegebenen Anfangsbedingungen. Sie ist identisch mit der Impulsantwort, wie in Kapitel 1.9 ausführlich beschrieben wird.



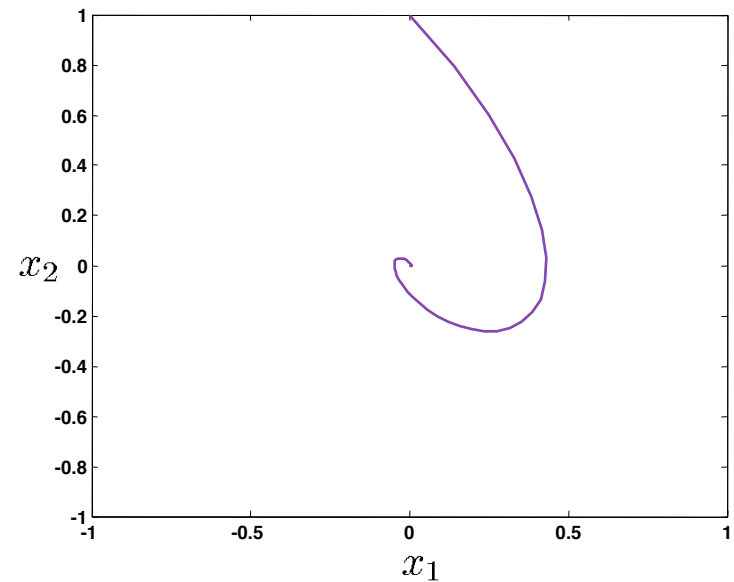
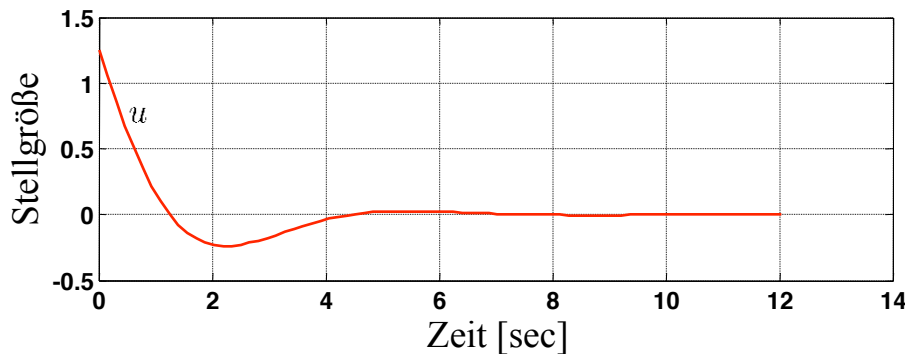
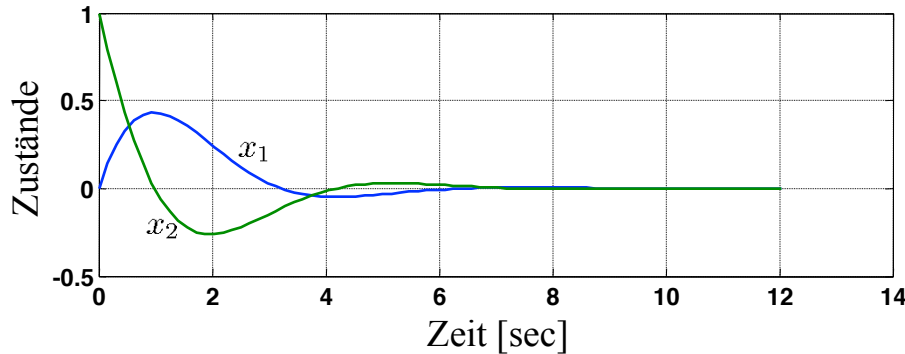
Pole : $-0,05 \pm i0,999$

1.5 Zustandsregler durch Optimierung (LQ)

Beispiel: Optimale Regelung eines Feder-Masse-Dämpfer-Systems

Entwurf eines LQ-Reglers zum Ausregeln der Anfangsbedingungen mit den folgenden

Gewichtungen $\underline{Q} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ und $r = 1$ liefert den optimalen Regler $\underline{k}^T = (0,414 \quad 1,256)$.



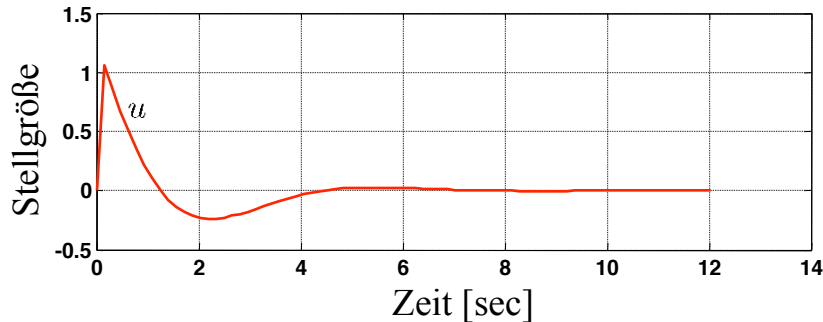
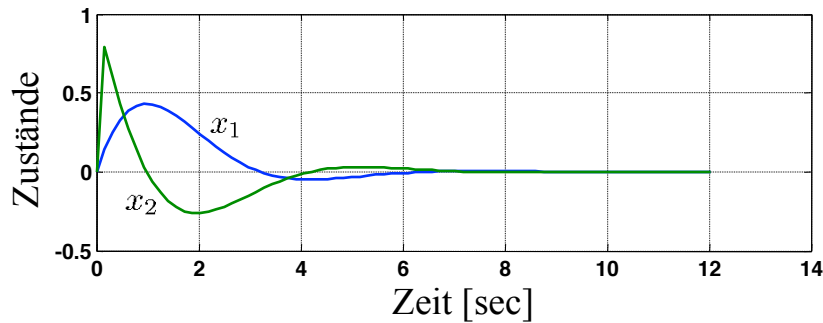
Pole : $-0,678 \pm i0,977$

1.5 Zustandsregler durch Optimierung (LQ)

Beispiel: Impuls und Sprung als Führungsgröße

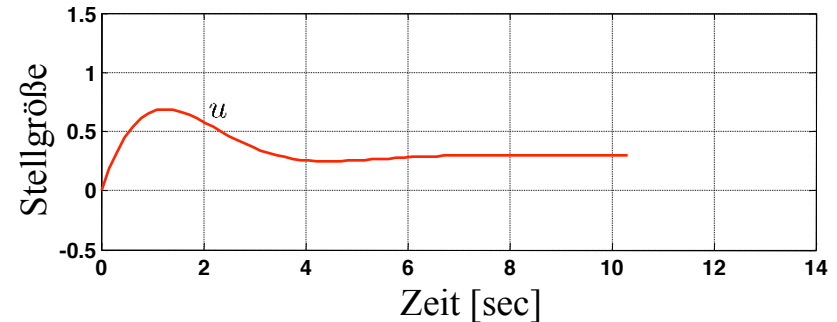
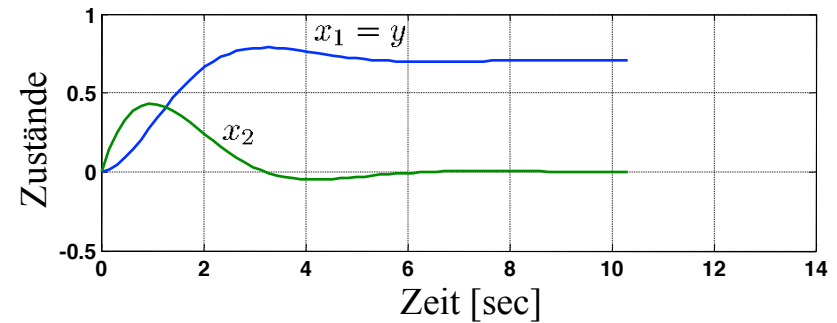
Ist der Regelkreis schon eingeschwungen, d.h. der Zustandsvektor auf $\underline{x} = \underline{0}$ gefahren, und wird durch eine Änderung in der Führungsgröße $w(t)$ angeregt, passiert Folgendes:

Impulsanregung $w(t) = \delta(t)$



Theoretisch identisches Ergebnis wie zuvor. Allerdings geringfügige Abweichungen wegen endlicher Schrittweite des Integrationsverfahrens.

Sprunganregung $w(t) = \sigma(t)$



Bleibende Regelabweichung ($y = x_1$), weil Zustandsregler keinen I-Anteil hat. Für Abhilfe siehe Kapitel 1.8.

1.5 Zustandsregler durch Optimierung (LQ)

Beispiel: Schnellere und langsamere Reglerauslegung $\underline{Q} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

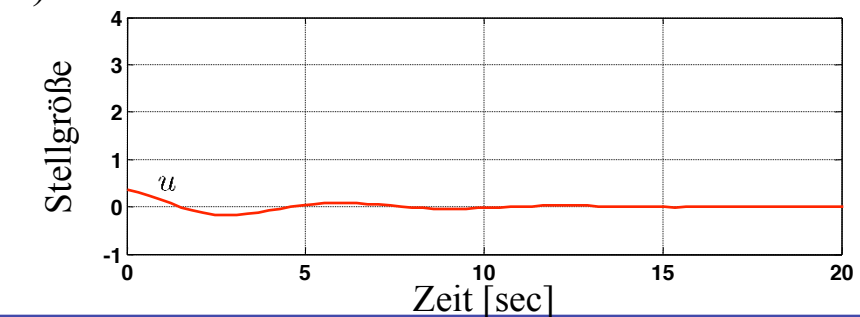
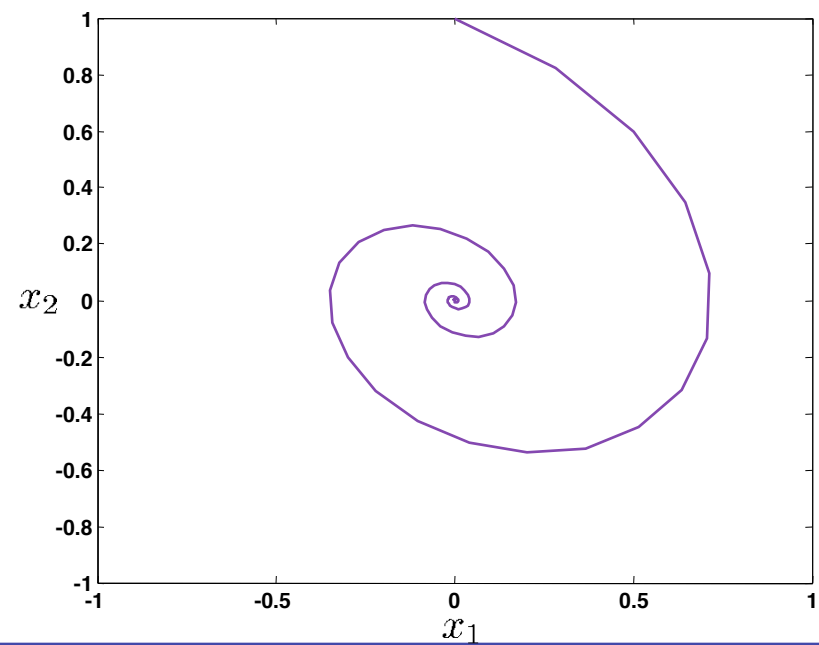
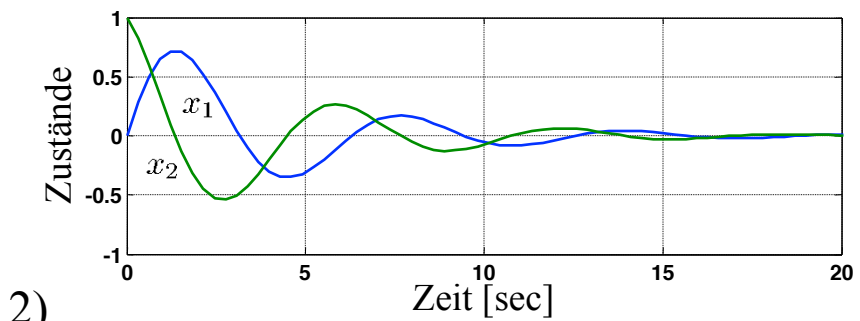
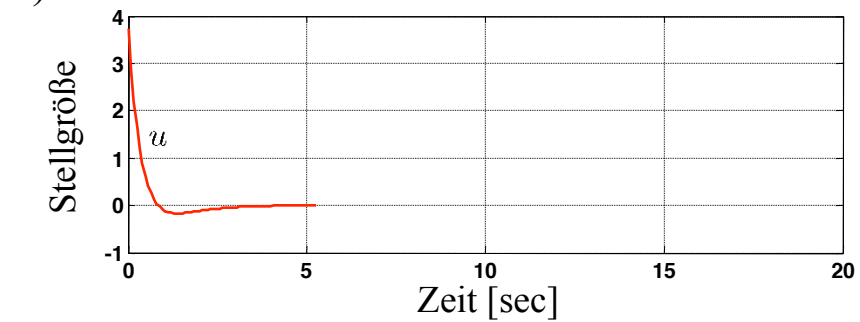
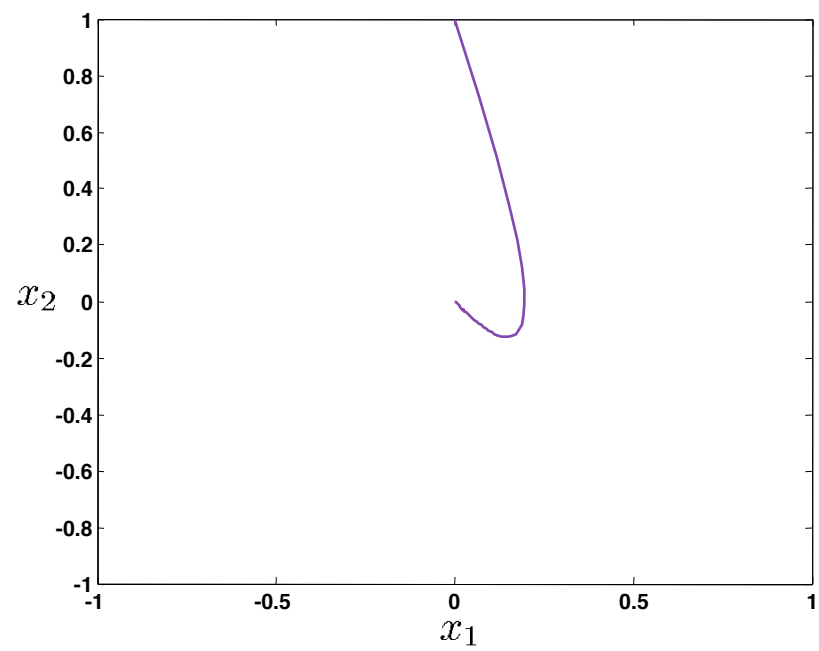
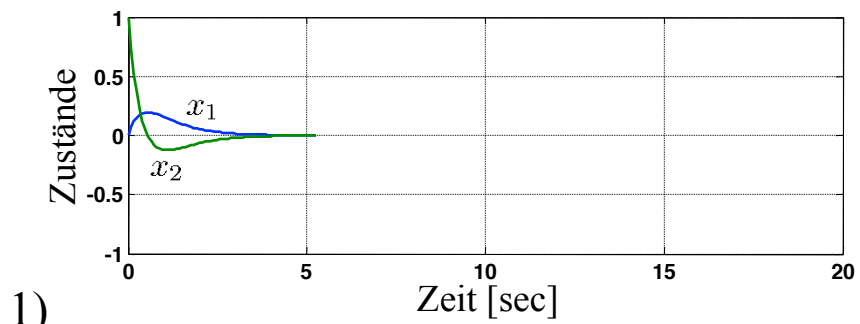
1) Schnellere Reglerauslegung (Faktor 10) durch geringere Bestrafung der Stellgröße

- Größere Reglerverstärkungen: $r = 0,1 \rightarrow \underline{k}^T = (2,317 \quad 3,727)$
- Größere (aggressivere) Stellgrößenwerte **Pole : $-1,327 / -2,5$**
- Mehr Leistungsaufwand, stärkere Stellgliedbeanspruchung
- Weniger Robustheit (kleinerer Phasenrand)

2) Langsamere Reglerauslegung (Faktor 10) durch stärkere Bestrafung der Stellgröße

- Kleinere Reglerverstärkungen: $r = 10 \rightarrow \underline{k}^T = (0,059 \quad 0,356)$
- Kleinere (trägere) Stellgrößenwerte **Pole : $-0,228 \pm i0,999$**
- Weniger Leistungsaufwand, schwächere Stellgliedbeanspruchung
- Mehr Robustheit (größerer Phasenrand)

Zum Vergleich: Originalregler: $r = 1 \rightarrow \underline{k}^T = (0,414 \quad 1,256)$ **Pole : $-0,678 \pm i0,977$**



1.5 Zustandsregler durch Optimierung (LQ)

Beispiel: Unterschiedliche Zustandsgewichtungen $r = 1$

1) Zustand 1 (Weg) wird stärker bestraft

- Größeres Verhältnis der Reglerverstärkungen für Zustand 1 zu Zustand 2:

$$\underline{Q} = \begin{pmatrix} 10 & 0 \\ 0 & 0,1 \end{pmatrix} \rightarrow \underline{k}^T = (2,317 \quad 2,078) \quad \text{Pole : } -1,089 \pm i1,460$$

- Weg wird schneller gegen 0 gefahren als mit Originalregler.

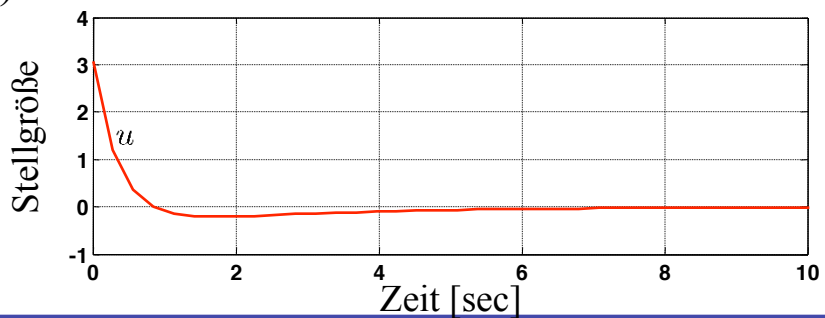
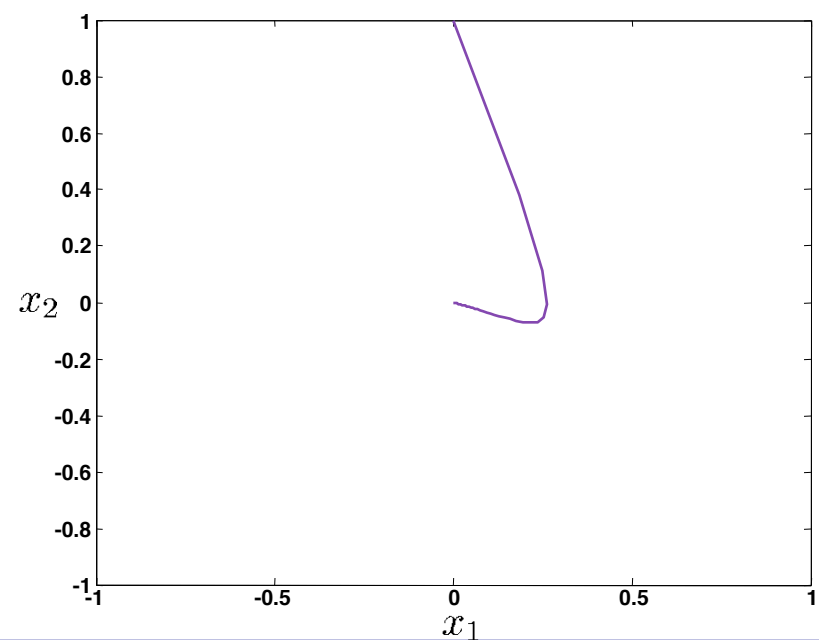
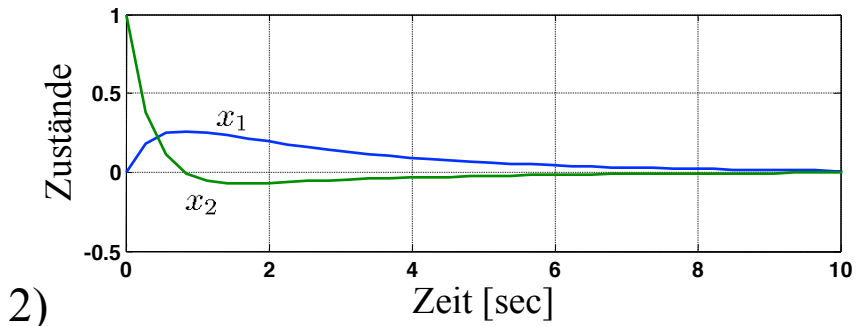
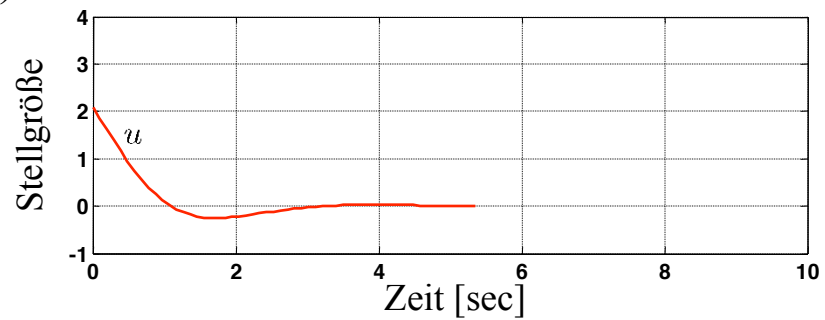
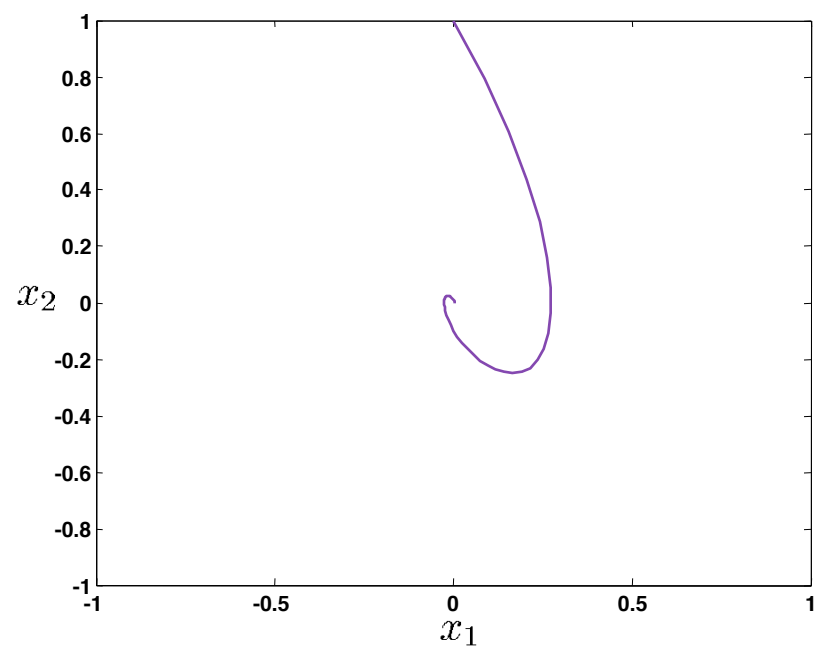
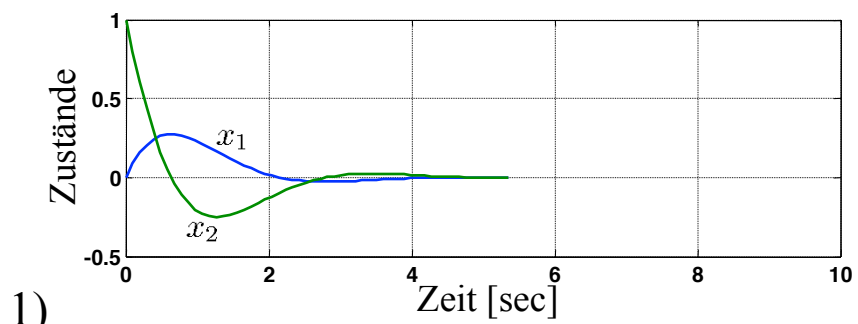
2) Zustand 2 (Geschwindigkeit) wird stärker bestraft

- Größeres Verhältnis der Reglerverstärkungen für Zustand 2 zu Zustand 1:

$$\underline{Q} = \begin{pmatrix} 0,1 & 0 \\ 0 & 10 \end{pmatrix} \rightarrow \underline{k}^T = (0,049 \quad 3,079) \quad \text{Pole : } -0,374 / -2,805$$

- Geschwindigkeit wird schneller gegen 0 gefahren als mit Originalregler.
- Dies ist schwierig und geht stark auf Kosten der Einregelzeit (ca. $2 \times$ so groß).

Originalregler: $\underline{Q} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \rightarrow \underline{k}^T = (0,414 \quad 1,256) \quad \text{Pole : } -0,678 \pm i0,977$



1.5 Zustandsregler durch Optimierung (LQ)

MATLAB



Reglerentwurf durch Optimierung

```
System = ss(A,b,c,d); % Definiere ein System im Zustandsraum
rank(ctrb(System))    % Prüfe dessen Steuerbarkeit (Ergebnis == n ?)
Q = diag([2 3 10]);  % Wichtigung für die Zustände
r = 1;                % Bestrafung für die Stellgröße
k = lqr(A,b,Q,r);    % Optimaler Reglerentwurf nach Matrix-Riccati
Ag = A - b*k;        % Systemmatrix des geschlossenen Kreises
eig(Ag)               % Eigenwerte des geschlossenen Kreises
```

Lösen einer Ljapunow-Gleichung

```
P = lyap(A',Q);      % Löst:  $A' * P + P * A = -Q$ 
```

1.6 Beobachter

Probleme bei der Realisierung eines Zustandsreglers:

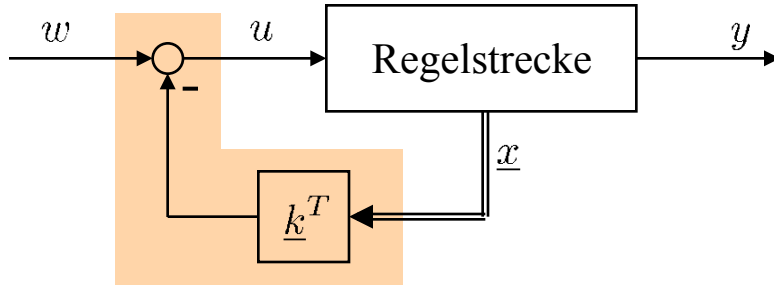
- *Alle* Zustände müssen gemessen werden!
- Man braucht also mindestens n Sensoren. → Aufwändig, teuer, evtl. unzuverlässig.
- Manche Zustände können evtl. gar nicht gemessen werden, weil z.B.: Messort räumlich nicht zugänglich oder Umgebungsbedingungen (Temperatur, Druck, Vibrationen, Schmutz, etc.) zu „feindlich“ sind.

Wie können wir das Konzept des Zustandsreglers retten?

- Beobachtung der Zustände!
- Rückkopplung der beobachteten Zustände anstelle der gemessenen Zustände.
- **Beobachter** = **virtueller Sensor**, d.h. die beobachteten Signale werden wie Messwerte verwendet, obwohl sie eigentlich (aus anderen Messsignalen) berechnete Größen sind.
- Benötigt ein gutes Modell der Regelstrecke.

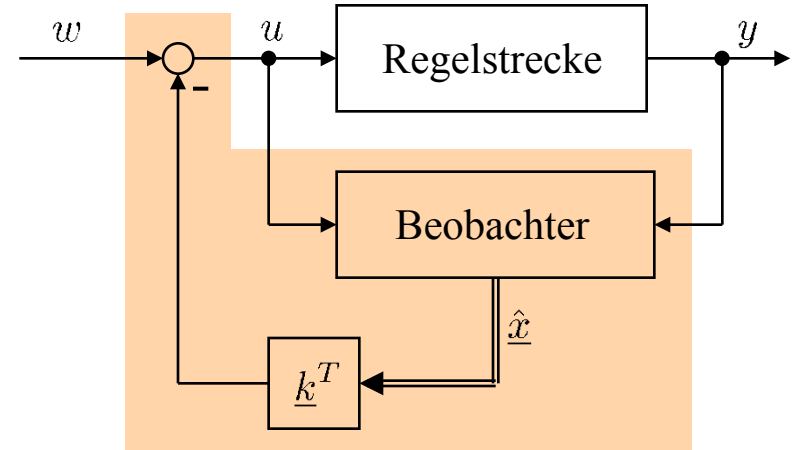
1.6 Beobachter

Zustandsregler



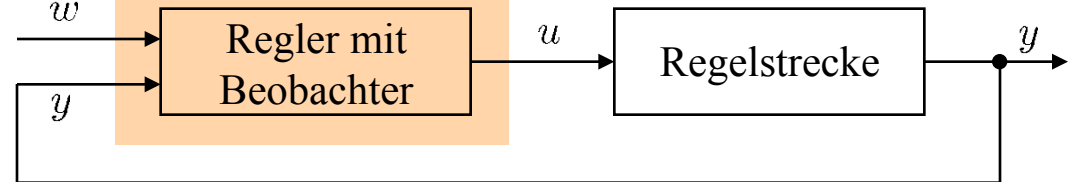
- Regler wird mit dem gemessenen n -dimensionalen Zustandsvektor \underline{x} gefüttert.

Zustandsregler mit Beobachter



Von Außen betrachtet ist das ein *Standardregelkreis*. Alles was mit "Zustandsraum" zu tun hat, findet in dem bunten Kasten statt.

→ Keine *prinzipiellen* Unterschiede zur Betrachtung mittels Übertragungsfunktionen.

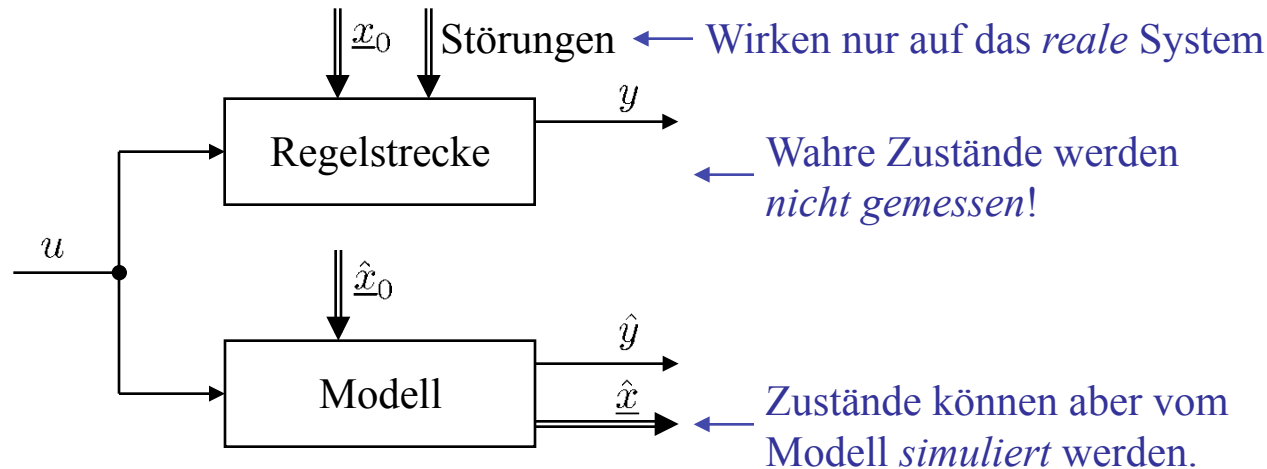


- Regler wird mit dem beobachteten n -dimensionalen Zustandsvektor $\hat{\underline{x}}$ gefüttert.
- ABER: Regler mit Beobachter basiert eigentlich nur auf der Messung zweier skalarer Größen: u und y .

1.6 Beobachter

Struktur eines Beobachters: Vorüberlegungen

Die einfachste und naheliegendste Idee für den Aufbau eines Zustandsbeobachters ist die Simulation der Regelstrecke mit einem Modell:



Nachteile dieser Vorgehensweise:

1. Die Anfangswerte \underline{x}_0 müssen zunächst aus den Messsignalen rekonstruiert werden. → Mühsam aber für beobachtbare Systeme möglich (Definition der Beobachtbarkeit).
2. Störungen wirken nur auf das reale System. Die dadurch verursachten Abweichungen zwischen \underline{x} und $\hat{\underline{x}}$ werden nie ausgeglichen. → $\hat{\underline{x}}$ entfernt sich immer weiter von \underline{x} .

1.6 Beobachter

Der 1. Nachteil ist der entscheidende. Dieser Nachteil ist in der Struktur der Simulation begründet:

Das Ausgangssignal y wird bei der Simulation (außer für die Rekonstruktion des Anfangswertes) nicht verwendet! So kann ein "Auseinanderlaufen" von realer Regelstrecke und Modell nicht festgestellt und verhindert werden.

Aus dieser Feststellung folgt unmittelbar folgende Idee für eine Verbesserung:

Das "Auseinanderlaufen" von realer Regelstrecke und Modell sollte festgestellt werden, indem man die Abweichung zwischen deren Ausgängen berechnet:

$$e_y(t) = y(t) - \hat{y}(t)$$

Ein Beobachter sollte so aufgebaut sein, dass diesem **Ausgangsfehler entgegen gewirkt wird. → **Prinzip Rückkopplung, analog zum Regelkreis!****

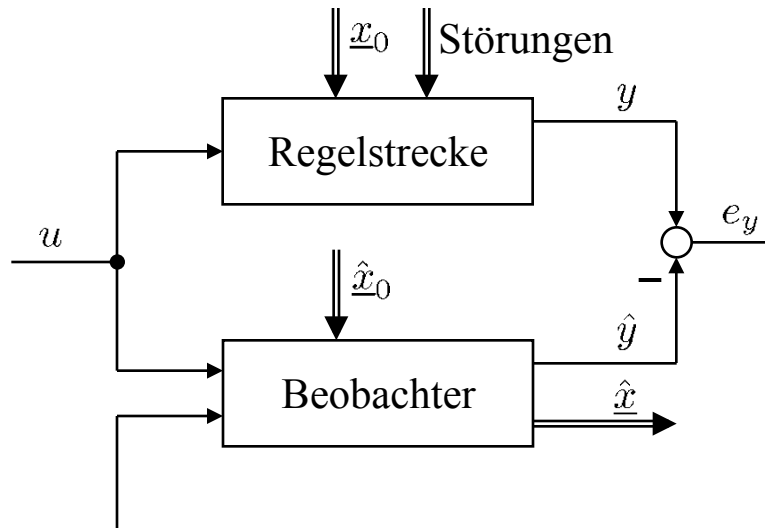
Der Ausgangsfehler hat im Wesentlichen zwei Ursachen:

- *Störungen*, d.h. nicht beeinflussbare Größen, die nur auf das reale System wirken.
- *Modellfehler*, d.h. Abweichung zwischen dem realen System und dessen Modell.

1.6 Beobachter

Struktur eines Beobachters

Die Struktur eines Beobachters nutzt das Prinzip eines Regelkreises, nämlich die *Rückkopplung*, für die Abgleichung des Ausgangs und der Zustände.



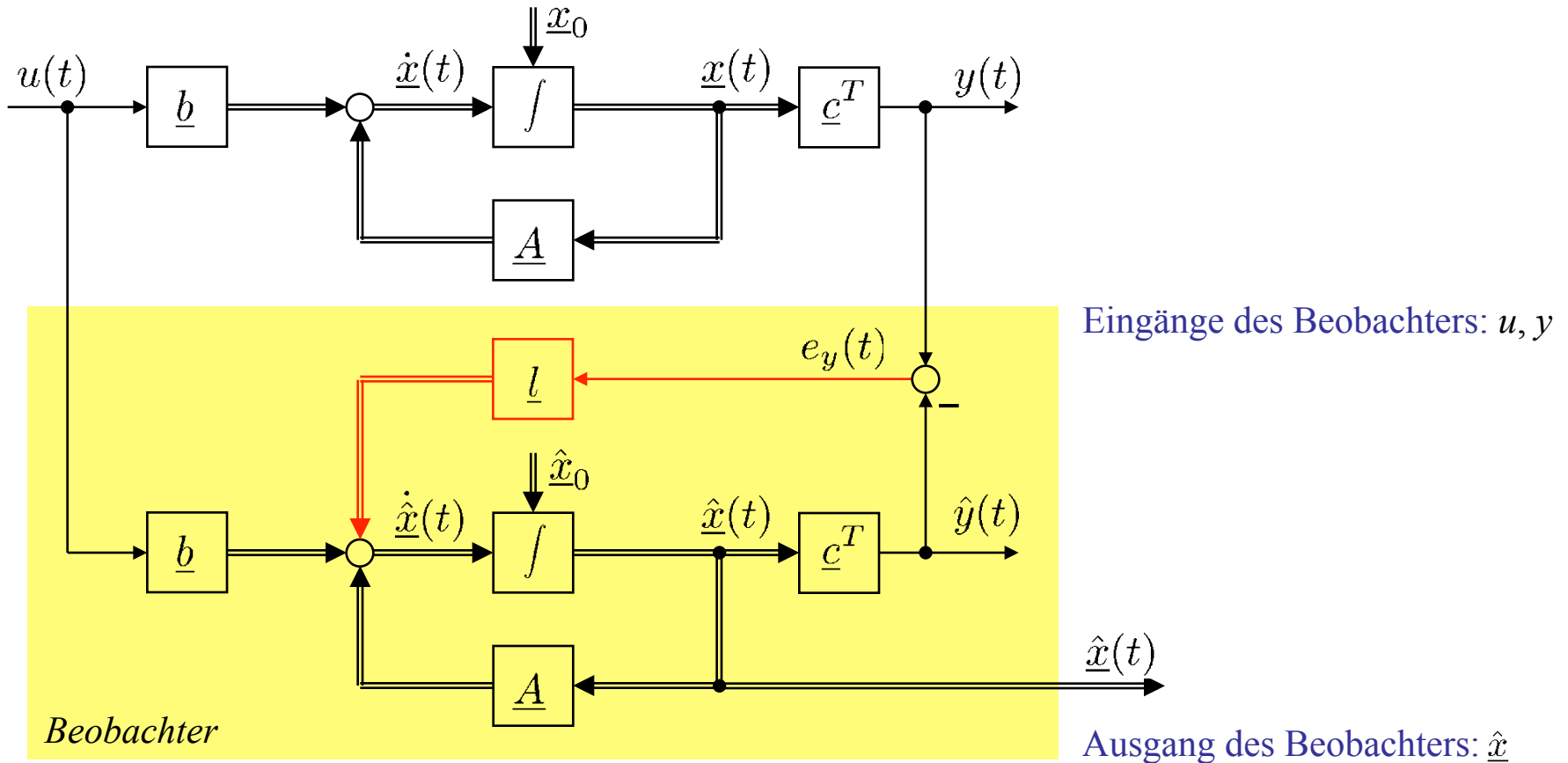
Im Beobachter läuft folgendes ab (bei ausschließlich positiven Verstärkungen):

- Ist $y > \hat{y} \rightarrow e_y > 0$ wird die Eingangsgröße in den Beobachter vergrößert, um die beobachteten Zustände und den Beobachteraussgang zu erhöhen.
- Ist $y < \hat{y} \rightarrow e_y < 0$ läuft es genau anders herum.

1.6 Beobachter

Luenberger-Beobachter

Der Beobachter nach Luenberger (1964) erweitert die Idee der Parallelschaltung von Regelstrecke und Modell genau um diese Rückkopplung.



1.6 Beobachter

Zustandsgleichungen für den Luenberger-Beobachter

$$\begin{aligned}\dot{\underline{\hat{x}}}(t) &= \underline{A} \underline{\hat{x}}(t) + \underline{b} u(t) + \underline{l} \underbrace{e_y(t)}_{= y(t) - \hat{y}(t)} \\ \hat{y}(t) &= \underline{c}^T \underline{\hat{x}}(t)\end{aligned}$$

Der Rückkopplungsvektor \underline{l} des Beobachters hat die Dimension $n \times 1$ und entspricht dem $1 \times n$ -Reglervektor \underline{k}^T des Zustandsreglers.

Um den Einfluss der Zustände des realen Systems und des Beobachters zu veranschaulichen, soll die 1. Zustandsgleichung nicht von den Ausgängen beeinflusst werden, sondern nur von den Zuständen und dem Eingang. Daher eliminieren wir $y(t) = \underline{c}^T \underline{x}(t)$ und $\hat{y}(t) = \underline{c}^T \underline{\hat{x}}(t)$:

$$\begin{aligned}\dot{\underline{\hat{x}}}(t) &= \underline{A} \underline{\hat{x}}(t) + \underline{b} u(t) + \underline{l} \underline{c}^T (\underline{x}(t) - \underline{\hat{x}}(t)) \\ \hat{y}(t) &= \underline{c}^T \underline{\hat{x}}(t)\end{aligned}$$

An dieser Gleichung erkennt man, dass sich die Rückkopplung nur dann bemerkbar macht, wenn eine Abweichung zwischen realem Zustand $\underline{x}(t)$ und beobachtetem Zustand $\underline{\hat{x}}(t)$ auftritt. Je größer diese Abweichung ist, desto stärker wird $\underline{\hat{x}}(t)$ geändert, um diese Abweichung auszugleichen. Der Rückkopplungsvektor \underline{l} stellt dabei die Geschwindigkeit dieses Ausgleichsvorgangs ein (entspricht der Reglerverstärkung).

1.6 Beobachter

Gleichungen für gemeinsames System aus Regelstrecke mit Beobachter

Die Gleichungen für die Regelstrecke und den Beobachter können bequem als ein "großes" Zustandsraumsystem mit $2n$ Zuständen (n Zustände für die Regelstrecke, n Zustände für den Beobachter) geschrieben werden.

Fassen wir folgende Zustandsgleichungen zusammen:

$$\text{Regelstrecke: } \underline{\dot{x}}(t) = \underline{A} \underline{x}(t) + \underline{b} u(t) \qquad y(t) = \underline{c}^T \underline{x}(t)$$

$$\text{Beobachter: } \underline{\dot{\hat{x}}}(t) = \underline{l} \underline{c}^T \underline{x}(t) + (\underline{A} - \underline{l} \underline{c}^T) \underline{\hat{x}}(t) + \underline{b} u(t) \qquad \hat{y}(t) = \underline{c}^T \underline{\hat{x}}(t)$$

Damit ergibt sich folgende Matrix/Vektor-Darstellung:

$$\begin{pmatrix} \underline{\dot{x}}(t) \\ \underline{\dot{\hat{x}}}(t) \end{pmatrix} = \begin{pmatrix} \underline{A} & \underline{0} \\ \underline{l} \underline{c}^T & \underline{A} - \underline{l} \underline{c}^T \end{pmatrix} \begin{pmatrix} \underline{x}(t) \\ \underline{\hat{x}}(t) \end{pmatrix} + \begin{pmatrix} \underline{b} \\ \underline{b} \end{pmatrix} u(t)$$

$$\begin{pmatrix} y(t) \\ \hat{y}(t) \end{pmatrix} = \begin{pmatrix} \underline{c}^T & \underline{0} \\ \underline{0} & \underline{c}^T \end{pmatrix} \begin{pmatrix} \underline{x}(t) \\ \underline{\hat{x}}(t) \end{pmatrix}$$

1.6 Beobachter

Beobachtergleichungen

Messen können wir nur den Eingang u und den Ausgang y , nicht aber die Zustände \underline{x} . Die Realisierung des Beobachters muss mit den Messgrößen auskommen. Daher nutzen wir wieder die Beziehung $y(t) = \underline{c}^T \underline{x}(t)$, um die Zustände der realen Regelstrecke zu eliminieren:

$$\text{Beobachtergleichungen: } \dot{\hat{\underline{x}}}(t) = (\underline{A} - \underline{l}\underline{c}^T) \hat{\underline{x}}(t) + \underline{b}u(t) + \underline{l}y(t)$$

Die Ausgangsgleichung ist nur zur Berechnung des Ausgangsfehlers e_y wichtig und wird sonst nicht weiter benötigt.

Die Beobachtergleichungen bedeuten:

- Der Rückkopplungsvektor \underline{l} ändert die Eigendynamik des Beobachters in einer ähnlichen Weise wie der Zustandsregler \underline{k}^T die Eigendynamik der Regelstrecke verändert.
- Der Entwurf des Beobachters kann analog zum Zustandsregler mittels Polvorgabe oder Optimierung nach Matrix-Riccati geschehen.
- Der Beobachterentwurf ist das duale Problem zum Reglerentwurf und daher können die bisher gelernten Methoden Verwendung finden!

1.6 Beobachter

Beobachterfehler

Die Beobachtergleichung kann wie folgt umgeschrieben werden. Führen wir den **Beobachterfehler** ein, also die Abweichung des beobachteten vom realen Zustandsvektor:

$$\underline{e}_x(t) = \underline{x}(t) - \hat{\underline{x}}(t)$$

Die Änderungsrate dieses Beobachtungsfehlers ist damit:

$$\begin{aligned}\dot{\underline{e}}_x(t) &= \dot{\underline{x}}(t) - \dot{\hat{\underline{x}}}(t) = \underline{A}\underline{x}(t) + \underline{b}u(t) - [\underline{A}\hat{\underline{x}}(t) + \underline{b}u(t) + \underline{l}\underline{c}^T(\underline{x}(t) - \hat{\underline{x}}(t))] \\ &= (\underline{A} - \underline{l}\underline{c}^T)(\underline{x}(t) - \hat{\underline{x}}(t))\end{aligned}$$

D.h. der Beobachtungsfehler folgt völlig *unabhängig* vom *Verlauf der Signale* $u(t)$ und $y(t)$ dieser Differentialgleichung:

Beobachtungsfehler: $\dot{\underline{e}}_x(t) = (\underline{A} - \underline{l}\underline{c}^T) \underline{e}_x(t)$

Der Beobachtungsfehler wird also durch ein *nicht steuerbares* System beschrieben (wir haben durch äußere Signale keinerlei Einfluss auf das Systemverhalten). Wenn die Matrix $\underline{A} - \underline{l}\underline{c}^T$ stabil ist, dann *klingt* jede Anfangsabweichung $\underline{e}_x(0) = \underline{x}_0 - \hat{\underline{x}}_0$ *exponentiell* schnell *ab*. Das Gleiche gilt für impulsartige Störungen $d_0\delta(t)$ zu beliebigen Zeitpunkten.

1.6 Beobachter

Dualität zwischen Zustandsregler und Beobachter

Für den Entwurf eines Beobachters muss der Rückkopplungsvektor \underline{l} so bestimmt werden, dass die Systemmatrix des Beobachters

$$\underline{A} - \underline{l}\underline{c}^T$$

die gewünschte Dynamik aufweist. Da eine Matrix \underline{M} und ihre Transponierte \underline{M}^T die selben Eigenwerte haben, können wir alternativ auch die Dynamik für folgende Matrix bestimmen:

$$(\underline{A} - \underline{l}\underline{c}^T)^T = \underline{A}^T - \underline{c}\underline{l}^T$$

Vom Entwurf eines Zustandsreglers wissen wir schon, wie man den Reglervektor \underline{k}^T berechnen kann, um die Dynamik folgender Matrix zu bestimmen:

$$\underline{A} - \underline{b}\underline{k}^T$$

Die beiden Probleme sind also zueinander **dual**:

	$\underline{k}^T \leftrightarrow \underline{l}^T$	
Zustandsregler	$\underline{A} \leftrightarrow \underline{A}^T$	Beobachter
	$\underline{b} \leftrightarrow \underline{c}$	

1.6 Beobachter

Konsequenzen der Dualität

Die Dualität zwischen Zustandsregler und Beobachter bedeutet, dass wir jedes Entwurfsverfahren für Zustandsregler auch für den Entwurf von Beobachtern nutzen können.

Statt \underline{A} und \underline{b} als Eingangsinformation verwenden wir \underline{A}^T und \underline{c} ; statt dem Reglervektor \underline{k}^T erhalten wir den Rückkopplungsvektor des Beobachters \underline{l}^T als Ergebnis.

Beobachterentwurf

- Optimaler Regler: Beim Entwurf nach Matrix-Riccati muss man nur den oben beschriebenen Tausch der Matrix und der Vektoren vornehmen.
- Polvorgabe: Die Polvorgabe ist besonders einfach, wenn die Zustandsgleichungen in Beobachternormalform vorliegen. Der Wechsel ins duale System führt nämlich von der Beobachter- in die Regelungsnormalform, wo wir die bereits bekannte Polvorgabeformel verwenden können:

$$\text{Beobachterrückführung: } \underline{l}^T = (n_0 \quad n_1 \quad \cdots \quad n_{n-1}) - (a_0 \quad a_1 \quad \cdots \quad a_{n-1})$$

Erzeugt das charakteristische Polynom $s^n + n_{n-1}s^{n-1} + \dots + n_1s + n_0$.

1.6 Beobachter

Wichtige Bemerkungen zum Beobachterentwurf


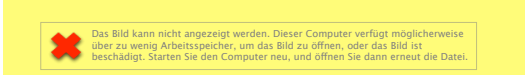
- Der Entwurf eines Zustandsreglers setzt Steuerbarkeit des Systems $(\underline{A}, \underline{b})$ voraus. Damit ist klar, dass wegen der Dualität für den Entwurf eines Beobachters die *Beobachtbarkeit* des Systems $(\underline{A}^T, \underline{c}^T)$ voraus gesetzt werden muss.
Es ist auch anschaulich verständlich, dass sich alle Zustände im Ausgangssignal "bemerktbar" machen müssen, denn nur die Differenz zwischen den Ausgangssignalen y und \hat{y} wird zurückgekoppelt. Und aus dieser Differenz soll ja gerade der Abgleich der einzelnen Zustände stattfinden.
- Die *Eigenwerte des Beobachters* sollten *deutlich schneller* sein als die dominierenden Eigenwerte des beobachtbaren Systems. D.h. die Eigenwerte von $\underline{A} - \underline{l}\underline{c}^T$ sollten in der komplexen Ebene deutlich links der dominierenden (mit größtem Realteil) Eigenwerte von \underline{A} liegen.
Mit einer solchen Wahl ist sichergestellt, dass der Beobachterfehler deutlich schneller abklingt als die Eigendynamik des Systems.
- Im Gegensatz zur realen Regelstrecke ist der Beobachter eine reine Simulation. Daher müssen wir bei der Wahl der Eigenwerte des Beobachters nicht auf Stellgrößenbeschränkungen oder Schonung des Stellglieds oder den Energieverbrauch achten!

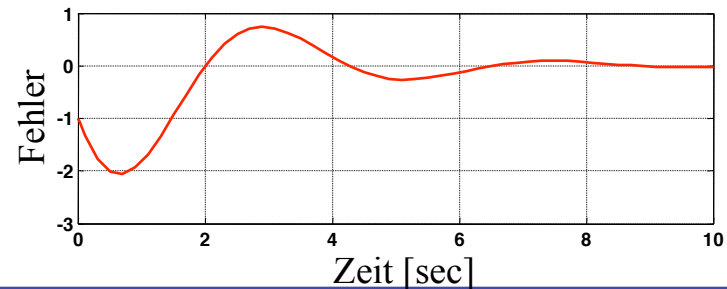
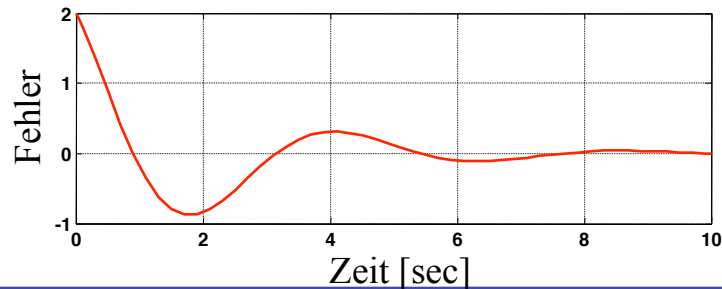
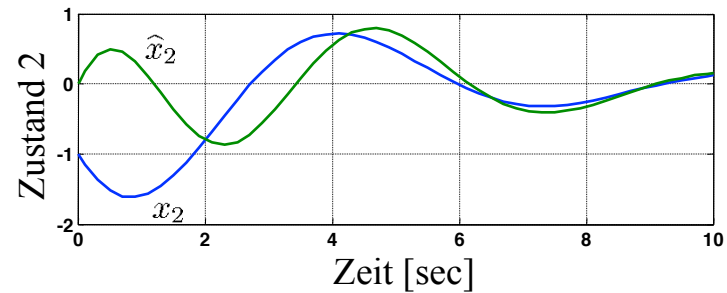
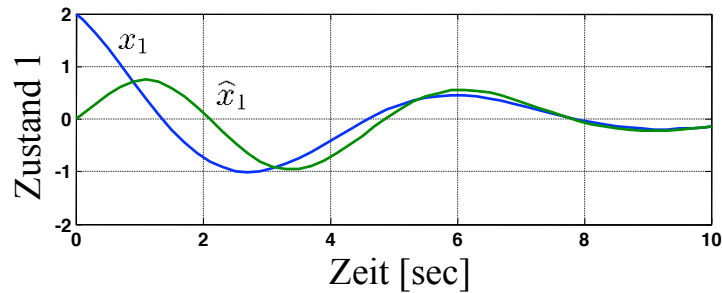
1.6 Beobachter

Beispiel: Optimaler Beobachterentwurf

Als Prozess dient das Feder-Masse-Dämpfer-System aus Kapitel 1.5, allerdings stärker gedämpft mit: $m = 1 \text{ kg}$, $c = 1 \text{ N/m}$, $d = 0,5 \text{ Nsec/m}$.

Pole : $-0,678 \pm i0,977$

- Anfangsbedingung $\underline{x}_0 = [2 \text{ m} \quad -1 \text{ m/sec}]^T$.
- Optimaler Beobachterentwurf mit Gewichtungen $\underline{Q} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ und $r = 1$ liefert die Rückführvektor 
- Die Beobachterpole, also die Eigenwerte von $\underline{A} - \underline{l} \underline{c}^T$, sind: 

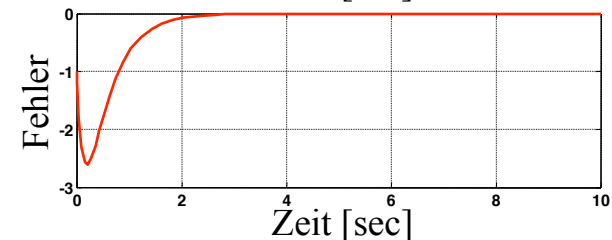
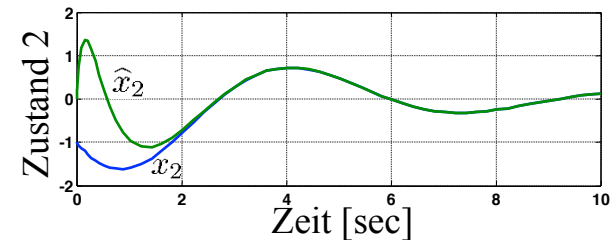
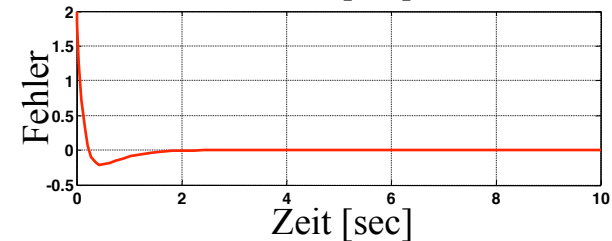
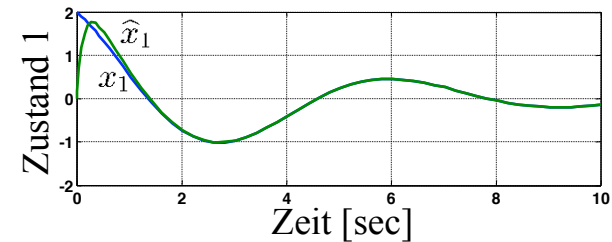
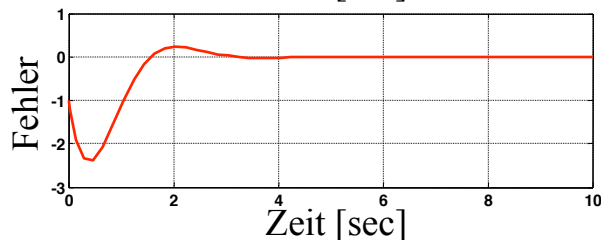
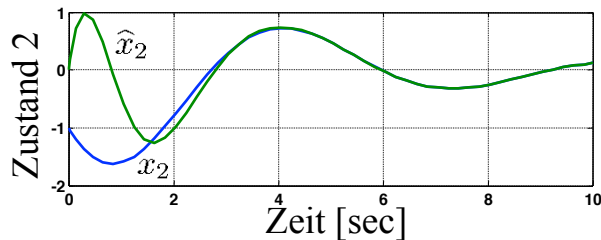
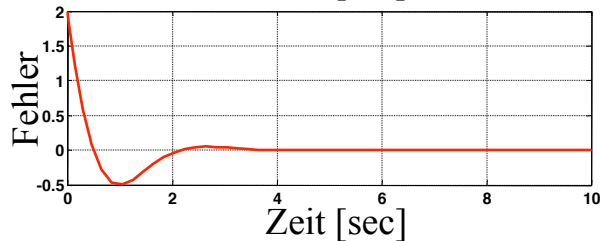
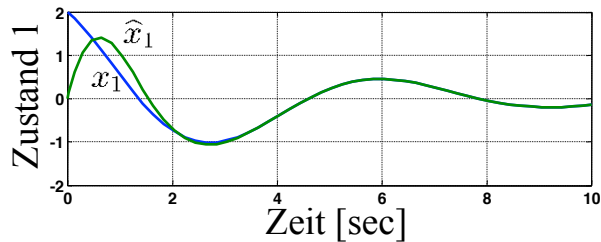


1.6 Beobachter

Beispiel: Beobachter beschleunigen durch Reduktion von r

 Das Bild kann nicht angezeigt werden. Dieser Computer verfügt möglicherweise über zu wenig Arbeitsspeicher, um das Bild zu öffnen, oder das Bild ist beschädigt. Starten Sie den Computer neu, und öffnen Sie dann erneut die Datei. Wenn weiterhin das rote x angezeigt wird, müssen Sie das Bild möglicherweise löschen und dann erneut einfügen.

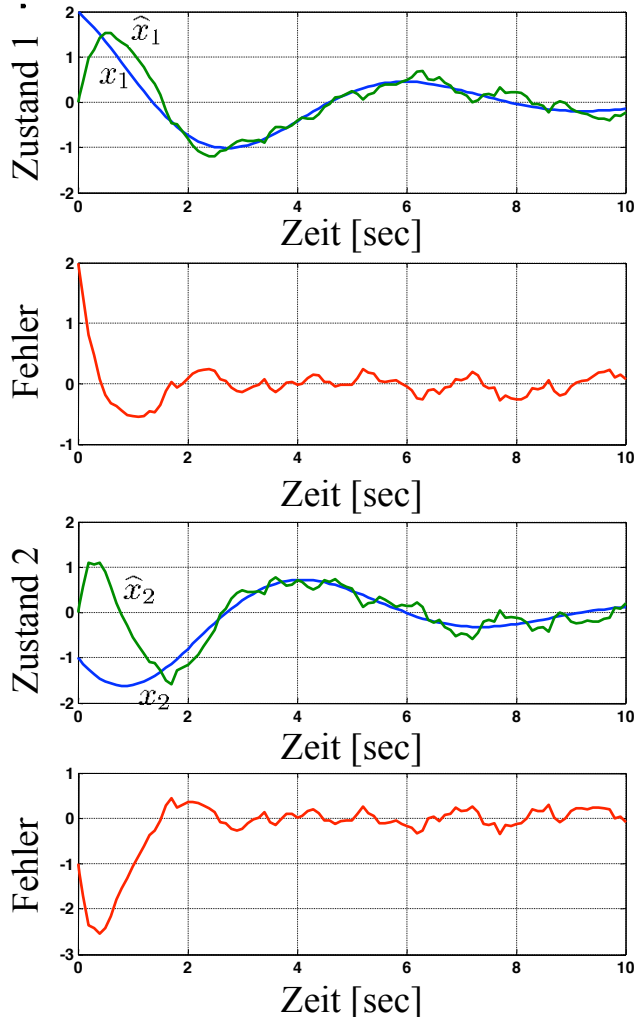
$$r = 0,01 \rightarrow \underline{l}^T = (9,050 \quad 10,38)$$



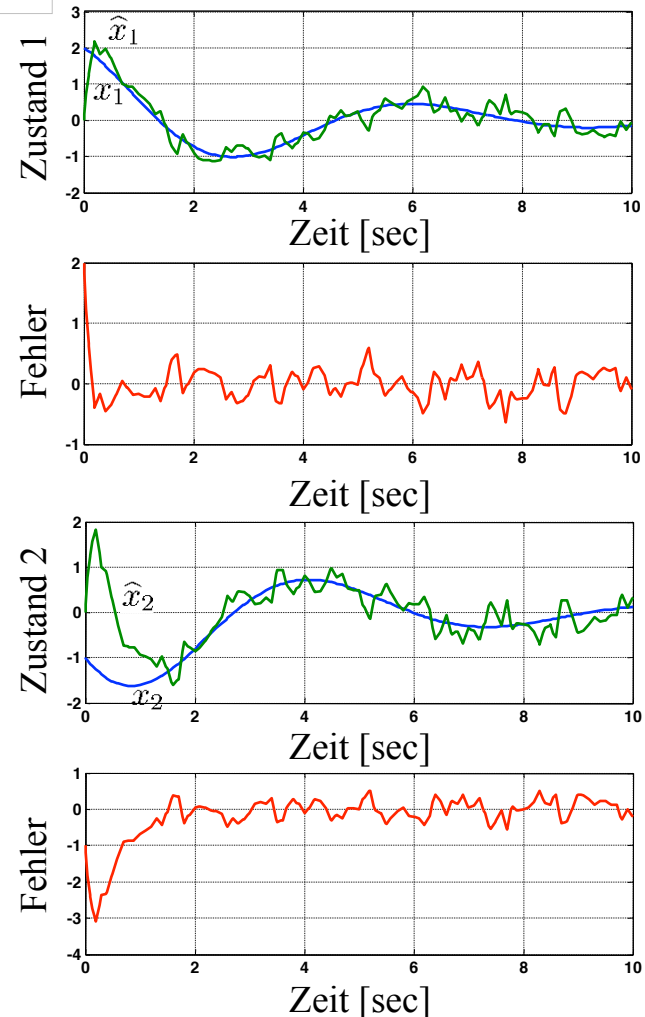
1.6 Beobachter

Beispiel: Beobachterempfindlichkeit in Bezug auf verrauschtes Ausgangssignal

$r = 0,1 :$



Das Bild kann nicht angezeigt werden. Dieser Computer



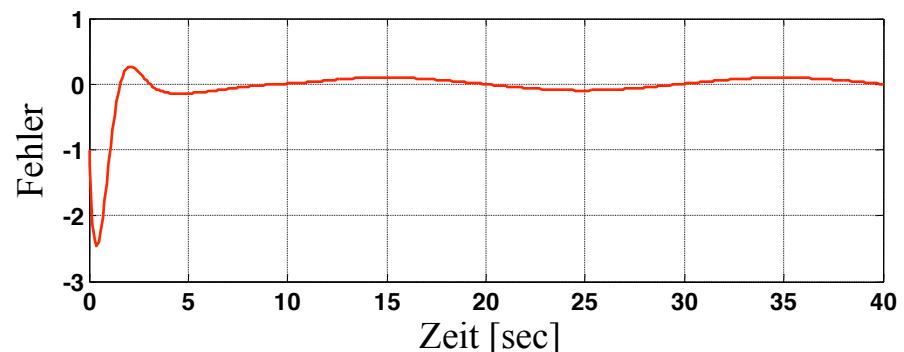
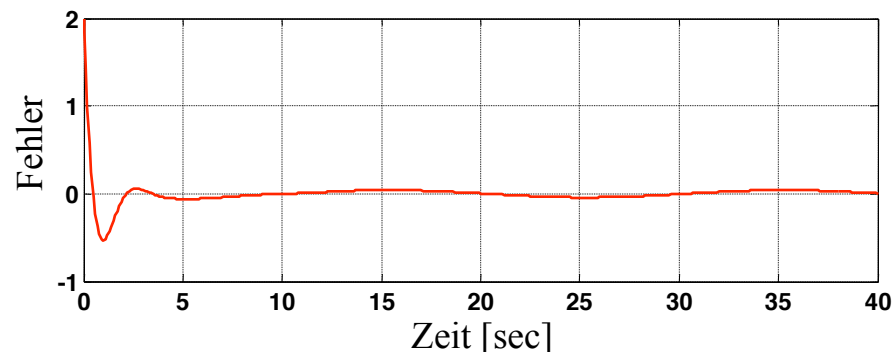
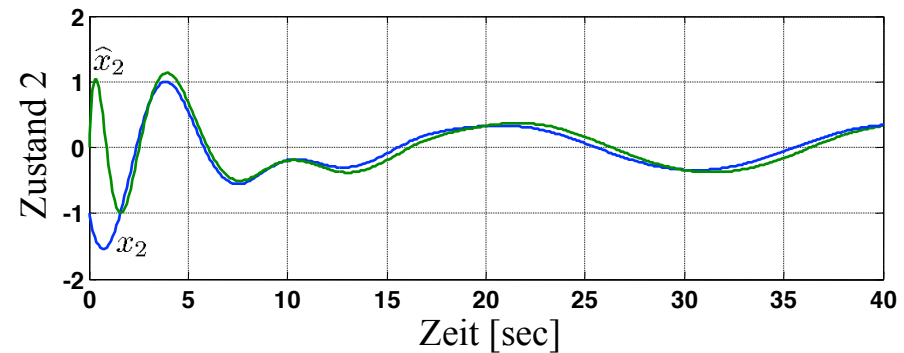
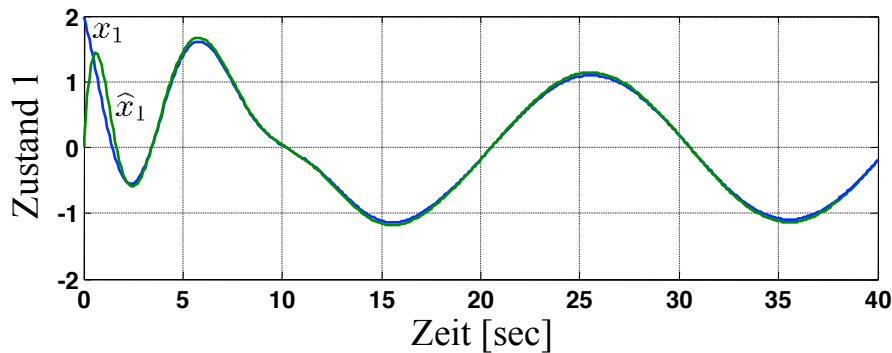
1.6 Beobachter

realer Prozess

Modell verwendet
in Beobachter

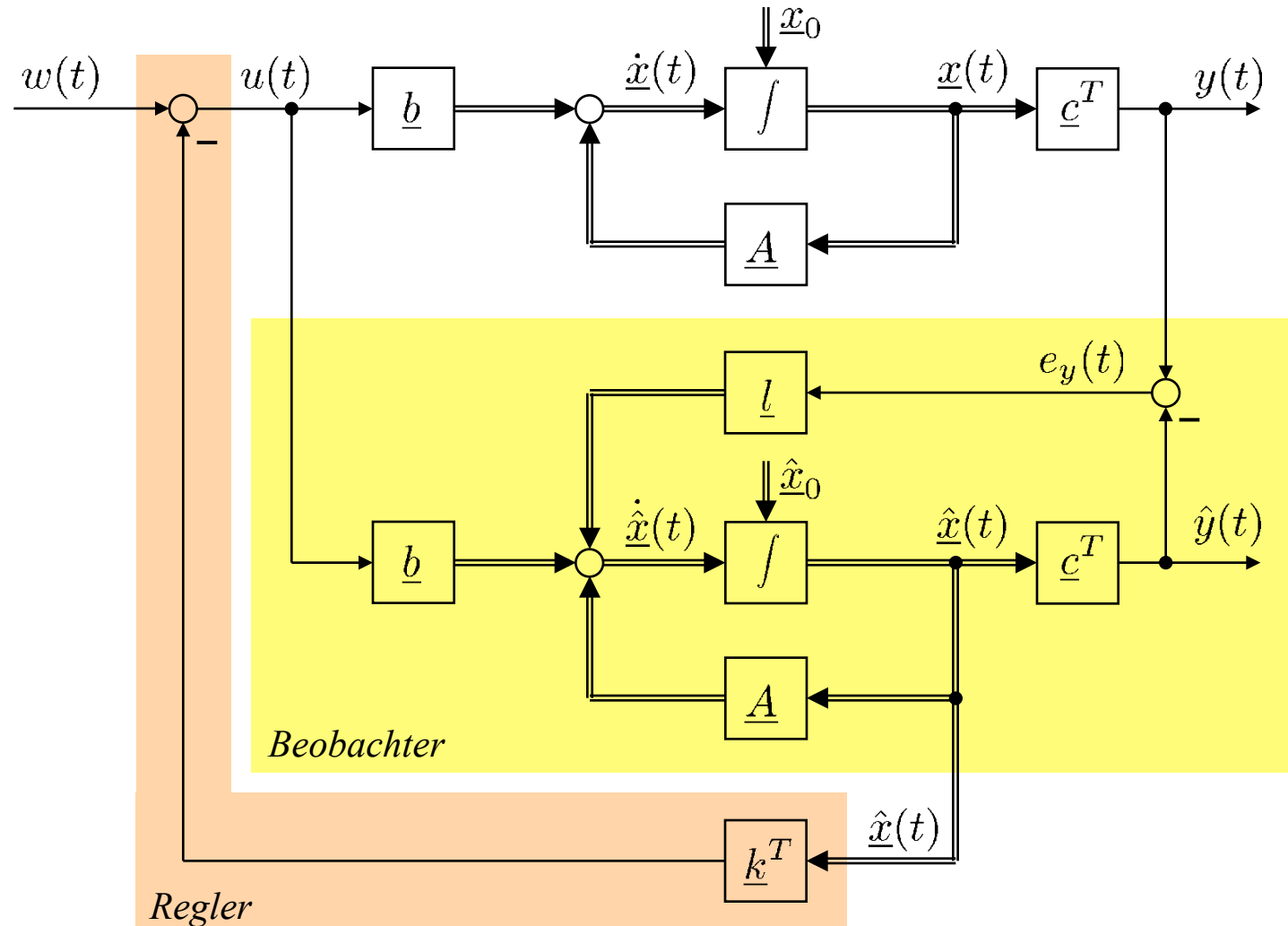
Beispiel: Beobachter bei 10%igen Modellfehlern (mit Sinus-Anregung)

- Modellfehler in \underline{A} und \underline{b} : $\underline{A} = \begin{pmatrix} 0 & 1 \\ -1 & -0,5 \end{pmatrix}$, $\underline{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $\hat{\underline{A}} = \begin{pmatrix} 0 & 1 \\ -0,9 & -0,4 \end{pmatrix}$, $\hat{\underline{b}} = \begin{pmatrix} 0 \\ 1,1 \end{pmatrix}$
- Ohne bleibende Anregung fällt der Beobachterfehler exponentiell auf 0 ab.
- Selbst mit Anregung $u(t) = \sin \omega t$ ist nur ein minimaler Beobachterfehler erkennbar!



1.7 Zustandsregler mit Beobachter (LQG)

Zustandsregler verwendet beobachtete statt gemessene Zustände



1.7 Zustandsregler mit Beobachter (LQG)

Gleichungen für Zustandsregler mit Beobachter

Nun wird im Regler \hat{x} statt x rückgekoppelt:

$$\dot{x} = \underline{A}x - \underline{b}k^T \hat{x} + \underline{b}w = \underline{A}x - \underbrace{\underline{b}k^T x + \underline{b}k^T x - \underline{b}k^T \hat{x}}_{\text{Wird addiert und subtrahiert, um auf die gewünschten Zustände zu kommen.}} + \underline{b}w$$

Zusammenfassen ergibt:

$$\dot{x} = \left(\underline{A} - \underline{b}k^T \right) x + \underbrace{\underline{b}k^T (x - \hat{x})}_{= e_x} + \underline{b}w$$

Gemeinsam mit der Beobachtergleichung $\dot{e}_x = (\underline{A} - \underline{l}c^T) e_x$ können wir folgende Zustandsdifferentialgleichungen mit $2n$ Zuständen (jeweils n Zustände in x und e_x) aufstellen:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{e}_x(t) \end{pmatrix} = \begin{pmatrix} \underline{A} - \underline{b}k^T & \underline{b}k^T \\ \underline{0} & \underline{A} - \underline{l}c^T \end{pmatrix} \begin{pmatrix} x(t) \\ e_x(t) \end{pmatrix} + \begin{pmatrix} \underline{b} \\ \underline{0} \end{pmatrix} w(t)$$

Aus Kapitel 1.6 wissen wir, dass die 1. Hälfte dieses Gleichungssystems, welche den Beobachterfehler e_x beschreibt, *nicht steuerbar* ist. Und aus Kapitel 1.3 kennen wir obige Struktur des Gleichungssystem als eine Zerlegung in einen steuerbaren Anteil (Zustandsvektor x) und einen nicht steuerbaren Anteil (Zustandsvektor e_x).

1.7 Zustandsregler mit Beobachter (LQG)

Gleichungen für gemeinsames System aus geschlossenem Regelkreis mit Beobachter

Während die Gleichungen auf der vorherigen Folie sehr gut für die Analyse des Systems geeignet sind, sollen hier die Gleichungen hergeleitet werden, die direkt auf die Zustände und geschätzten Zustände führen und daher besser für eine Simulation des geschlossenen Regelkreises mit Beobachter geeignet sind.

Fassen wir folgende Zustandsgleichungen zusammen:

$$\text{Regelkreis: } \dot{\underline{x}}(t) = \underline{A} \underline{x}(t) - \underline{b} \underline{k}^T \hat{\underline{x}}(t) + \underline{b} w(t) \qquad y(t) = \underline{c}^T \underline{x}(t)$$

$$\text{Beobachter: } \dot{\hat{\underline{x}}}(t) = \underline{l} \underline{c}^T \underline{x}(t) + \left(\underline{A} - \underline{b} \underline{k}^T - \underline{l} \underline{c}^T \right) \hat{\underline{x}}(t) + \underline{b} w(t) \qquad \hat{y}(t) = \underline{c}^T \hat{\underline{x}}(t)$$

Damit ergibt sich folgende Matrix/Vektor-Darstellung:

$$\begin{pmatrix} \dot{\underline{x}}(t) \\ \dot{\hat{\underline{x}}}(t) \end{pmatrix} = \begin{pmatrix} \underline{A} & -\underline{b} \underline{k}^T \\ \underline{l} \underline{c}^T & \underline{A} - \underline{b} \underline{k}^T - \underline{l} \underline{c}^T \end{pmatrix} \begin{pmatrix} \underline{x}(t) \\ \hat{\underline{x}}(t) \end{pmatrix} + \begin{pmatrix} \underline{b} \\ \underline{b} \end{pmatrix} w(t)$$

$$y(t) = \begin{pmatrix} \underline{c}^T & \underline{0} \end{pmatrix} \begin{pmatrix} \underline{x}(t) \\ \hat{\underline{x}}(t) \end{pmatrix}$$

1.7 Zustandsregler mit Beobachter (LQG)

Wichtige Fragen

- Wie beeinflussen die Eigenwerte des Beobachters die Eigenwerte des Regelkreises?
- Kann man den Zustandsregler mit Beobachter genauso entwerfen wie den Zustandsregler ohne Beobachter (also mit gemessenen Zuständen)?
- Wie sind die Eigenwerte für Regelkreis und Beobachter sinnvoll zu wählen?
- Wie beeinflusst der Beobachterfehler die Stabilität und die Dynamik des Regelkreises?

Den Schlüssel zur Antwort dieser Fragen liefert das

Separationstheorem

Die Eigenwerte des Regelkreises setzen sich aus den Eigenwerten der Matrix $\underline{A} - \underline{b}\underline{k}^T$ und den Eigenwerten der Matrix $\underline{A} - \underline{l}\underline{c}^T$ zusammen. Die erste Matrix beschreibt die *Eigenwerte* des Regelkreises mit *Zustandsregler ohne Beobachter*. Die zweite Matrix beschreibt die *Eigenwerte* des *Beobachters*.

1.7 Zustandsregler mit Beobachter (LQG)

Der Beweis für das Separationstheorem ergibt sich sofort aus der Block-Dreiecks-Struktur der Systemmatrix der Zustandsdifferentialgleichungen. Diese führt dazu, dass das charakteristische Polynom der Systemmatrix gleich dem Produkt der charakteristischen Polynome der beiden Matrizen auf der Hauptdiagonale ist:

$$\det \begin{pmatrix} s\underline{I} - \underline{A} + \underline{b}\underline{k}^T & -\underline{b}\underline{k}^T \\ \underline{0} & s\underline{I} - \underline{A} + \underline{l}\underline{c}^T \end{pmatrix} = \det \left(s\underline{I} - \underline{A} + \underline{b}\underline{k}^T \right) \cdot \det \left(s\underline{I} - \underline{A} + \underline{l}\underline{c}^T \right)$$

Konsequenzen des Separationstheorems

- Der Beobachter ändert nichts an den Eigenwerten des Regelkreises mit Zustandsregler ohne Beobachter. Insbesondere ist die Stabilität nicht berührt. Dies liegt *nicht* daran, dass eine gute Zustandsschätzung $\hat{\underline{x}} \approx \underline{x}$ vorausgesetzt würde. Vielmehr ist es in der Struktur der Gleichungen begründet und gilt selbst dann, wenn der Beobachterfehler groß ist!
- Dennoch hängt die Güte der Regelung natürlich von der Größe des Beobachterfehlers ab.
- Der Regelkreis hat $2n$ Eigenwerte, n davon werden durch die Zustandsrückführung \underline{k}^T bestimmt, die n anderen durch die Beobachterrückführung \underline{l} .
- Der Zustandsregler mit Beobachter kann genau so entworfen werden wie der ohne.

1.7 Zustandsregler mit Beobachter (LQG)

Wahl der Eigenwerte des Beobachters

- Die *Eigenwerte des Beobachters* sollten *deutlich schneller* sein als die dominierenden Eigenwerte des beobachtbaren, zustandsgeregelten Systems. D.h. die Eigenwerte von $\underline{A} - \underline{l} \underline{c}^T$ sollten in der komplexen Ebene deutlich links der dominierenden (mit größtem Realteil) Eigenwerte von $\underline{A} - \underline{b} \underline{k}^T$ liegen.
Mit einer solchen Wahl ist sichergestellt, dass der Beobachterfehler deutlich schneller abklingt als die Eigendynamik des geschlossenen Regelkreises. So wird auch gewährleistet, dass der Beobachter die Regelgüte nur wenig verschlechtert im Vergleich zu einem Zustandsregler mit gemessenen Zuständen.
- Als Faustregel macht man den Beobachter ca. 2 bis 6 Mal so schnell wie den geschlossenen Regelkreis.
- Die Beobachterdynamik darf nicht beliebig schnell gemacht werden! Zwar spielen Einschränkungen durch Stellgrößenbeschränkungen, Schonung des Stellglieds, Sparen von Energie, usw. keine Rolle, weil der Beobachter ein rein simuliertes System ist. Aber es muss ein *Kompromiss* zwischen *schneller Beobachterdynamik* (Wunsch: \underline{l} groß) einerseits und *Rauschempfindlichkeit* (Wunsch: \underline{l} klein) gefunden werden. Je stärker $y(t)$ verrauscht ist, desto vorsichtiger (also langsamer) sollte man den Beobachter einstellen.

1.7 Zustandsregler mit Beobachter (LQG)

Zustandsregler mit Beobachter im *Frequenzbereich*

Die *Führungsübertragungsfunktion* $w \rightarrow y$ eines Regelkreises mit Zustandsregler mit Beobachter ist gegeben durch:

$$G_w(s) = \underline{c}^T (s\underline{I} - \underline{A} + \underline{b}\underline{k}^T)^{-1} \underline{b}$$

Es spielt für die Führungsübertragungsfunktion keine Rolle, ob wir einen Beobachter verwenden, um die Zustände zu rekonstruieren, oder die Zustände direkt messen!

Mathematisch erklärt sich dieses erstaunliche Ergebnis damit, dass die n Zustände des Beobachterfehlers $\underline{e}_x(t)$ nicht steuerbar sind, d.h. sie sind von $w(t)$ nicht beeinflussbar:

$$\begin{pmatrix} \dot{\underline{x}}(t) \\ \dot{\underline{e}}_x(t) \end{pmatrix} = \begin{pmatrix} \underline{A} - \underline{b}\underline{k}^T & \underline{b}\underline{k}^T \\ \underline{0} & \underline{A} - \underline{l}\underline{c}^T \end{pmatrix} \begin{pmatrix} \underline{x}(t) \\ \underline{e}_x(t) \end{pmatrix} + \begin{pmatrix} \underline{b} \\ \underline{0} \end{pmatrix} w(t)$$

Obwohl der geschlossene Regelkreis $2n$ Pole hat, die sich aus den jeweils n Eigenwerten der Matrizen $\underline{A} - \underline{b}\underline{k}^T$ und $\underline{A} - \underline{l}\underline{c}^T$ zusammensetzen, tauchen in der Führungsübertragungsfunktion nur die Eigenwerte von $\underline{A} - \underline{b}\underline{k}^T$ auf.

D.h. die Eigenwerte des Beobachters, also der Matrix $\underline{A} - \underline{l}\underline{c}^T$, werden weggekürzt!

1.7 Zustandsregler mit Beobachter (LQG)

Warum taucht der Beobachter nicht in der Führungsübertragungsfunktion auf?

Eine weitere anschauliche Erklärung liefert folgende Überlegung: Bei der Berechnung von Übertragungsfunktionen kommt es nur auf den partikulären Teil der Lösung der DGL an, d.h. den Teil der durch das Eingangssignal (hier $w(t)$) verursacht wird. Der homogene Teil der Lösung, welcher das Abklingen der Anfangswerte beschreibt, wird nicht berücksichtigt. Man nimmt an, der Anfangswert sei schon abgeklungen. Dies bedeutet für die Beobachtergleichung, dass $\underline{e}_x(0) = 0$, also $\underline{x}_0 = \hat{\underline{x}}_0$, angenommen wird.

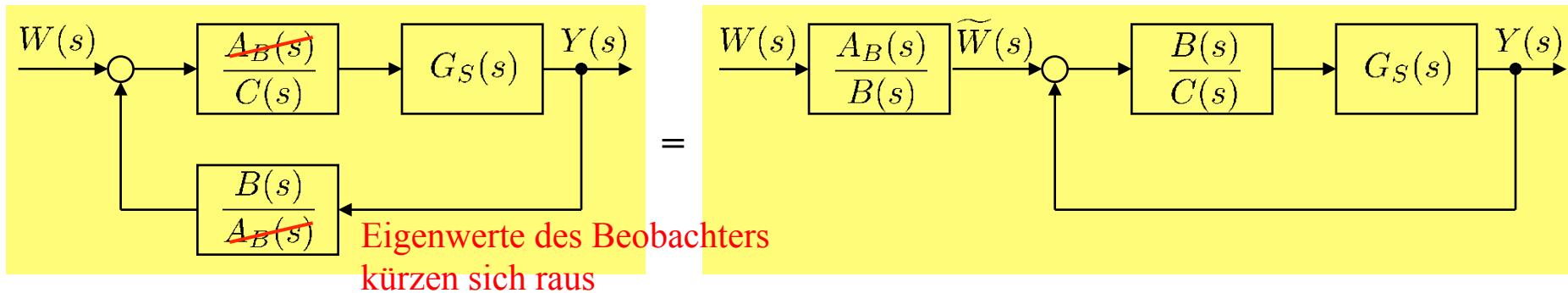
Das ist in der Praxis natürlich nicht der Fall. Die Dynamik mit der ein anfänglicher Beobachterfehler $\underline{e}_x(0) \neq 0$ abklingt, wird durch die Beobachterdynamik mitbestimmt. Daher beeinflusst die Wahl der Beobachterpole in der Realität selbstverständlich die Güte der Regelung.

Man muss sich auch klarmachen, dass in der Realität der Beobachterfehler nicht ausschließlich durch den Anfangswert bestimmt ist. Vielmehr werden durch *Störungen* und *Modellfehler* ständig neue Abweichungen zwischen dem wahren Zustandsvektor und dem beobachteten Zustandsvektor hervorgerufen. All dies ist in den Gleichungen nicht berücksichtigt.

1.7 Zustandsregler mit Beobachter (LQG)

Zustandsregler mit Beobachter als Standardregelkreis

Man kann zeigen, dass ein Regelkreis mit Zustandsregler mit Beobachter durch folgende Blockschaltbilder dargestellt werden kann:



D.h. er lässt sich als Standardregelkreis mit dynamischem Vorfilter (Vorsteuerung) darstellen. Ob wir mit den obigen Übertragungsfunktionen rechnen oder der Zustandsraum-darstellung ist nur für den Entwurf, die Analyse und die Implementierung interessant; aus Sicht der Regelstrecke ist das völlig egal. Die Polynome lassen sich recht einfach berechnen:

$$A_B(s) = \det(s\underline{I} - \underline{A} + \underline{l}\underline{c}^T) \quad (\text{char. Polynom des Beobachters})$$

$$C(s) = \det(s\underline{I} - \underline{A} + \underline{b}\underline{k}^T + \underline{l}\underline{c}^T) \quad (\text{char. Polynom des Beobachters im geregelten Kreis})$$

$$\frac{B(s)}{C(s)} = \underline{k}^T (s\underline{I} - \underline{A} + \underline{b}\underline{k}^T + \underline{l}\underline{c}^T)^{-1} \underline{l}$$

1.7 Zustandsregler mit Beobachter (LQG)

Verbindung zum Kalman-Filter

Ein **Kalman-Filter**, von Kalman 1960 entwickelt, ist auch ein Zustandsbeobachter. Bei der Herleitung des Kalman-Filters geht man im Vergleich zum Luenberger-Beobachter von einer anderen, nämlich stochastischen, Beschreibungen der Signale und der Verlustfunktion aus:

$$\begin{aligned}\dot{\underline{x}}(t) &= \underline{A} \underline{x}(t) + \underline{b} u(t) + \underline{\epsilon}(t) \\ y(t) &= \underline{c}^T \underline{x}(t) + \nu(t)\end{aligned}$$

Prozessrauschen mit Kovarianzmatrix $\underline{Q} = E\{\underline{\epsilon}(t) \underline{\epsilon}(t)^T\}$

Messrauschen mit Varianz $r = E\{\nu^2(t)\}$

Man geht also davon aus, der Zustandsvektor \underline{x} sei durch ein Prozessrauschen und der Ausgang y durch ein Messrauschen gestört. Die Stärke dieser Rauschsignale wird durch die Kovarianzmatrix \underline{Q} bzw. die Varianz r bestimmt. Des Weiteren werden Annahmen über die Natur dieser Zufallsprozesse gemacht. So nimmt man an, sie folgen einer *Gaußschen Wahrscheinlichkeitsverteilung* und sind statistisch unabhängig.

Es wird nun der Filter bzw. Beobachter (die Begriffe sind hier synonym zu verstehen) gesucht, der den Erwartungswert des quadratischen Fehlers zwischen dem realem Zustandsvektor \underline{x} und dem geschätzten (beobachtetem) Zustandsvektor $\hat{\underline{x}}$ möglichst klein macht:

1.7 Zustandsregler mit Beobachter (LQG)

$$E\{\underline{e}^T \underline{e}\} \rightarrow \min. \quad \text{mit } \underline{e} = \underline{x} - \hat{\underline{x}}$$

Als *stationäre* Lösung (also für $t \rightarrow \infty$) dieses Problems ergeben sich die *selben Gleichungen* wie für den *Luenberger-Beobachter* entworfen nach Matrix-Riccati! Die mathematischen Lösungen sind zwar gleich, aber sie können/müssen anders interpretiert werden:

Kalman-Filter

Messrauschen ist groß: Ausgangssignal y ist nicht zuverlässig, weil stark verrauscht.
→ Mache Rückkopplung schwach (\underline{l} klein).

Prozessrauschen ist groß: Geschätzte Zustände sind stark gestört, d.h. weichen stark von den realen Zuständen ab. → Mache Rückkopplung groß, um geschätzte Zustände schnell den realen anzugleichen (\underline{l} groß).

Optimaler Luenberger-Beobachter

Stellgrößenbestrafung r ist groß: Große Stellgrößen führen zu hohen Bestrafungen in der Verlustfunktion. → Mache Stellgrößen klein, d.h. mache \underline{l} klein.

Beobachterfehlerbestrafung Q ist groß: Die Abweichung zwischen realen und beobachtbaren Zuständen wird stark bestraft.
→ Mache Rückführungsvektor \underline{l} groß, damit die Dynamik des Beobachters schnell wird und sich daher der Beobachtungsfehler schnell der Null nähert.

1.7 Zustandsregler mit Beobachter (LQG)

Die Interpretation von Q und r sind völlig verschieden, aber die Auswirkungen sind die selben.

- Der Luenberger-Beobachter entworfen nach Matrix-Riccati erzielt einen Kompromiss zwischen den Bestrafungen Q und r .
- Der Kalman-Filter erzielt einen Kompromiss zwischen der Zuverlässigkeit des Messsignals y (je zuverlässiger, desto stärker die Rückführung \underline{d}) und der Genauigkeit des geschätzten Zustandsvektors \hat{x} (je genauer, d.h. näher an \underline{x} , desto schwächer die Rückführung \underline{d}).

Interpretation des Prozess- und Messrauschens

Die beiden Zufallsprozesse sollen nicht notwendigerweise wirkliches Rauschen nachbilden. Stattdessen will man mit ihnen die Zuverlässigkeit oder Genauigkeit *modellieren*, mit der ein Signal zu erhalten ist.

- Das Messrauschen modelliert das wirkliche Rauschen auf dem Signal y aber z.B. auch Fehler im Sensor.
- Das Prozessrauschen modelliert Ungenauigkeiten im Modell (\underline{A} , \underline{b} , \underline{c}).

1.7 Zustandsregler mit Beobachter (LQG)

Zeitvariantes Kalman-Filter

Das eigentlich Innovative am Kalman-Filter war, dass man nicht nur die stationär (für $t \rightarrow \infty$) optimale Lösung berechnen konnte (die äquivalent zum Luenberger-Beobachter ist), sondern auch die zu jedem Zeitpunkt optimale Lösung. Sie erfordert eine zeitvariante Rückführung: $\underline{l} = \underline{l}(t)$. Diese Lösung ist dual zu dem zeitvarianten Zustandsregler, der in Kapitel 1.5 im Zusammenhang mit der Riccati-Differentialgleichung besprochen wurde.

Wenn alle Randbedingungen über die Zeit gleich bleiben, schwingt $\underline{l}(t)$ sehr schnell auf seinen stationären Endwert $\underline{l}(t \rightarrow \infty)$ ein, der identisch mit der äquivalenten Luenberger-Beobachter-Rückführung ist. Besonderen Vorteil bietet der Kalman-Filter daher nur, wenn oft von einem unbekanntem Anfangswert gestartet werden muss, so dass dieser Einschwingvorgang ständig vorkommt oder wenn sich bestimmte Randbedingungen über die Zeit verändern. Dies könnten z.B. die Stärke des Rauschens sein oder es könnte sich um einen zeitvarianten Prozess handeln. Solch zeitvariantes Verhalten kann der Kalman-Filter berücksichtigen und zu jedem Zeitpunkt die optimale Rückführung berechnen.

Einen Zustandsregler mit Kalman-Filter als Beobachter nennt man oft LQG von "linear quadratic Gaussian", wobei sich letzteres auf die Verteilungsannahme der Zufallsprozesse bei der Herleitung des Kalman-Filters bezieht.

1.7 Zustandsregler mit Beobachter (LQG)

Reduzierter Beobachter

Ein Beobachter, so wie wir ihn bisher kennen, rekonstruiert n Zustände. Oft ist aber der Ausgang y gleich oder proportional zu einem Zustand, z.B. in der Regelungsnormalform, in der die Ausgangsgleichung lautet:

$$y(t) = (b_0 \ 0 \ \dots \ 0) \underline{x}(t) \quad \rightarrow \quad y(t) = b_0 x_1(t)$$

Außerdem kommt es vor, dass einige weitere Zustände gemessen werden und andere nicht. Mit dem Beobachter, den wir bisher kennen gelernt haben, werden aber immer *alle* Zustände rekonstruiert, auch diejenigen, die gemessen werden können. Das ist unnötig aufwändig und verschenkt Qualität, da vorhandene Informationen nicht genutzt werden.

Einen Beobachter, der diesen Nachteil vermeidet und *nur die nicht messbaren Zustände* rekonstruiert nennt man **reduzierten Beobachter**. Auf eine Herleitung und Angabe der Gleichungen des reduzierten Beobachters verzichten wir hier, siehe Kapitel 8.4 in [Lunze 2]. Die Grundidee der Herleitung ist, alle messbaren Zustände in dem Ausgangsvektor $\underline{y}(t)$ zu sammeln und diese Zustände durch die korrespondierenden gemessenen Elemente in $\underline{y}(t)$ zu ersetzen. Dann entsteht ein Beobachter mit $n-r$ Zuständen, wenn r die Anzahl der messbaren Zustände ist.

1.7 Zustandsregler mit Beobachter (LQG)

Robustheit des Zustandsreglers mit Beobachter

Der Zustandsregler mit Beobachter wurde in den 60er und 70er Jahren sehr erfolgreich eingesetzt, insbesondere in der Luft- und Raumfahrt. Gerade die leichte Erweiterbarkeit auf Mehrgrößensysteme und die natürliche Integration linearer Modelle spielten dabei eine wichtige Rolle.

Zunehmend gab es aber Misserfolge aus dem Bereich der verfahrenstechnischen Industrie. Dort sind genaue Modelle meist nicht verfügbar. Die Prozesse sind nichtlinear, teilweise mit verteilten Parametern und oft nicht genau verstanden und modellierbar. Man muss generell mit groben, stark vereinfachten Modellen leben. Hier erwies sich der Zustandsregler mit Beobachter als **nicht** sonderlich **robust** in Bezug auf die Modellfehler.

Im Jahre 1978 zeigt J. Doyle in seinem bahnbrechenden Aufsatz "**Guaranteed Margins for LQG Regulators**" mit dem Abstract "There are none!", dass *Zustandsregler mit Beobachter beliebig unrobust* sein können, d.h. im ungünstigsten Extremfall schon bei kleinsten Modellfehlern instabil werden können. Das steht im krassen Gegensatz zum Zustandsregler mit gemessenen Zuständen (LQ), der ja mindestens 60° Phasenreserve aufweist.

Diese Enttäuschung löste den Boom der *robusten Regelungstheorie* in den 80-90er Jahren aus.

1.8 Folgeregelung (*Tracking*)

Vermeidung einer bleibenden Regelabweichung bei Führungsgrößensprüngen

Bisher haben wir uns nur mit der Ausregelung von Anfangswerten \underline{x}_0 des Zustandsvektors beschäftigt. Es wurde angenommen, dass die Führungsgröße $w(t) = 0$ ist bzw. $w(t) = \text{konst.}$ und durch eine einfache Transformation auf Null gebracht werden kann.

Nun wollen wir das Problem der Festwertregelung (*regulation*) auf das Problem der Folgeregelung (*tracking*) erweitern. Eine Minimalforderung ist dabei das *Vermeiden* einer *bleibenden Regelabweichung* bei Sprüngen der Führungsgröße w . Wie wir aus Kapitel 10.2 von MRT I wissen, können wir dies auf 2 Arten gewährleisten:

1. Statischer Vorfilter.
2. I-Anteil im Regler oder Strecke.

Der 1. Weg vermeidet die bleibende Regelabweichung mit Hilfe einer *Steuerung*. Dies wirkt sich nicht auf die Dynamik, insbesondere auch nicht auf die Stabilität, des Regelkreises aus, ist aber empfindlich gegenüber Modellfehlern und Störungen.

Der 2. Weg ist robust gegenüber Modellfehlern und Störungen (d.h. selbst dann wird meist eine bleibende Regelabweichung vermieden), senkt aber die Phase des offenen Regelkreises um 90° ab, was destabilisierend wirkt.

1.8 Folgeregelung (*Tracking*)

Statisches Vorfilter

Ein statisches Vorfilter skaliert einfach die Führungsgröße, so dass bei einem Sprung von $w(t)$ keine bleibende Regelabweichung entsteht.

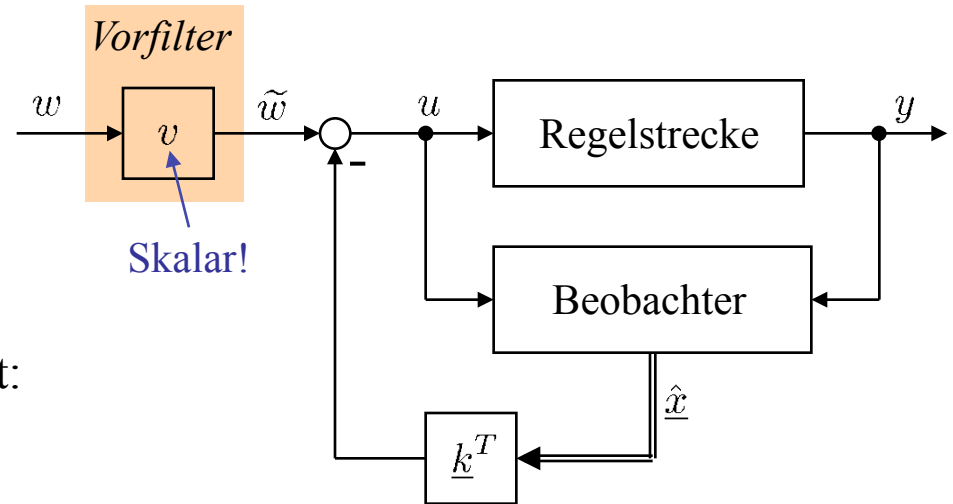
Die Übertragungsfunktion von w nach y ist:

$$G_w(s) = \underline{c}^T (s\underline{I} - A + \underline{b}\underline{k}^T)^{-1} \underline{b} v$$

Dies gilt *unabhängig* davon ob wir *mit oder ohne Beobachter* regeln. Zur Vermeidung bleibender Regelabweichungen fordern wir eine Verstärkung ($s \rightarrow 0$) der Führungsübertragungsfunktion von 1:

$$\underline{c}^T (-A + \underline{b}\underline{k}^T)^{-1} \underline{b} v = 1 \quad \rightarrow \quad v = \frac{1}{\underline{c}^T (-A + \underline{b}\underline{k}^T)^{-1} \underline{b}}$$

In allen Gleichungen müssen wir nun $w(t)$ durch $v w(t)$ ersetzen. Um das Führungsverhalten weiter zu verbessern, kann v durch eine Übertragungsfunktion $V(s)$ mit Verstärkung v ersetzt werden, die gewünschte dynamische Eigenschaften hat, s. Vorsteuerung/Vorfilter in MRT I.

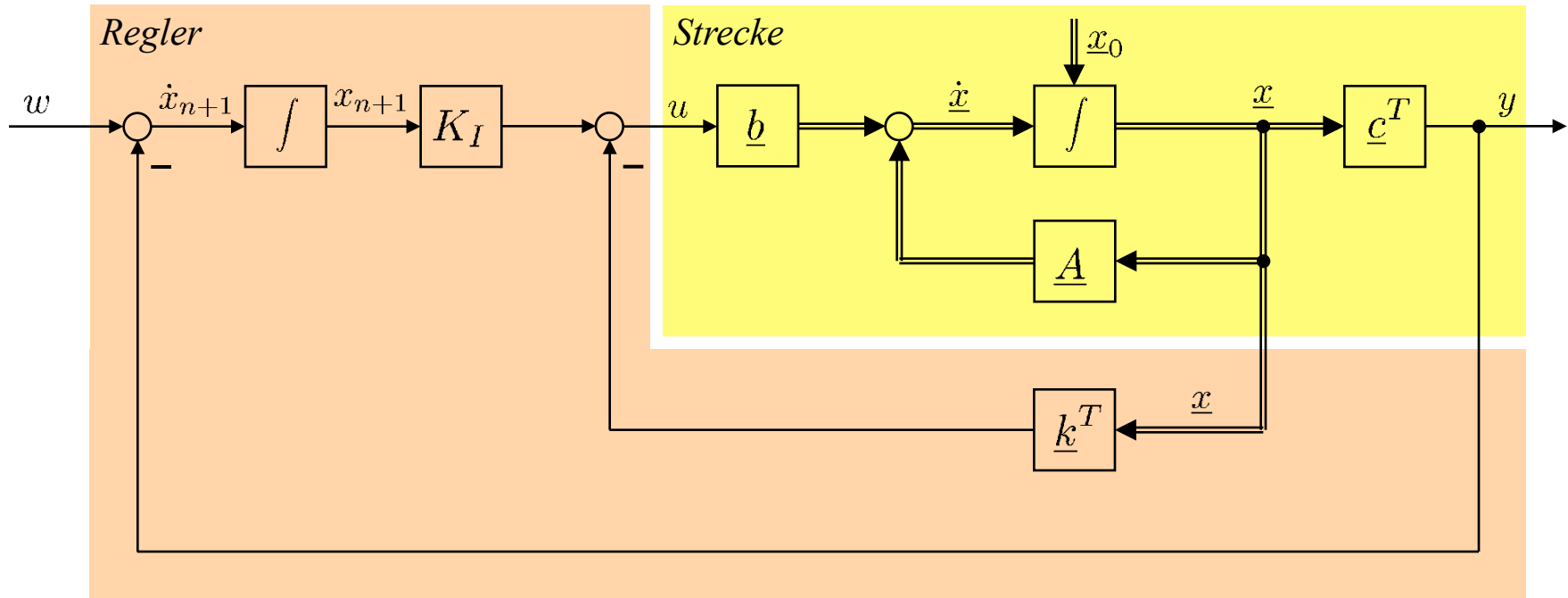


1.8 Folgeregelung (*Tracking*)

PI-Zustandsregelung (ohne Beobachter)

In folgendem Blockschaltbild wurde ad hoc ein Integrator eingebaut, der die Regelabweichung aufintegriert. So wird es möglich eine konstante Stellgröße $u(t)$ zu erzeugen, auch wenn die Regelabweichung gleich Null ist.

Der zusätzliche Integrator macht die Einführung eines zusätzlichen Zustandes x_{n+1} erforderlich. Der Parameter $K_I = 1/T_I$ ist der zugehörige Integrationsbeiwert.



1.8 Folgeregelung (*Tracking*)

Gleichungen für PI-Zustandsregler

Zunächst die Gleichungen ohne die Zustandsrückführung:

$$\begin{aligned}\underline{\dot{x}} &= \underline{A}\underline{x} + \underline{b}u & y &= \underline{c}^T \underline{x} \\ \dot{x}_{n+1} &= w - y = w - \underline{c}^T \underline{x}\end{aligned}$$

Zusammengefasst zu einem $n+1$ -dimensionalen Zustandsvektor:

$$\underbrace{\begin{pmatrix} \dot{x} \\ \dot{x}_{n+1} \end{pmatrix}}_{\underline{\dot{\tilde{x}}}} = \underbrace{\begin{pmatrix} \underline{A} & \underline{0} \\ -\underline{c}^T & 0 \end{pmatrix}}_{\underline{\tilde{A}}} \underbrace{\begin{pmatrix} \underline{x} \\ x_{n+1} \end{pmatrix}}_{\underline{\tilde{x}}} + \underbrace{\begin{pmatrix} \underline{b} \\ 0 \end{pmatrix}}_{\underline{\tilde{b}}} u + \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{\underline{\tilde{f}}} w$$

$$y = \underbrace{(\underline{c}^T \quad 0)}_{\underline{\tilde{c}}^T} \begin{pmatrix} \underline{x} \\ x_{n+1} \end{pmatrix}$$

Das führt auf folgendes Gleichungssystem:

$$\begin{aligned}\underline{\dot{\tilde{x}}} &= \underline{\tilde{A}}\underline{\tilde{x}} + \underline{\tilde{b}}u + \underline{\tilde{f}}w \\ y &= \underline{\tilde{c}}^T \underline{\tilde{x}}\end{aligned}$$

1.8 Folgeregelung (*Tracking*)

Die Stellgröße berechnet sich zu:

$$u = -\underbrace{(\underline{k}^T - K_I)}_{\tilde{\underline{k}}^T} \begin{pmatrix} \underline{x} \\ x_{n+1} \end{pmatrix}$$

Einsetzen dieser Reglergleichung führt auf:

$$\dot{\tilde{\underline{x}}} = \left(\tilde{\underline{A}} - \tilde{\underline{b}}\tilde{\underline{k}}^T \right) \tilde{\underline{x}} + \tilde{\underline{f}}w \quad y = \tilde{\underline{c}}^T \tilde{\underline{x}}$$

Eine Verwendung der "Schlange"-Größen ("~") statt der normalen in den Standardformeln für Polvorgabe oder LQ-Entwurf nach Matrix-Riccati liefert $n+1$ Reglerparameter, wovon der letzte der Stärke des I-Anteils entspricht: $k_{n+1} = -K_I$.

Eine Verwendung der "Schlange"-Größen liefert auch unmittelbar die Gleichungen für einen Zustandsregler mit Beobachter.

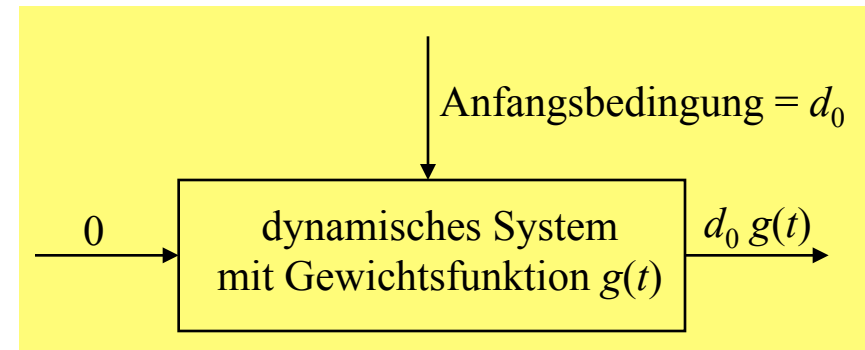
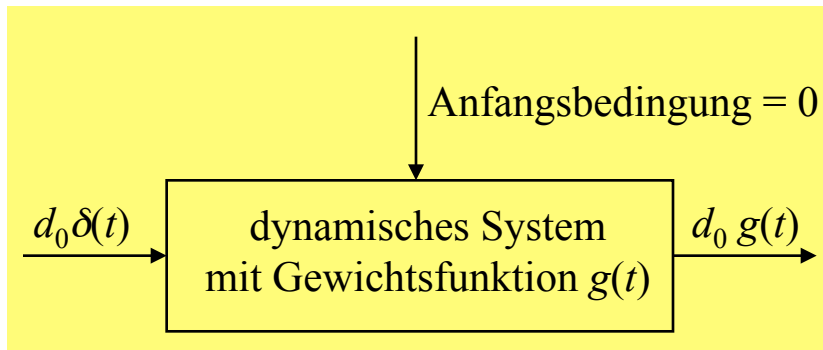
Eine solche Vorgehensweise hat neben vielen anderen Möglichkeiten, einen I-Anteil in die Zustandsregelung einzubringen, den Vorteil, dass sich die bekannten Entwurfsverfahren direkt nutzen lassen und keine Reglerparameter "von Hand" eingestellt werden müssen.

1.9 Führungs- und Störgrößenmodelle

Modellierung verschiedener Signaltypen

Die Gewichtsfunktion $g(t)$ eines linearen dynamischen Systems $G(s)$ charakterisiert alle seine Eigenschaften. Die Gewichtsfunktion gibt die homogene Lösung der DGL des Systems an, also wie es auf eine bestimmte *Anfangsbedingung* antwortet. Sie beschreibt ebenfalls die partikuläre Lösung auf einen *Delta-Impuls* $\delta(t)$ als *Eingangssignal*.

Für ein System 1. Ordnung können wir das im Blockschaltbild so darstellen:



Wir können also bestimmte *Signalklassen* (Impulse, Sprünge, Rampen, Sinus-Funktionen mit einer bestimmten Frequenz, etc.) mit Hilfe eines dynamischen Systeme charakterisieren, das die entsprechende Gewichtsfunktion besitzt.

1.9 Führungs- und Störgrößenmodelle

Äquivalenz zwischen Ausregelung einer Impuls-Führungsgröße und Anfangsbedingung

Allgemein gilt:

$$\underline{x}(t) = e^{At} \underline{x}_0 + \int_0^t e^{A(t-\tau)} \underline{b} u(\tau) d\tau \rightarrow \underline{x}(0) = \underline{x}_0 + \int_{0^-}^{0^+} e^{-A\tau} \underline{b} u(\tau) d\tau = \underline{x}_0 + \int_{0^-}^{0^+} \underline{I} \underline{b} \delta(\tau) d\tau$$

Für einen *Dirac-Impuls als Eingang* $u(t) = \delta(t)$ gilt daher: $\underline{x}(0) = \underline{x}_0 + \underline{b}$

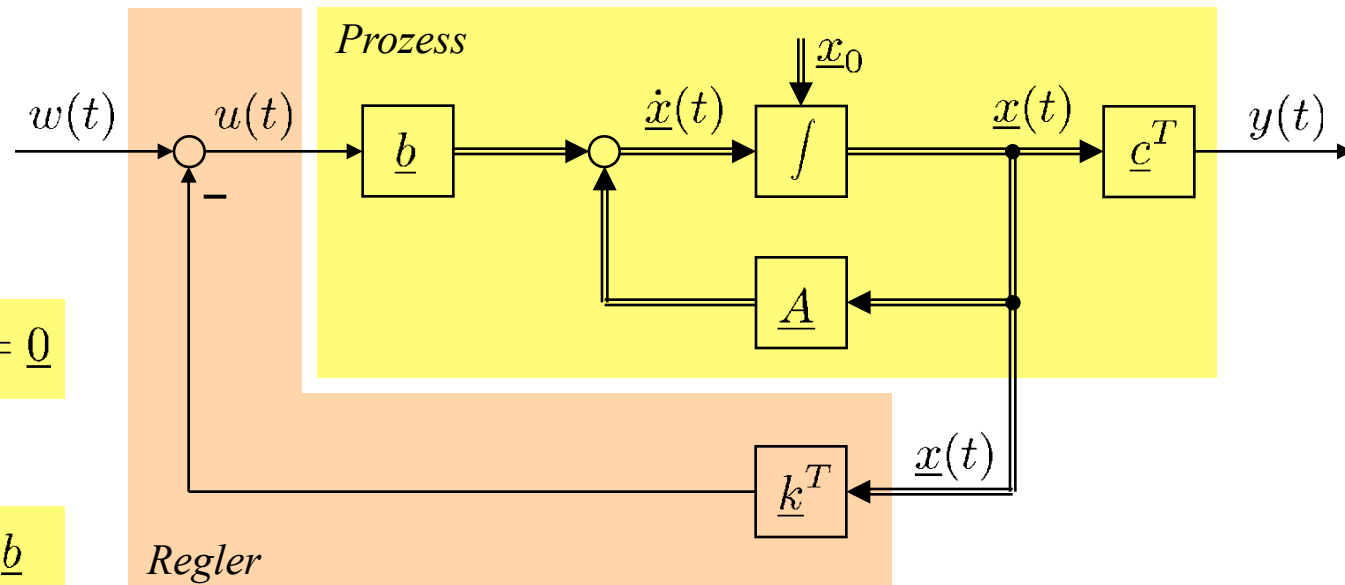
D.h. ein System reagiert auf einen Dirac-Impuls als Eingangssignal genauso wie auf eine Anfangsbedingung $\underline{x}_0 = \underline{b}$!

Gleiches gilt für einen Regelkreis. Die beiden folgenden Szenarien sind äquivalent:

$$w(t) = \delta(t) \text{ und } \underline{x}_0 = \underline{0}$$

oder

$$w(t) = 0 \text{ und } \underline{x}_0 = \underline{b}$$



1.9 Führungs- und Störgrößenmodelle

"Inneres Modell"-Prinzip

Von dem **"Inneren Modell"-Prinzip** wissen wir, dass die Regelgröße einer Führungsgröße asymptotisch (d.h. für $t \rightarrow \infty$) fehlerfrei folgt bzw. eine derartige Ausgangsstörung asymptotisch unterdrückt, wenn im offenen Regelkreis $G_0(s)$ eine Übertragungsfunktion enthalten ist, die mit der Signalklasse der Führungsgröße bzw. Ausgangsstörung korrespondiert.

Typische Signalklassen sind:	Zeitsignal	Inneres Modell in $G_0(s)$
Delta-Impuls	$\delta(t)$	1
Sprung	$\sigma(t)$	$1/s$
Rampe	$t \sigma(t)$	$1/s^2$
Sinus- oder Cosinus-Schwingung der Frequenz ω_0	$\sin(\omega_0 t) \sigma(t)$ $\cos(\omega_0 t) \sigma(t)$	$1/(s^2 + \omega_0^2)$ $1/(s^2 + \omega_0^2)$

Zeitliche Verschiebungen und verschiedene Amplituden spielen dabei keine Rolle. Es kommt nur auf die *Signalform* an (deshalb können auch sin- und cos-Signale gleich behandelt werden).

1.9 Führungs- und Störgrößenmodelle

Signalklassenmodell im Zustandsraum

Man kann die inneren Modelle von der Übertragungsfunktionsform in den Zustandsraum transformieren. Dabei verwendet man Modelle *ohne Eingang* u , deren Ausgang sich autonom aus den Anfangsbedingungen entwickelt. Diese Modelle sind daher auch *nicht steuerbar*. Normalerweise sind diese Modelle *grenzstabil* oder *instabil*, da sonst die Anfangsbedingungen exponentiell gegen Null abklingen würden (keine sehr typische Signalklasse).

Signalklasse Sprung:

Ein Integrator wird im Zustandsraum durch folgendes System 1. Ordnung beschrieben:

$$\dot{x}_1 = 0 \quad \text{mit } x_1(0) = d_0$$

$$y = x_1$$

Signalklasse Rampe:

Ein Doppelintegrator wird im Zustandsraum durch dieses System 1. Ordnung beschrieben:

$$\dot{x}_1 = x_2 \quad \text{mit } x_1(0) = d_0$$

$$\dot{x}_2 = 0 \quad \text{mit } x_2(0) = 0$$

$$y = x_1$$

1.9 Führungs- und Störgrößenmodelle

In Matrix/Vektor-Schreibweise ist das:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \text{mit} \quad \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} d_0 \\ 0 \end{pmatrix}$$

$$y = (1 \ 0) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

Signalklasse Sinus/Cosinus-Schwingung mit Frequenz ω_0 :

In Kapitel 1.2 wurde erklärt wie die reelle Block-Diagonalform für konjugiert komplexe Pole aussieht. Damit erhalten wir:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & \omega_0 \\ -\omega_0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \text{mit} \quad \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} d_0 \\ 0 \end{pmatrix} \quad \rightarrow \text{Cosinus}$$

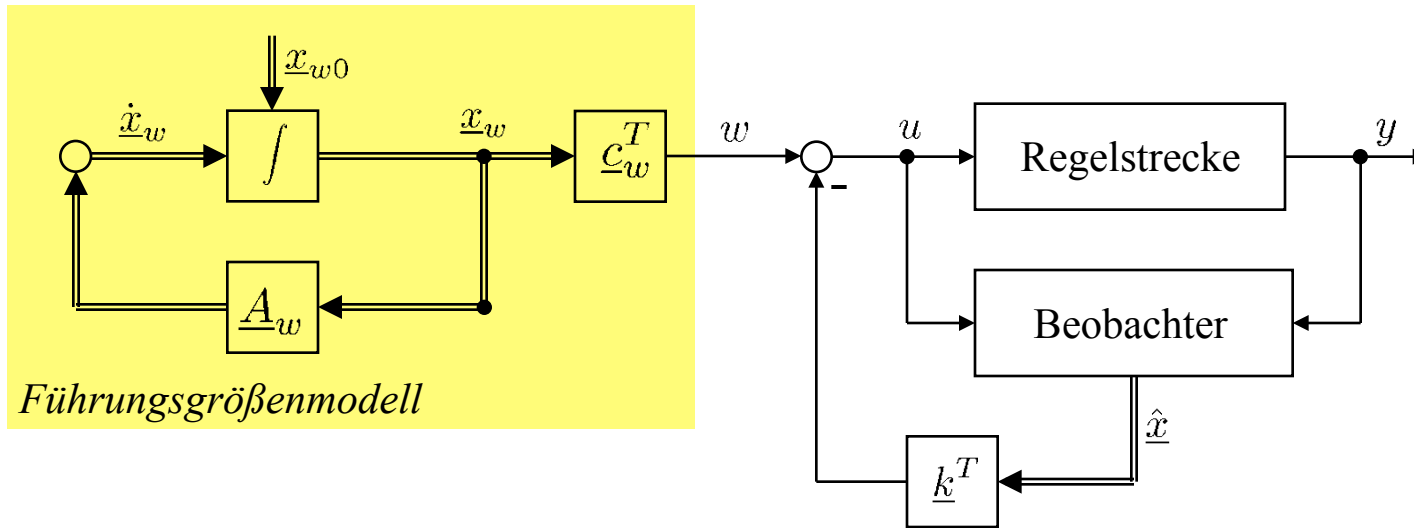
$$y = (1 \ 0) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \text{oder mit} \quad \begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 0 \\ d_0 \end{pmatrix} \quad \rightarrow \text{Sinus}$$

1.9 Führungs- und Störgrößenmodelle

Führungs- und Störgrößenmodell

Diese, eine bestimmte Signalklasse beschreibende, Zustandsraummodelle können in die "normalen" Zustandsraumgleichungen, welche die Regelstrecke (evtl. mit Zustandsregler und evtl. mit Beobachter) beschreiben, integriert werden.

Wenn wir also z.B. Führungsgrößen in Rampenform erwarten, dann setzen wir das entsprechende Zustandssystem zur "virtuellen" Generierung der Führungsgrößen ein:



Analog kann man beim Störgrößenmodell vorgehen.

1.9 Führungs- und Störgrößenmodelle

Durch die Integration eines Führungs- oder Störgrößenmodells erweitert sich der Zustandsvektor, z.B.:

$$\underline{x} \rightarrow \begin{pmatrix} \underline{x} \\ \underline{x}_w \end{pmatrix}$$

Entsprechendes gilt für einen evtl. Beobachter.

Auf diese Weise enthält dann der offene Regelkreis das innere Modell des entsprechenden Signals und kann es nach dem "Inneren Modell"-Prinzip ausregeln.

Diese Vorgehensweise wurde auch in Kapitel 1.8 beim Herleiten des PI-Zustandsreglers verfolgt, wo ein zusätzlicher Zustand den I-Anteil repräsentiert. Den gleichen Regelkreis mit I-Anteil hätte man auch erhalten, wenn man anstelle des manuellen Hinzufügens des Integrators, ein Führungs- oder Störgrößenmodell mit Signalklasse "Sprung" integriert hätte.

2. Digitale Regelung: Einführung

Inhalt Kapitel 2

2. Digitale Regelung

- 2.1 Zeitdiskrete Systeme
- 2.2 Differenzgleichungen
- 2.3 Digitaler Regelkreis
- 2.4 Digitaler PID-Regler
- 2.5 Faltungssumme
- 2.6 Abtasttheorem und Aliasing

2.1 Zeitdiskrete Systeme

Digitale Regelung

Mit dem Begriff "**Digitale Regelung**" bezeichnet man die Regelung zeitabgetasteter Systeme. Synonym werden auch die Begriffe **zeitdiskrete Regelung** oder **Abtastregelung** verwendet.

Hauptvorteile der digitalen gegenüber der analogen Regelung

- Realisierung in Software statt Hardware ist kostengünstiger und flexibler.
- Größerer Funktionsumfang kann realisiert werden (komplexere Regelalgorithmen, Identifikation, Adaption, Überwachung, Diagnose, Lernfähigkeiten, ...).

Aus diesen Gründen sind heute fast nur noch digitale Regler im Einsatz. Die Behandlung digitaler Regelsysteme unterscheidet sich von den analogen wie folgt:

- Differenzgleichungen statt Differentialgleichungen. Summen statt Integralen. Differenzen statt Ableitungen.
- Z-Transformation statt Laplace-Transformation.
- Andere Stabilitätskriterien. Einfachere Behandlung von Totzeiten. Einfachere Simulation.
- Ganz neue Möglichkeiten (Deadbeat-Regler) und Einschränkungen (Abtasttheorem).

2.1 Zeitdiskrete Systeme

Digitale oder analoge Regelung?

Beispiel: Stromregler für elektrische angesteuerte Ventile in einem elektronischen Motor- oder Getriebesteuergerät für Pkws

Analoge Realisierung

- Regler wird mit Operationsverstärkern, Widerständen und Kondensatoren als *Hardware* auf der Platine des Steuergeräts realisiert.
- Baueile und Platz auf der Platine kosten eine bestimmte Summe pro gebautem Steuergerät. → *Stückzahlabhängige Kosten*.
- Hohe Regelgüte, da kein Informationsverlust durch Abtastung.
- Für beliebig *schnelle Prozesse* geeignet (keine Rechenleistung notwendig).
- Nur für *einfache Reglerstrukturen* (wie PID) sinnvoll.
- Reglerstruktur und Reglerparameter können nach Abschluss des Hardware-Designs nicht mehr verändert werden. → *Frühzeitige Festlegung*, wenig zeitliche Flexibilität.
- *Kein Update* ohne Austausch (oder Änderung) der Hardware möglich.

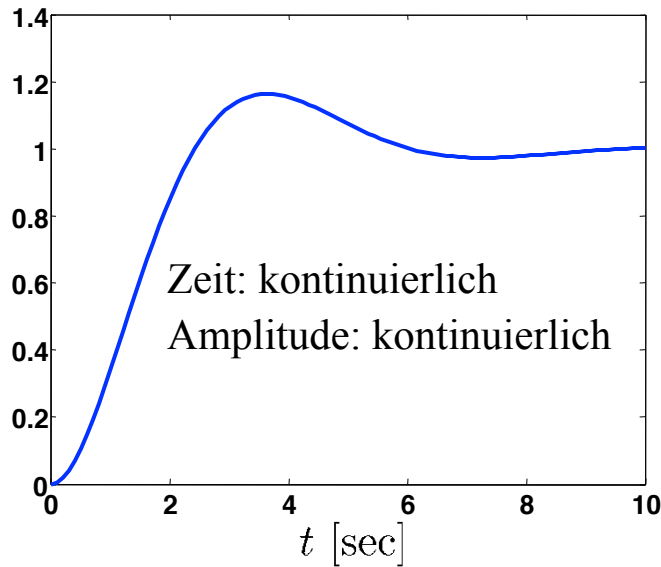
2.1 Zeitdiskrete Systeme

Digitale Realisierung

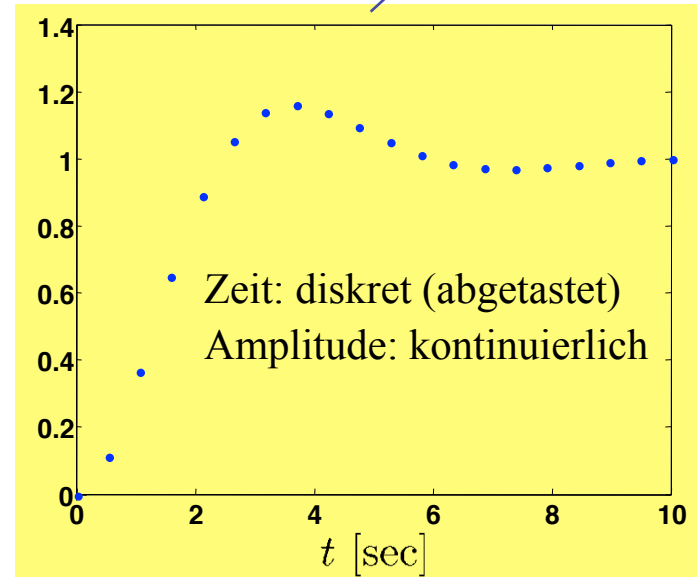
- Regler wird in *Software* realisiert; typischerweise in *C* programmiert.
- Einmalig fällt ein Entwicklungsaufwand an; Hardwareaufwand entsteht nicht (vorausgesetzt es ist noch genügend Rechenleistung des Mikrocontrollers frei). Entwicklungsaufwand kann evtl. auf mehrere Projekte umgelegt werden → *Fixkosten*.
- Benötigte *Rechenleistung steigt linear mit der Abtastfrequenz* an.
- Beliebige *komplexe Reglerstrukturen* können leicht realisiert werden. Auch beliebige Nichtlinearitäten lassen sich umsetzen.
- *Reglerstruktur* lässt sich durch *Umprogrammieren* ändern.
- *Reglerparameter* lassen sich durch *Kalibration*, d.h. auch nach Freigabe der Hard- und Software des Steuergerätes ganz kurzfristig (auch noch in der Fertigung) ändern.
→ Sehr große zeitliche Flexibilität (Änderung am Ende erhöhen allerdings das Risiko).
- *Updates* lassen sich *leicht durchführen*, sowohl während der Fertigung ab einer bestimmten Seriennummer als auch (im Notfall) als Rückruf. → Geringe Kosten.
- *Verschiedene Varianten* für verschiedene Baureihen lassen sich leicht erzeugen.

2.1 Zeitdiskrete Systeme

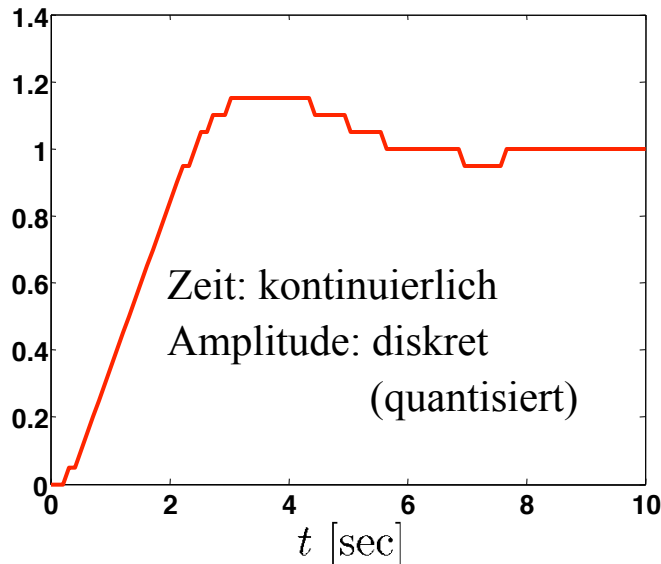
Digitale Regelung



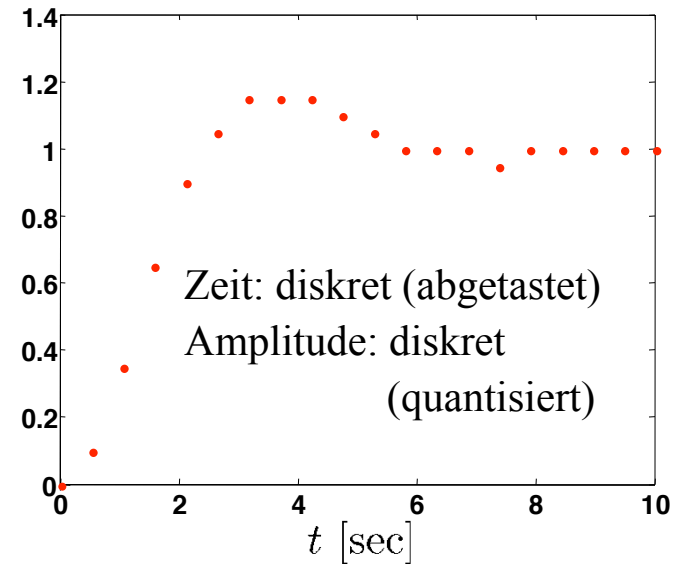
Abtastung



Quantisierung



Abtastung



Quantisierung

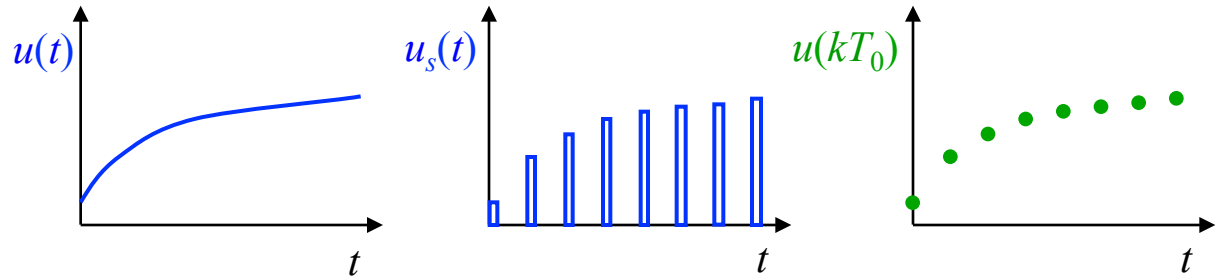
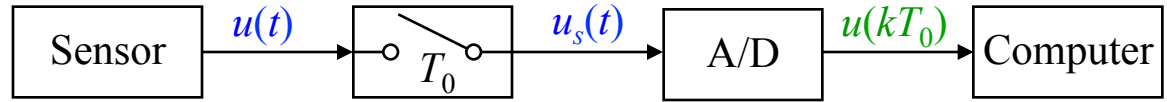
2.1 Zeitdiskrete Systeme

Abkürzung: $u(k) = u(kT_0)$
 $y(k) = y(kT_0)$

Analog/Digital- und Digital/Analog-Wandlung

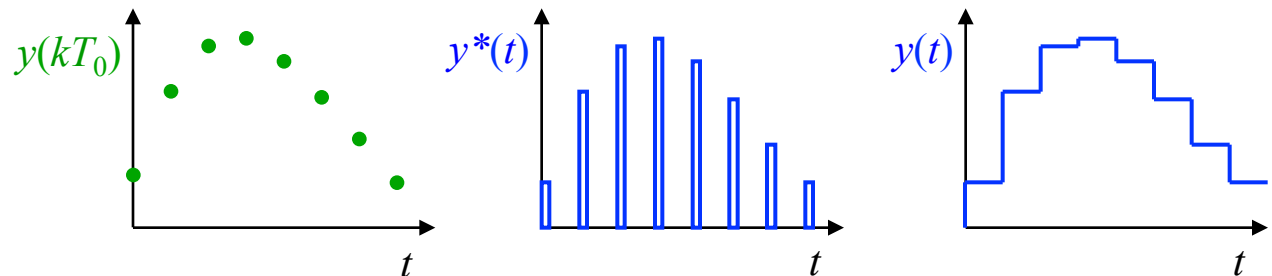
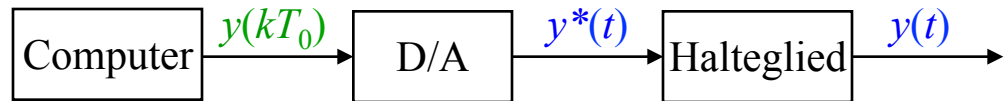
A/D-Wandlung

- Abtastzeit T_0 liegt zwischen μsec (Signalverarbeitung) und Stunden (thermisch, biologische Prozesse)
- Amplitudenauflösung von 8, 12 oder 16 Bit.



D/A-Wandlung

- Computer rechnet mit zeitdiskreten Folgen.
- Halteglied 0. Ordnung erzeugt stückweise konstante Signale.



2.2 Differenzgleichungen

Beziehung zwischen Differential- und Differenzgleichungen

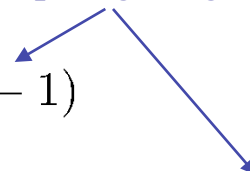
Der Computer verarbeitet **Signalfolgen**, die *zeitsynchron* mit der Abtastzeit T_0 abgetastet werden. Aus der Eingangsfolge $u(k)$ berechnet der Computer die Ausgangsfolge $y(k)$.

$$y(k) + a_1y(k-1) + \dots + a_ny(k-n) = b_0u(k) + b_1u(k-1) + \dots + b_mu(k-m)$$

Beispiele für die zeitkontinuierliche und zeitdiskrete Beschreibung dynamischer Systeme:

	zeitkontinuierlich	zeitdiskret
P	$y(t) = Ku(t)$	$y(k) = Ku(k)$
PT ₁	$y(t) + a_1\dot{y}(t) = b_0u(t)$	$y(k) + \tilde{a}_1y(k-1) = \tilde{b}_1u(k-1)$
PT ₂	$y(t) + a_1\dot{y}(t) + a_2\ddot{y}(t) = b_0u(t)$	$y(k) + \tilde{a}_1y(k-1) + \tilde{a}_2y(k-2) = \tilde{b}_1u(k-1)$
D	$y(t) = K_D\dot{u}(t)$	$y(k) = \tilde{b}_0(u(k) - u(k-1))$
I	$y(t) = K_I \int_0^t u(\tau)d\tau$	$y(k) = y(k-1) + \tilde{b}_1u(k-1)$

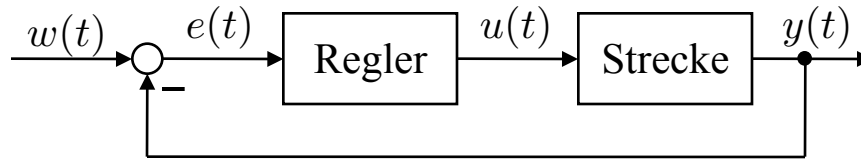
nicht
sprungfähig!



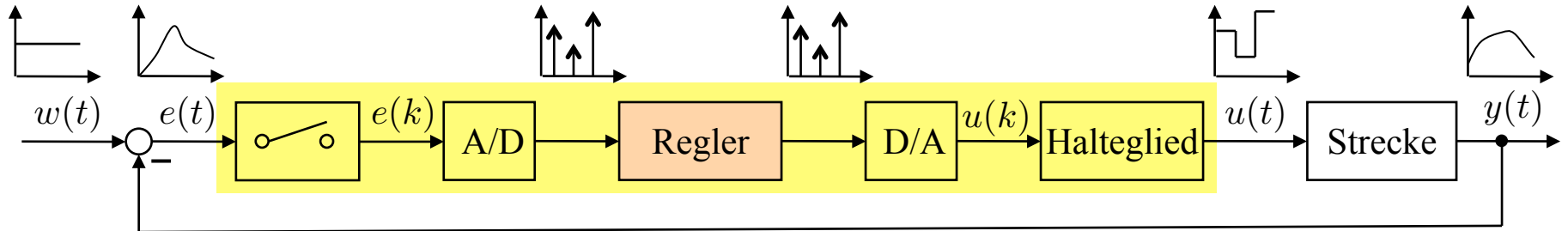
2.3 Digitaler Regelkreis

Aufbau eines digitalen Regelkreises

Analoger Regelkreis



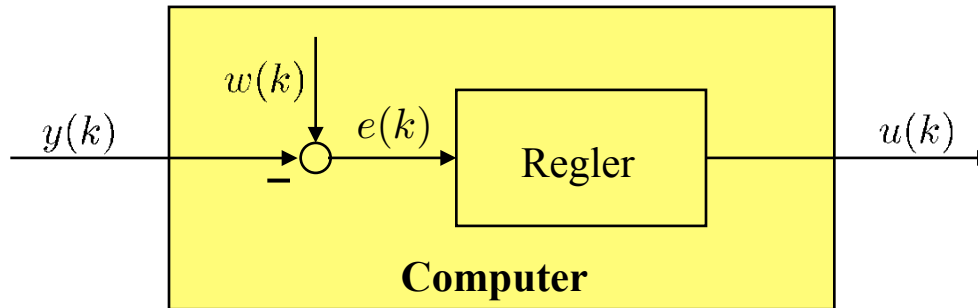
Digitaler Regelkreis



2.3 Digitaler Regelkreis

Aufbau eines digitalen Regelkreises

Alternativ kann die Führungsgröße auch im Computer direkt erzeugt werden und der Soll/Istwert-Vergleich wird im Digitalen durchgeführt:



Der Regler ist ein Computerprogramm, welches die Folge der Stellgrößen ausrechnet, z.B. für einen I-Regler:

$$u(k) = u(k - 1) + Ke(k)$$

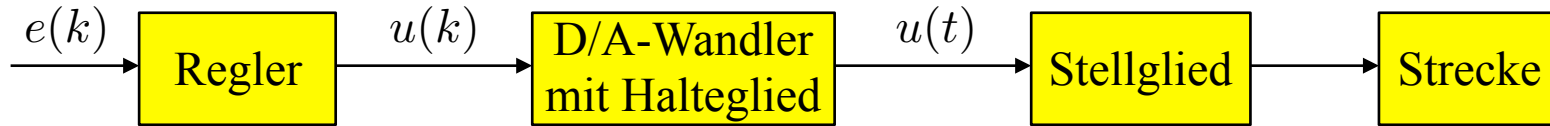
Bei integralwirkenden Stellgliedern berechnet man so die Änderung der Stellgröße:

$$\Delta u(k) = Ke(k)$$

mit $\Delta u(k) = u(k) - u(k - 1)$. Dies wird als *Geschwindigkeitsalgorithmus* bezeichnet.

2.3 Digitaler Regelkreis

Regler → Stellglied → Strecke



Regler erzeugt eine Signalfolge

Z.B. ein I-Regler mit Verstärkung 1 berechnet aus der Regelabweichung $e(k)$ und der alten Stellgröße $u(k-1)$ die neue Stellgröße $u(k)$:

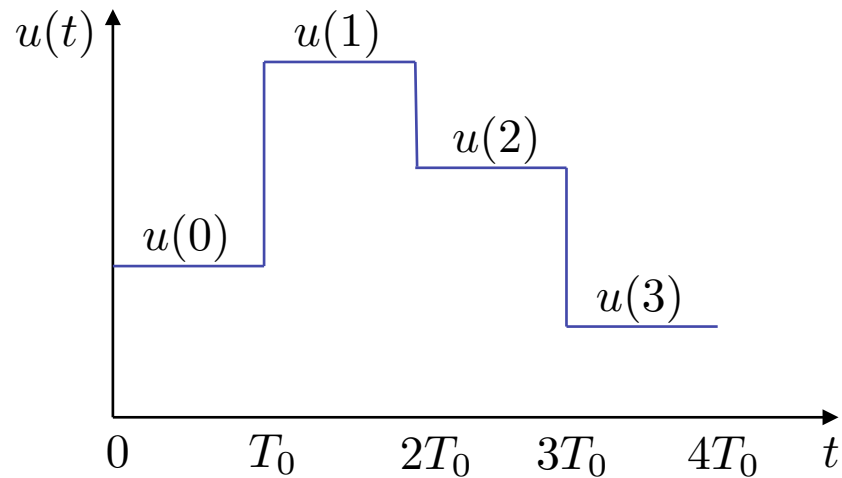
$$u(k) = u(k-1) + e(k)$$

Diese Signalfolge muss ins Analoge gewandelt werden. Üblicherweise verwendet man dafür ein Halteglied 0. Ordnung, d.h. die zeitdiskreten Werte werden für einen Abtastschritt konstant gehalten.

Aus der Folge:

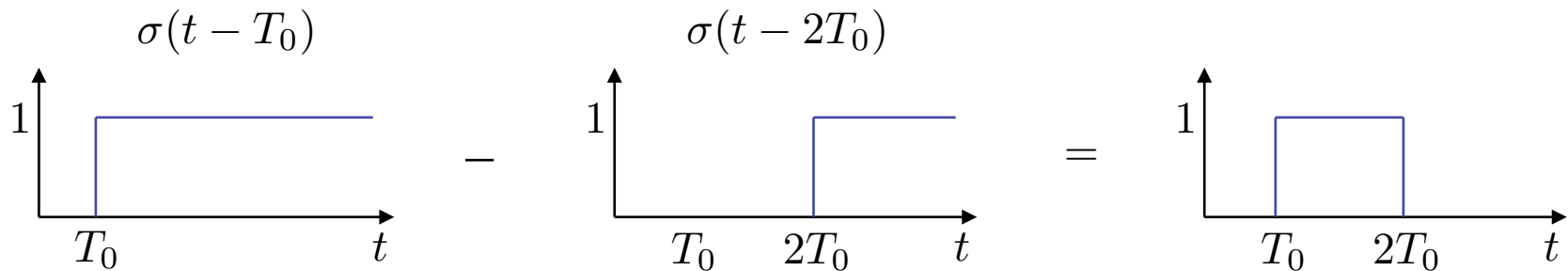
wird das kontinuierliche Zeitsignal: $u(0), u(1), u(2), u(3), \dots$

2.3 Digitaler Regelkreis



Dieses Signal lässt sich aus Rechtecken zusammensetzen: $u(t) =$

$$u(0)[\sigma(t) - \sigma(t - T_0)] + u(1)[\sigma(t - T_0) - \sigma(t - 2T_0)] + u(2)[\sigma(t - 2T_0) - \sigma(t - 3T_0)] + \dots$$



2.3 Digitaler Regelkreis

Regler erzeugt eine Signalfolge

Im Laplace-Bereich ist dies:

$$U(s) = u(0) \left[\frac{1}{s} - \frac{1}{s} e^{-T_0 s} \right] + u(1) \left[\frac{1}{s} e^{-T_0 s} - \frac{1}{s} e^{-2T_0 s} \right] + u(2) \left[\frac{1}{s} e^{-2T_0 s} - \frac{1}{s} e^{-3T_0 s} \right] + \dots$$
$$= \frac{1 - e^{-T_0 s}}{s} (u(0) + u(1)e^{-T_0 s} + u(2)e^{-2T_0 s} + \dots)$$

Halteglied 0. Ordnung

$$= \frac{1 - e^{-T_0 s}}{s} \sum_{k=0}^{\infty} u(k) e^{-kT_0 s}$$

Abtastwerte Totzeiten kT_0

2.3 Digitaler Regelkreis

Beispiel: P-Regler mit PT₁-Glied

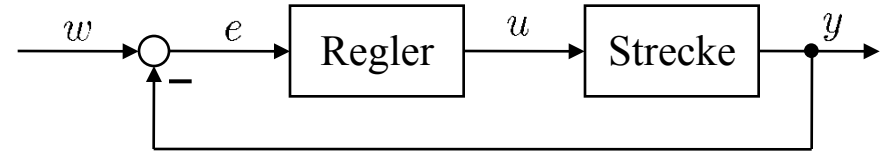
Zeitkontinuierlicher Fall:

P-Regler: $u(t) = K_R e(t)$

PT₁-Strecke: $T\dot{y}(t) + y(t) = u(t)$

Geschlossener Regelkreis: $T\dot{y}(t) + y(t) = K_R[w(t) - y(t)]$

$$\rightarrow T\dot{y}(t) + (K_R + 1)y(t) = K_R w(t) \quad \rightarrow \quad \boxed{\frac{T}{K_R + 1}\dot{y}(t) + y(t) = \frac{K_R}{K_R + 1}w(t)}$$



Der geschlossene Regelkreis hat also auch PT₁-Verhalten. Und zwar mit einer Zeitkonstanten von $T/(K_R+1)$, die umso kleiner wird, je größer die Reglerverstärkung ist und einer Verstärkung von $K_R/(K_R+1)$, die umso weniger von 1 abweicht (also eine umso kleinere bleibende Regelabweichung aufweist), je größer K_R ist.

Der geschlossene Regelkreis ist *strukturstabil*, d.h. K_R kann gegen unendlich gehen und die Stabilität ist (theoretisch) nicht gefährdet. (Mit Hilfe der Laplace-Transformation hätten wir das schneller herleiten können. Aber da wir die z-Transformation als deren zeitdiskretes Äquivalent noch nicht kennen, beschränken wir uns hier auf den Zeitbereich.)

2.3 Digitaler Regelkreis

Beispiel: P-Regler mit PT₁-Glied

Zeitdiskreter Fall:

$$\text{P-Regler: } u(k) = K_R e(k)$$

$$\text{PT}_1\text{-Strecke: } y(k) + a_1 y(k-1) = b_1 u(k-1)$$

$$\text{Geschlossener Regelkreis: } y(k) + a_1 y(k-1) = b_1 K_R [w(k-1) - y(k-1)]$$

$$\rightarrow \boxed{y(k) = -(a_1 + b_1 K_R) y(k-1) + b_1 K_R w(k-1)}$$

$= \tilde{a}_1$. Muss betragsmäßig < 1 sein, damit Stabilität gesichert ist!

$$\text{Für } w = 0 \text{ gilt: } y(k) = \tilde{a}_1 y(k-1)$$

$$y(1) = \tilde{a}_1 y(0)$$

$$y(2) = \tilde{a}_1 y(1) = \tilde{a}_1^2 y(0)$$

\vdots

$$y(k) = \tilde{a}_1^k y(0)$$

Dieser Ausdruck strebt nur gegen 0, wenn $|\tilde{a}_1| < 1$ ist.

$$a_1 = -e^{-\frac{T_0}{T}}$$

$$b_1 = 1 - e^{-\frac{T_0}{T}}$$

Der geschlossene Regelkreis ist stabil für:

$$-\frac{1 + a_1}{b_1} < K_R < \frac{1 - a_1}{b_1}$$

Im Gegensatz zum analogen Regelkreis, ist der digitale Regelkreis *nicht strukturstabil!*

2.4 Digitaler PID-Regler

$$\text{PID: } u(t) = K_P \left(e(t) + \frac{1}{T_I} \int_0^t e(\tau) d\tau + T_D \frac{d}{dt} e(t) \right)$$

Zeitdiskretisierung des PID-Reglers

Es gibt viele Möglichkeiten, ein zeitkontinuierliches System in ein zeitdiskretes System umzuwandeln. Am einfachsten ist die Approximation mittels Differenzenquotienten. Eine solche Approximation hat aber den schwerwiegenden Nachteil, dass sie nur für sehr kleine Abtastzeiten eine gute Näherung darstellt:

Approximation mittels Differenzenquotienten (nur für sehr kleine T_0)

$$P\text{-Anteil: } u(k) = K_P e(k)$$

$$D\text{-Anteil: } u(k) = T_D \frac{e(k) - e(k-1)}{T_0}$$

$$I\text{-Anteil: } u(k) = u(k-1) + \frac{T_0}{T_I} e(k-1)$$

$$u(k) = \frac{T_0}{T_I} \sum_{i=1}^k e(i-1)$$

$$u(k-1) = \frac{T_0}{T_I} \sum_{i=1}^{k-1} e(i-1)$$

$$\text{PID-Regler: } u(k) = u(k-1) + K_P \left(1 + \frac{T_D}{T_0} \right) e(k) + K_P \left(\frac{T_0}{T_I} - \frac{T_D}{T_0} \right) e(k-1)$$

2.4 Digitaler PID-Regler

Übliche Variante des digitalen PID-Reglers

Die obigen Näherungen werden sehr schlecht, wenn die Abtastzeit nicht sehr klein gegenüber den relevanten Prozesszeitkonstanten gewählt wird. Darüber hinaus ist nicht gewährleistet, dass stabile Prozesse nach dieser Approximation auch im Zeitdiskreten stabil sind. Eine wesentlich bessere Variante nutzt die Trapezregel zur Integration:

Trapezregel für den I-Anteil:

$$u(k) = \frac{T_0}{T_I} \left(\frac{e(0)}{2} + \sum_{i=1}^{k-1} e(i) + \frac{e(k)}{2} \right)$$

Dieses Vorgehen ist gleichbedeutend mit der Anwendung der **bilinearen Transformation** oder **Tustin-Transformation** von der Laplace-Transformation zur z-Transformation:

$$s = \frac{2}{T_0} \frac{z-1}{z+1} = \frac{2}{T_0} \frac{1-z^{-1}}{1+z^{-1}}$$

2.4 Digitaler PID-Regler

Digitaler PID-Regler (mit Trapezregel):

$$u(k) = u(k-1) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2)$$

$$q_0 = K_P \left(1 + \frac{T_0}{2T_I} + \frac{T_D}{T_0} \right) \quad q_1 = K_P \left(-1 + \frac{T_0}{2T_I} - 2\frac{T_D}{T_0} \right) \quad q_2 = K_P \frac{T_D}{T_0} e(k-2)$$

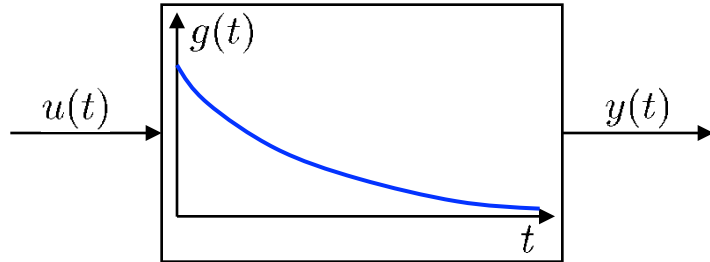
Die 3 Parameter des digitalen PID-Reglers q_0 , q_1 und q_2 können, wie im obigen Kasten angegeben, zu den Parametern des analogen PID-Reglers in Beziehung gesetzt werden. Das ist wichtig, wenn man einen analogen in einen digitalen Regler umwandeln möchte oder die Interpretation in Form eines P-, I- und D-Anteils benötigt. Generell wird dieses Vorgehen nur empfohlen, wenn die Abtastzeit um den Faktor 10 kleiner ist als die dominierende Prozesszeitkonstante.

Ansonsten können die 3 Parameter q_0 , q_1 und q_2 auch als unabhängige Werte gesehen werden und direkt durch beliebige Reglerentwurfsverfahren bestimmt werden.

2.5 Faltungssumme

Faltungsintegral und Faltungssumme

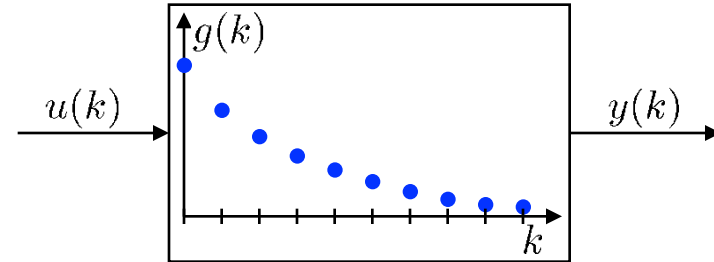
zeitkontinuierlich



$$\begin{aligned}y(t) &= \int_0^t g(t - \tau)u(\tau)d\tau \\ &= \int_0^t g(\tau)u(t - \tau)d\tau\end{aligned}$$

- Die Gewichtsfunktion $g(t)$ enthält alle Informationen über das lineare System.
- Leichter rechnet es sich im Frequenzbereich mit der Laplace-Transformierten $G(s)$.

zeitdiskret



$$\begin{aligned}y(k) &= \sum_{i=0}^k g(k - i)u(i) \\ &= \sum_{i=0}^k g(i)u(k - i)\end{aligned}$$

- Die Gewichtsfunktion $g(k)$ enthält alle Informationen über das lineare System.
- Leichter rechnet es sich im Frequenzbereich mit der z-Transformierten $G(z)$.

2.5 Faltungssumme

$$y(k) = \sum_{i=0}^k g(k-i)u(i) = \sum_{i=0}^k g(i)u(k-i)$$

Eigenschaften der Faltungssumme

- Ausgeschrieben liest sich die Faltungssumme wie folgt:

$$y(k) = g(k)u(0) + g(k-1)u(1) + \dots + g(1)u(k-1) + g(0)u(k)$$

- Wie im Zeitkontinuierlichen ist die Gewichtsfunktion identisch mit der Systemantwort auf einen (zeitdiskreten) Dirac-Impuls am Eingang $u(k) = \delta(k)$:

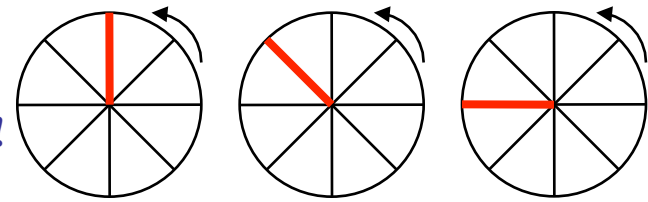
$$u(0) = 1, u(1) = 0, u(2) = 0, \dots \rightarrow y(k) = g(k).$$

- Nur für sprungfähige Systeme existiert $g(0)$. Im Normalfall gilt also $g(0) = 0$.
- Hat die Gewichtsfunktion die Länge $l+1$, ist also endlich, d.h. alle $g(i) = 0$ für $i > l$, dann hat das System eine **endliche Impulsantwort (finite impulse response, FIR)**. Man spricht dann auch davon, dass $y(k)$ ein **gleitender Mittelwert (moving average, MA)** ist.
- Ist die Gewichtsfunktion unendlich lang, dann hat das System eine **unendliche Impulsantwort (infinite impulse response, IIR)**. Ähnlich wie im Zeitkontinuierlichen kann dann der Ausgang aber über eine endliche Gleichung aus vergangenen (im Zeitkontinuierlichen: abgeleiteten) Ausgangswerten berechnet werden:

$$y(k) = -a_1y(k-1) - \dots - a_ny(k-n) + b_0u(k) + b_1u(k-1) + \dots + b_mu(k-m)$$

2.6 Abtasttheorem und Aliasing

Rad steht
scheinbar!

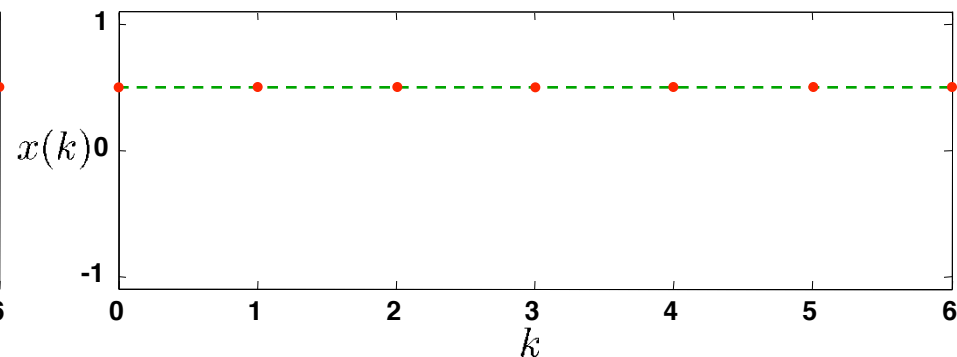
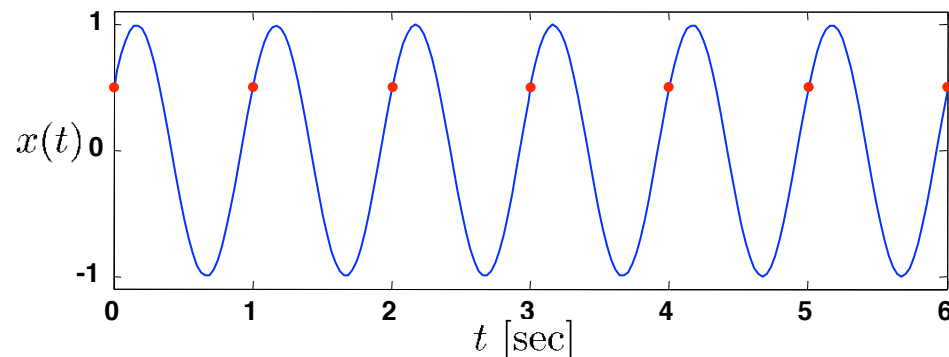


Abtastung zeitkontinuierlicher Signal

Jeder hat schon einmal drehende Speichenräder einer losfahrenden Kutsche im Film gesehen. Zuerst sieht man das Rad beschleunigen. Ab einer gewissen Drehzahl scheint das Rad plötzlich seine Drehrichtung zu ändern und schnell rückwärts zu drehen (obwohl die Kutsche weiter beschleunigt). Dann scheint das Rad immer langsamer zu werden bis es sogar stehen bleibt. Das ist ein offensichtlicher Widerspruch zur immer schneller fahrenden Kutsche.

Diese komischen Effekte werden durch das sog. **Aliasing** verursacht und treten bei allen abgetasteten Systemen auf. Offensichtlich gibt es Probleme, wenn wir Signale abtasten, deren Frequenz in den Bereich der Abtastfrequenz kommt. Der *Film* spielt dabei die Rolle des *Abtasters* mit einer Bildwiederholrate bzw. Abtastfrequenz von $f_0 = 25$ Hz.

Was passiert, wenn wir ein Signal der Frequenz $f = 1$ Hz mit $f_0 = 1$ Hz abtasten?

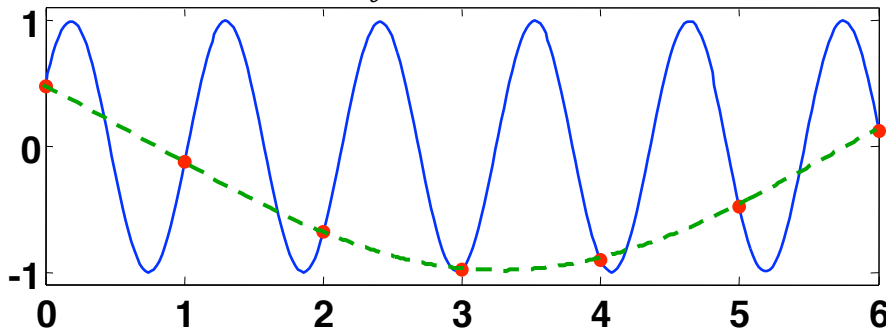


2.6 Abtasttheorem und Aliasing

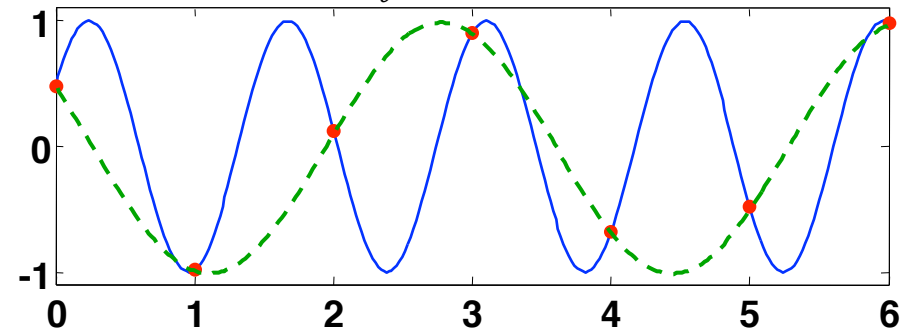
Aliasing

Offensichtlich geht die Schwingung komplett verloren. Wir erhalten ein Signal der Frequenz null (einen Gleichwert). Diese Tatsache ist unabhängig von der Phasenlage des Abtasters (nur die Höhe des Gleichwertes hängt davon ab). Zur Illustration noch ein paar weitere Beispiele mit $f = 0.9$ Hz, 0.7 Hz, 0.5 Hz und 0.3 Hz und jeweils $f_0 = 1$ Hz.

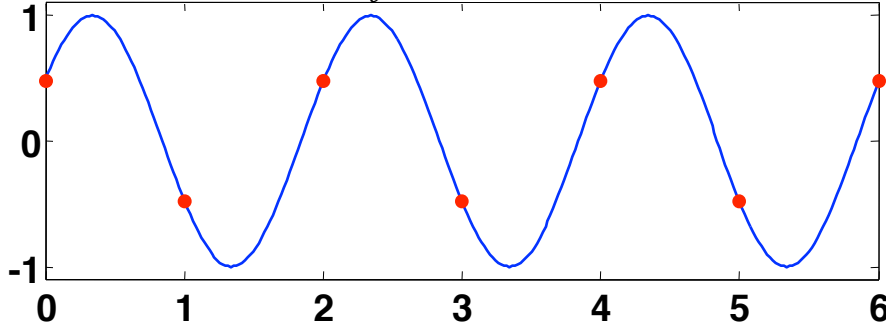
$f = 0.9$ Hz



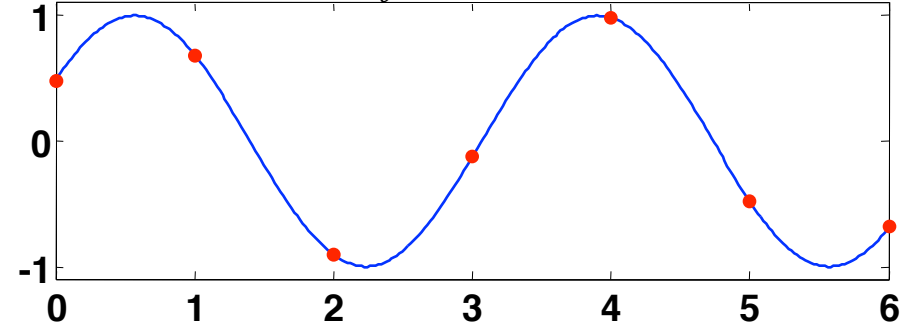
$f = 0.7$ Hz



$f = 0.5$ Hz



$f = 0.3$ Hz



2.6 Abtasttheorem und Aliasing

Abtasttheorem

Aus den Beispielen der vorangegangenen Seite erkennen wir empirisch, dass wir mindestens mit der doppelten Signalfrequenz abtasten müssen ($f = 0.5 \text{ Hz}$, $f_0 = 1 \text{ Hz}$), um das Signal nach der Abtastung noch richtig wiedergeben zu können. Reale Signale bestehen aus einem Gemisch vieler Frequenzen. Dann bezieht sich die Forderung nach der doppelten Abtastfrequenz auf den Signalanteil mit der höchsten Frequenz f_{\max} .

Shannonsches Abtasttheorem

Das Signal $x(t)$ soll abgetastet werden. Die höchste Frequenzkomponente von $x(t)$ sei f_{\max} . Dann muss die Abtastfrequenz mindestens doppelt so groß gewählt werden, wie die höchste Frequenzkomponente von $x(t)$:

$$f_0 > 2f_{\max}$$

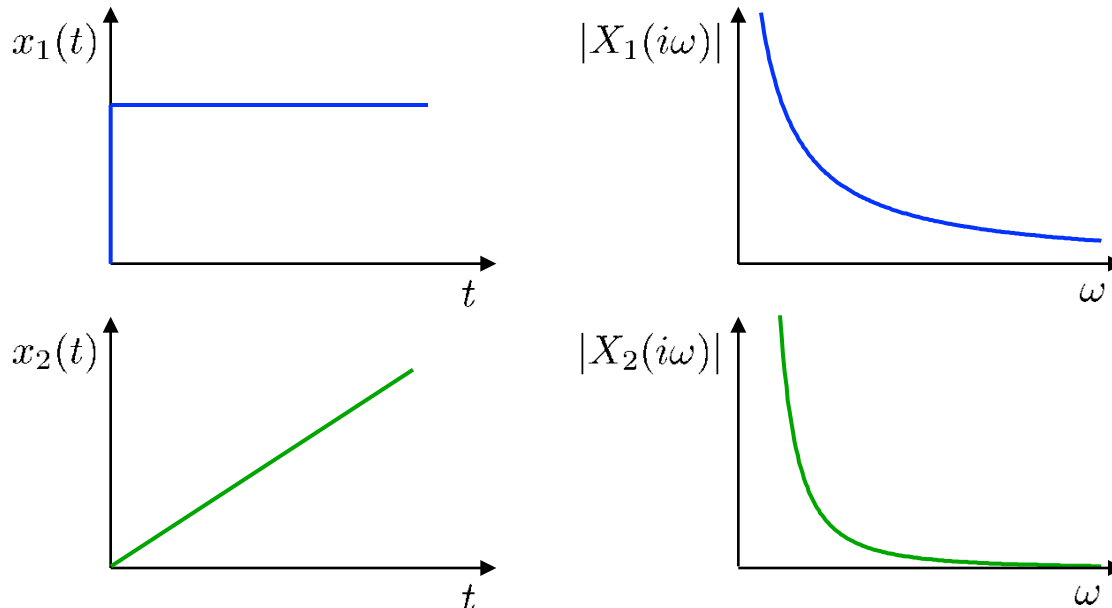
Ist dies nicht gewährleistet, kommt es zu Aliasing, d.h. die Frequenzkomponenten mit $f > \frac{1}{2}f_0$ werden in den Bereich niedrigerer Frequenzen gespiegelt. Auf diese Weise können hochfrequente Störsignale großen Schaden anrichten und niederfrequente Nutzsignale überlagern.

In der Praxis wählt man ca. $f_0 = 5 \dots 10 f_{\max}$

2.6 Abtasttheorem und Aliasing

Veranschaulichung des Abtasttheorems und des Aliasing-Effekts

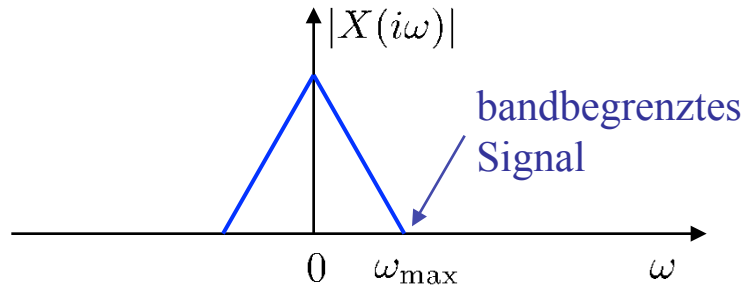
Wenn das Abtasttheorem eingehalten wird, kann das Originalsignal aus dem abgetasteten Signal rekonstruiert werden, d.h. es findet durch die Abtastung kein Informationsverlust statt. In Realität sind aber die meisten Signale nicht bandbegrenzt, haben also gar keine Komponente maximaler Frequenz f_{\max} bzw. $f_{\max} = \infty$. Vielmehr sind in vielen typischen Signalen (Sprünge, Rampen, Rechtecke, ...) alle Frequenzen von 0 bis unendlich enthalten. Solche Signale lassen sich daher aus dem abgetasteten Signal nicht perfekt rekonstruieren.



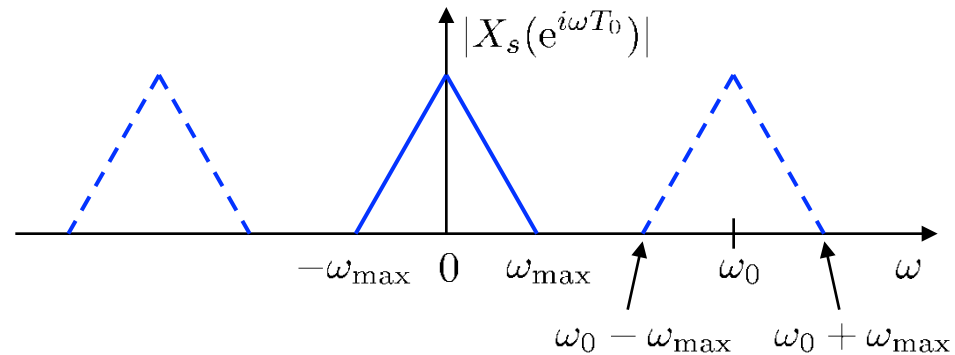
2.6 Abtasttheorem und Aliasing

Veranschaulichung des Abtasttheorems und des Aliasing-Effekts

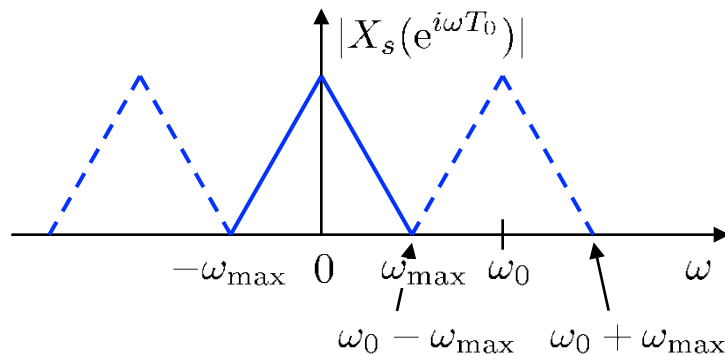
Spektrum des kontinuierlichen Signals



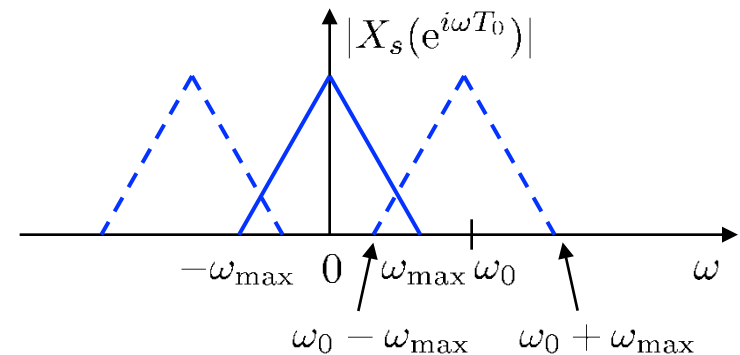
Spektrum des abgetasteten Signals $\omega_0 > 2\omega_{\max}$



Spektrum des abgetasteten Signals $\omega_0 = 2\omega_{\max}$

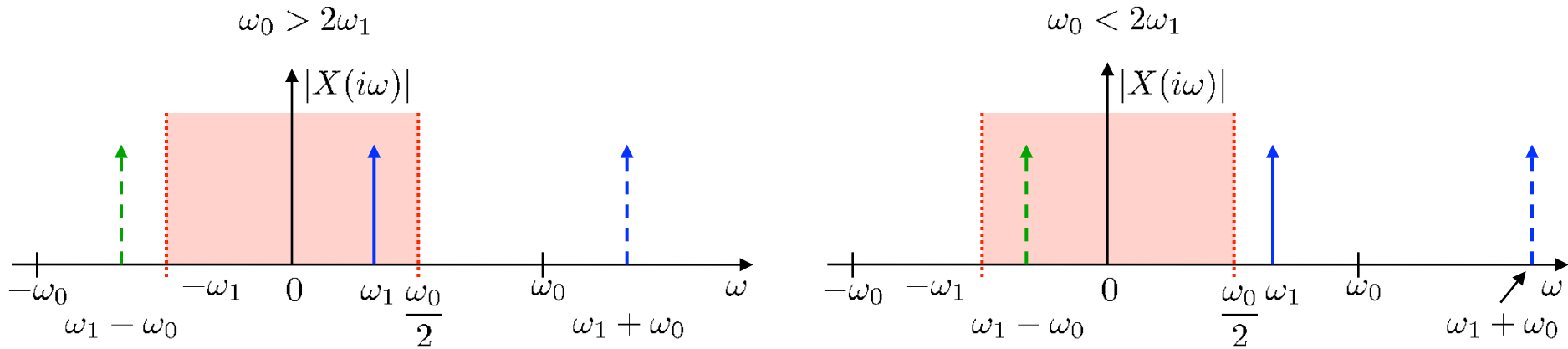


Spektrum des abgetasteten Signals $\omega_0 < 2\omega_{\max}$



2.6 Abtasttheorem und Aliasing

Aliasing bei der Abtastung einer Sinus-Schwingung mit Kreisfrequenz ω_1



Jede Signalkomponente der Frequenz ω_1 wird durch das Abtasten an folgende Frequenzen gespiegelt:

$$\omega_l = \omega_1 + l\omega_0 \quad \text{mit } l = \dots, -2, -1, 0, 1, 2, \dots$$

Solange ω_1 im roten Bereich bleibt, also das Abtasttheorem nicht verletzt, liegen die gespiegelten Komponenten außerhalb des roten Bereichs (linkes Bild).

Sobald ω_1 den roten Bereich verlässt, wandert eine gespiegelte Komponente hinein und verursacht Aliasing (rechtes Bild)!

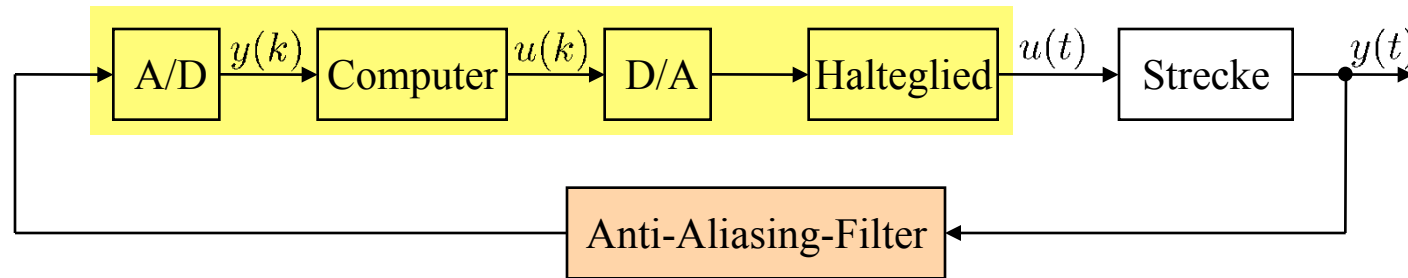
Wandert ω_1 bis ω_0 , so entsteht eine Alias-Komponente bei $\omega = 0$ (stehendes Speichenrad).

2.6 Abtasttheorem und Aliasing

Anti-Aliasing-Filter

Um Aliasing zu verhindern, muss das analoge Signal *vor* der Abtastung mit einem **Anti-Aliasing-Filter** gefiltert werden. Dieses Filter ist notwendigerweise ein analoges Filter, denn nach dem Abtasten kann das Aliasing nicht mehr rückgängig gemacht werden.

Typischerweise werden sehr steile Filter mit einer Grenzfrequenz bei ω_0 eingesetzt, um die niederfrequenten Signalanteile möglichst wenig zu verzerren aber alle Frequenzen oberhalb von ω_0 sehr stark zu dämpfen.



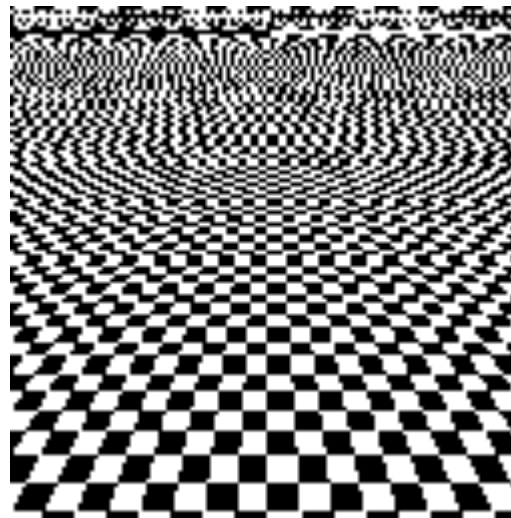
Mit einem Trick kann auch ein digitales Anti-Aliasing-Filter eingesetzt werden. Dazu muss zunächst mit viel höherer als der geplanten Abtastrate abgetastet werden, z.B. mit $8\omega_0$ (8-fach Oversampling). Die Frequenz muss so hoch sein, dass dort keine signifikanten Signalkomponenten mehr existieren. Die gespiegelten Frequenzen liegen dann weit außerhalb des Bereichs $|\omega| < \omega_0/1$. Danach wird digital gefiltert und downgesampelt.

2.6 Abtasttheorem und Aliasing

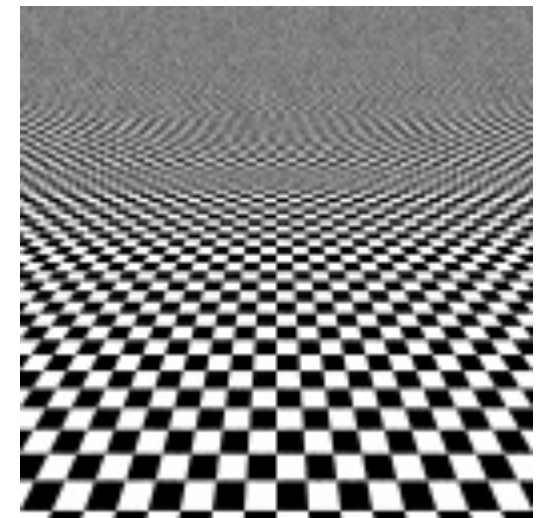
Aliasing in der Bildverarbeitung

Die Signalverarbeitung taucht nicht nur im Zusammenhang mit *zeitlichen* Signalen auf. Auch für *räumliche* Signale wie z.B. Bilder (2-dim. räumliche Signale) gelten die selben Gesetze. So wie man zeitliche Signale filtern kann, ist Gleiches auch mit räumlichen Signalen möglich. In der Bildverarbeitung tritt deshalb ebenfalls der Aliasing-Effekt auf. Eine hohe (räumliche) Frequenz entspricht einer dichten Abfolge von abwechselnd hellen und dunklen Punkten. Wenn bei der Reduzierung der Auflösung (entspricht der Abtastung) kein Anti-Aliasing-Filter verwendet wird, spiegeln die hohen Frequenzen in die niederfrequenten Bereiche und stören das Bild. Ähnliches kennt man vom Fernsehen, wenn jemand Kleidung mit kleinem Karomuster trägt. In der Bildverarbeitung nennt man dies **Moiré-Effekt**.

Ohne Anti-Aliasing



Mit Anti-Aliasing



3. Kurzübersicht: Zeitdiskrete Systeme

Inhalt Kapitel 2

3. Zeitdiskrete Signale und Systeme

3.1 Zeitdiskrete Signale

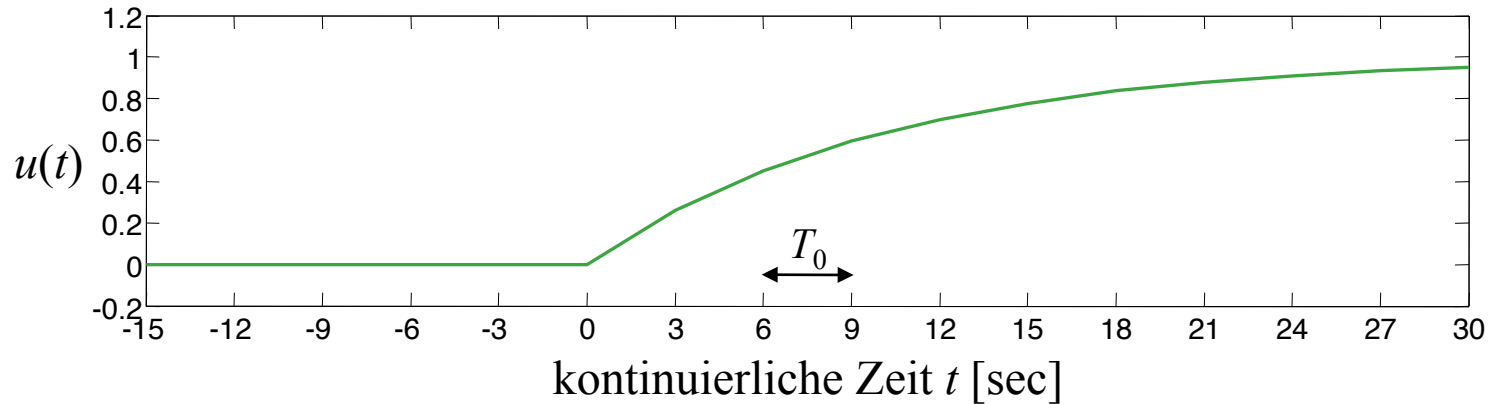
3.2 Differenzengleichungen

3.3 Z-Transformation

3.4 Übertragungsfunktionen

3.1 Zeitdiskrete Signale

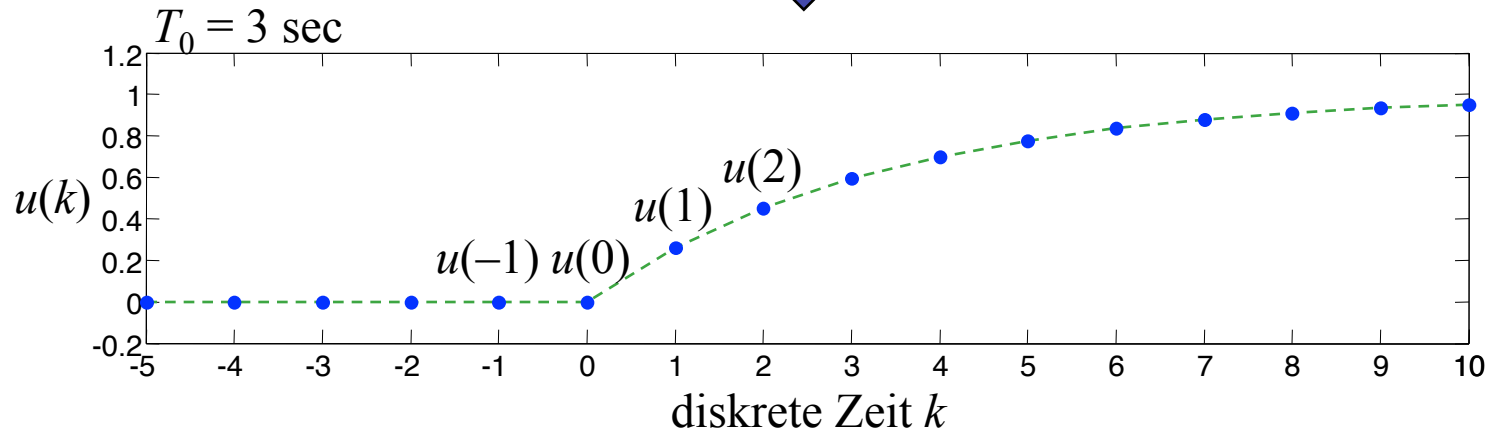
Äquidistante Abtastung eines zeitkontinuierlichen Signals mit Abtastzeit T_0



Abtastung zu den
Abtastzeitpunkten $t = kT_0$

$k = \dots, -2, -1, 0, 1, 2, 3, \dots$

Folge: $\{u(k)\} = \{\dots, 0, 0, 0, 0.26, 0.45, 0.59, \dots\}$

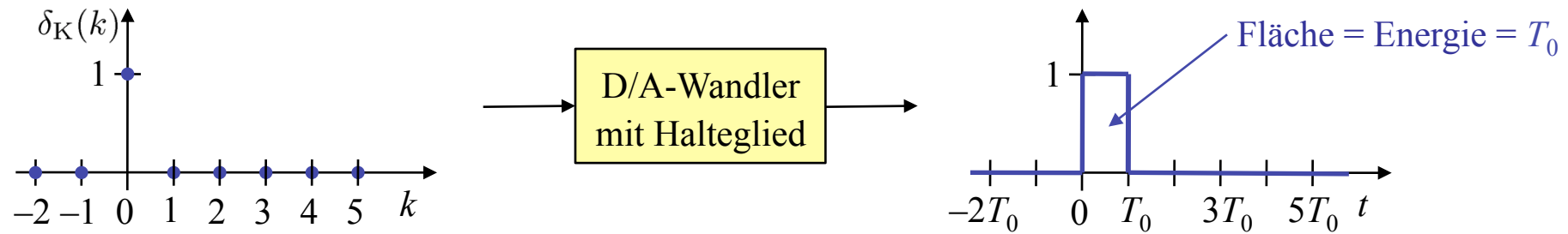


3.1 Zeitdiskrete Signale

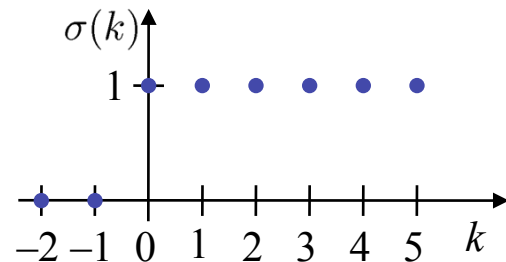


Einheitsimpuls und Einheitssprung

Der zeitdiskrete Einheitsimpuls (Kronecker-Delta) ist mit Höhe 1 hingegen unterschiedlich zum zeitkontinuierlichen Dirac-Impuls (Höhe ∞) definiert. Daher kann der Einheitsimpuls im Zeitdiskreten tatsächlich in der Praxis realisiert werden und stellt keinen, nur theoretischen Grenzfall dar. Wird ein zeitdiskreter Einheitsimpuls auf einen D/A-Wandler mit Halteglied gegeben, so ist seine Länge und damit die darin enthaltene Energie proportional zu T_0 .



Der zeitdiskrete Einheitssprung entspricht einfach dem abgetasteten zeitkontinuierlichen Einheitssprung. Während des 1. Abtastschrittes sind $\sigma(k)$ und $\delta_K(k)$ identisch!



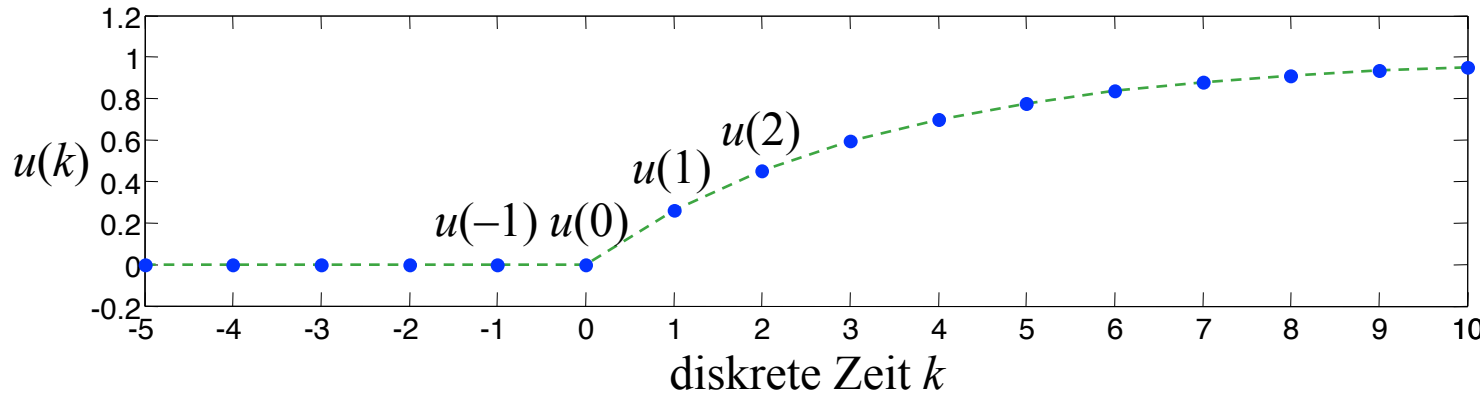
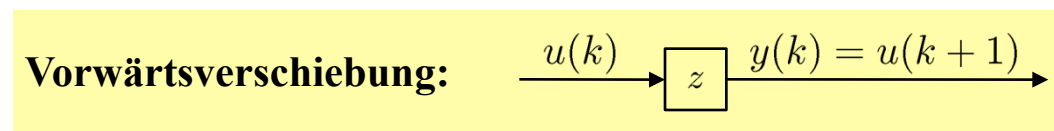
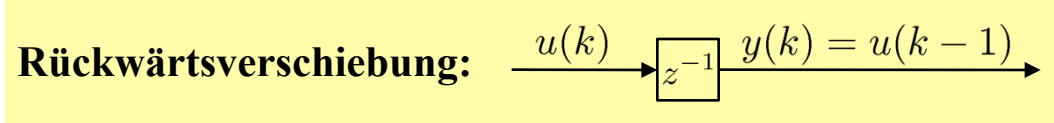
Beziehung: $\delta_K(k) = \sigma(k) - \sigma(k - 1)$

Das entspricht: $\delta(t) = \frac{d}{dt} \sigma(t) = \lim_{T_0 \rightarrow 0} \frac{\sigma(t) - \sigma(t - T_0)}{T_0}$

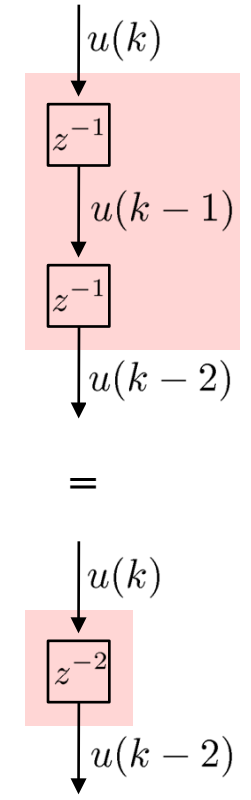
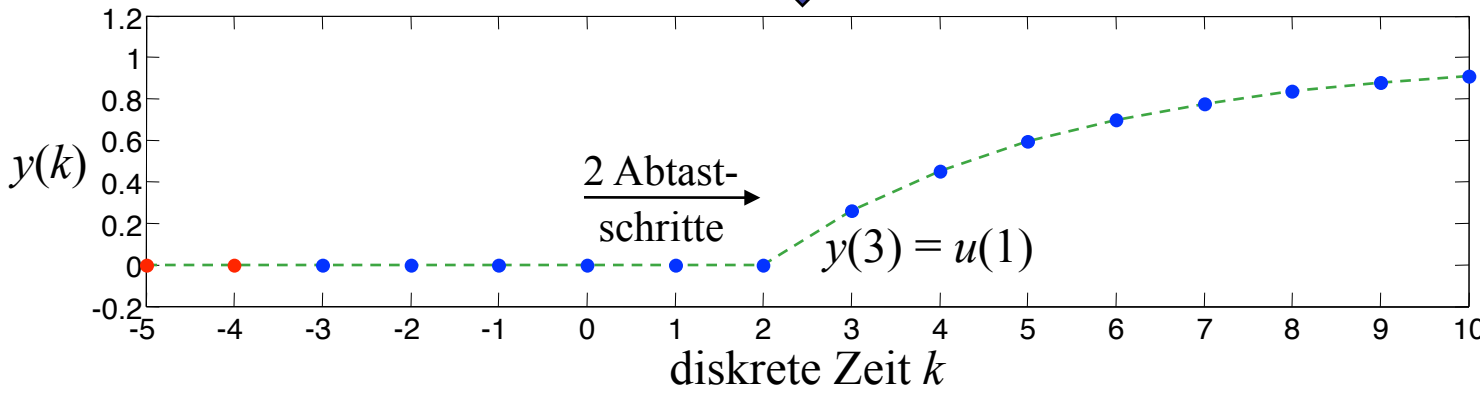
3.1 Zeitdiskrete Signale

Zeitlicher Verschiebe-Operator

$$z^i = \begin{cases} i < 0 : & i \text{ Schritte Verzögerung} \\ i = 0 : & \text{keine Verschiebung} \\ i > 0 : & i \text{ Schritte Prädiktion} \end{cases}$$



Verzögerung um 2 Schritte: $z^{-2} \downarrow y(k) = u(k-2)$

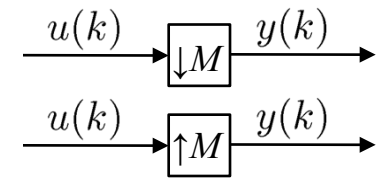


3.1 Zeitdiskrete Signale

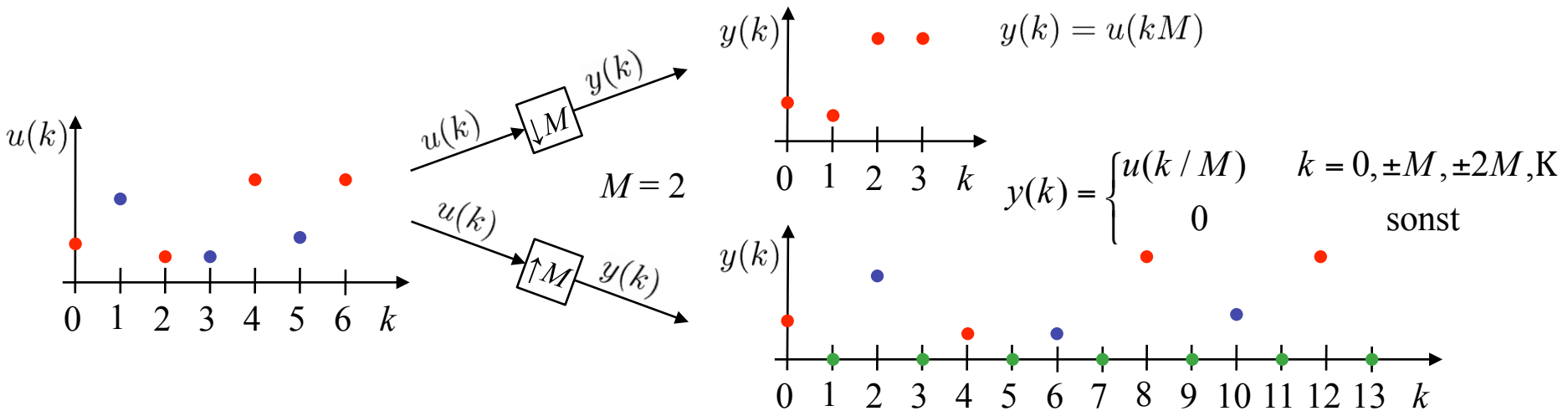
Up-Sampling und Down-Sampling

Wenn die Abtastfrequenz eines bereits abgetasteten, d.h. zeitdiskreten Signals verändert werden soll, dann werden folgende Operationen benötigt:

- *Down-Sampling*: Reduktion der Abtastfrequenz um den Faktor M .
- *Up-Sampling*: Erhöhung der Abtastfrequenz um den Faktor M .



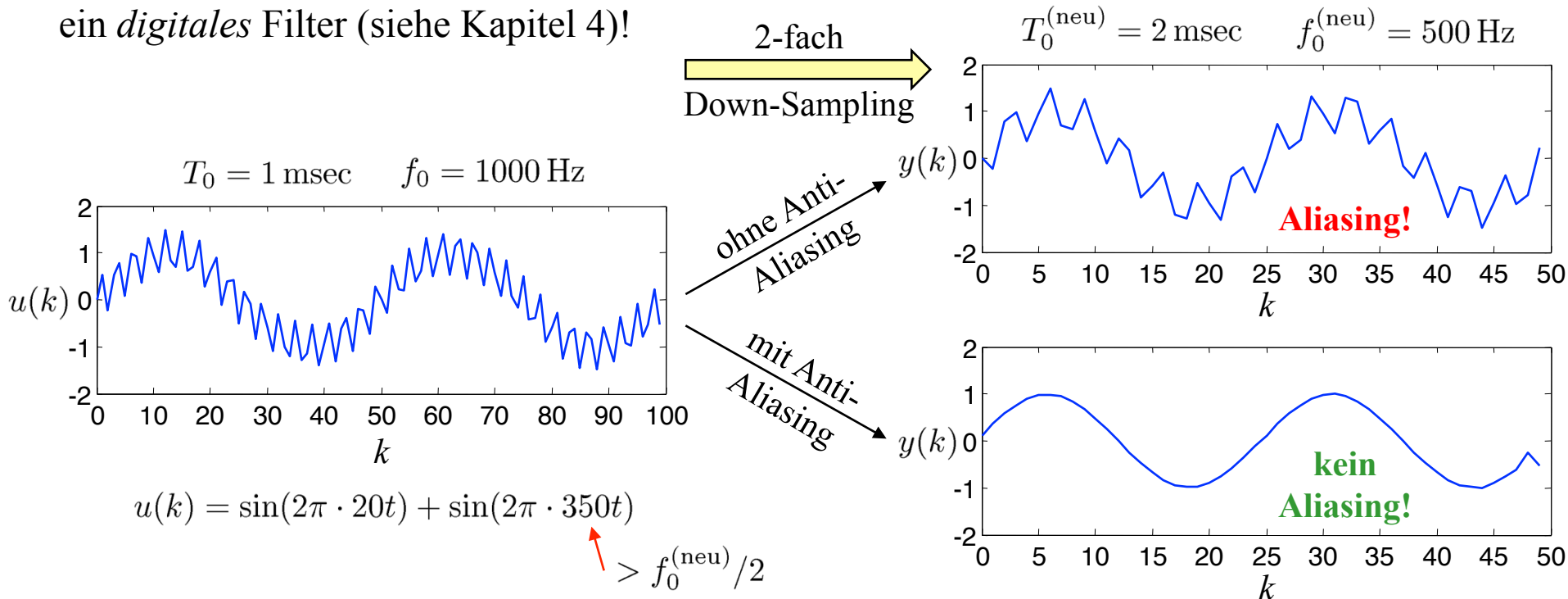
Diese Operationen werden z.B. benötigt, um unterschiedlich schnell abgetastete Signale auf eine gemeinsame Abtastzeit zu synchronisieren oder um für versehentlich bzw. vorsichtshalber (um Informationsverlust zu vermeiden) zu schnell abgetastete Signale den Speicherbedarf zu reduzieren und die numerischen Eigenschaften zu verbessern.



3.1 Zeitdiskrete Signale

Aliasing beim Down-Sampling

Beim Abtasten eines zeitkontinuierlichen Signals kommt es zu Aliasing, wenn die Abtastfrequenz f_0 nicht mindestens 2 Mal größer ist als die höchste Signalfrequenz f_{\max} . Dasselbe gilt auch für das Down-Sampling eines zeitdiskreten Signals! Vor dem Down-Sampling müssen daher alle Frequenzen größer als $f_0/2$ mit einem *Anti-Aliasing-Filter* herausgefiltert werden. Dieses Anti-Aliasing-Filter ist hierbei ein *digitales* Filter (siehe Kapitel 4)!



3.2 Differenzengleichungen

DGL → Differenzgleichung

Für kleine Abtastzeiten $T_0 \rightarrow 0$ kann jede DGL näherungsweise diskretisiert werden, indem man die Ableitungen durch Differenzenquotienten ersetzt:

$$\dot{x}(t) \approx \frac{x(t) - x(t - T_0)}{T_0}, \quad \ddot{x}(t) \approx \frac{\dot{x}(t) - \dot{x}(t - T_0)}{T_0} = \frac{x(t) - 2x(t - T_0) + x(t - 2T_0)}{T_0^2}, \quad \dots$$

Diese Näherungsmethode weist für $T_0 \gg 0$ erhebliche Nachteile auf. Für einen Überblick über alternative Methoden siehe Kapitel 4. Aus einer DGL n . Ordnung ($m \leq n$)

$$y(t) + a_1 \dot{y}(t) + a_2 \ddot{y}(t) + \dots + a_n y^{(n)}(t) = b_0 u(t) + b_1 \dot{u}(t) + b_2 \ddot{u}(t) + \dots + b_m u^{(m)}(t)$$

wird so eine Differenzengleichungen ebenfalls n . Ordnung:

$$y(k) + a_1 y(k-1) + a_2 y(k-2) + \dots + a_n y(k-n) = b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) + \dots + b_m u(k-m)$$

Während die Simulation eines zeitkontinuierlichen Systems die Integration der DGL erfordert, lässt sich ein zeitdiskretes System ganz einfach durch Auflösen der Differenzgleichung nach $y(k)$ simulieren, wenn man die n vergangenen Ein- und Ausgangswerte kennt:

$$y(k) = b_0 u(k) + b_1 u(k-1) + b_2 u(k-2) + \dots + b_m u(k-m) - a_1 y(k-1) - a_2 y(k-2) - \dots - a_n y(k-n)$$

3.2 Differenzgleichungen

Moving Average (MA) System

Der Ausgang berechnet sich aus einer gewichteten Mittelung des verzögerten Eingangssignals:

$$y(k) = b_0u(k) + b_1u(k-1) + b_2u(k-2) + \dots + b_mu(k-m)$$

Ein solches System wird auch als FIR (*finite impulse response*) charakterisiert, da es eine auf exakt null abfallende Impulsantwort (Gewichtsfunktion) aufweist.

Autoregressive (AR) System

Der Ausgang berechnet sich aus einer gewichteten Mittelung des verzögerten Ausgangssignals:

$$y(k) = -a_1y(k-1) - a_2y(k-2) - \dots - a_ny(k-n)$$

Ein solches System weist eine IIR (*infinite impulse response*), d.h. eine unendliche Impulsantwort (Gewichtsfunktion) auf.

Moving Average Autoregressive (ARMA) System

Hierbei kommen sowohl MA- als auch AR-Terme vor. Dies stellt die allgemeinste Beschreibungsform dar und weist natürlich ebenfalls eine IIR auf.

3.2 Differenzengleichungen

Homogene Lösung: Simulation für $u(k) = 0$

Wenn der Eingang $u(k) = 0$ ist, hängt der Zeitverlauf nur von den Anfangswerten ab. Als einfaches Beispiel betrachten wir eine Differenzengleichung erster Ordnung mit $b_1 = 0$:

$$y(k) = b_0 u(k) - a_1 y(k-1)$$

Wenn die Anfangsbedingung $y(-1)$ bekannt ist, kann der Zeitverlauf der Ausgangsgröße $y(k)$ für alle k berechnet werden:

$$k = 0 : y(0) = b_0 u(0) - a_1 y(-1) = -a_1 y(-1)$$

$$k = 1 : y(1) = b_0 u(1) - a_1 y(0) = -a_1 y(0) = (-a_1)^2 y(-1)$$

$$k = 2 : y(2) = b_0 u(2) - a_1 y(1) = -a_1 y(1) = (-a_1)^3 y(-1)$$

⋮

$$k : y(k) = b_0 u(k) - a_1 y(k-1) = -a_1 y(k-1) = (-a_1)^{k+1} y(-1)$$

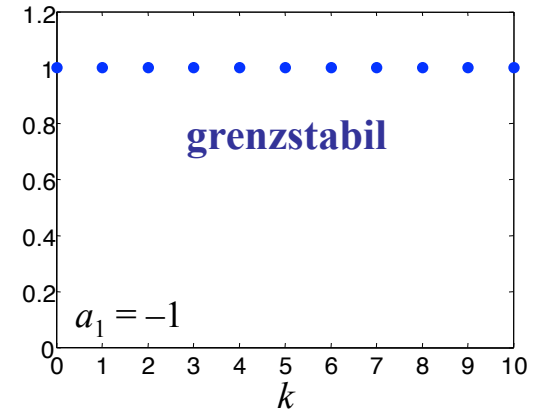
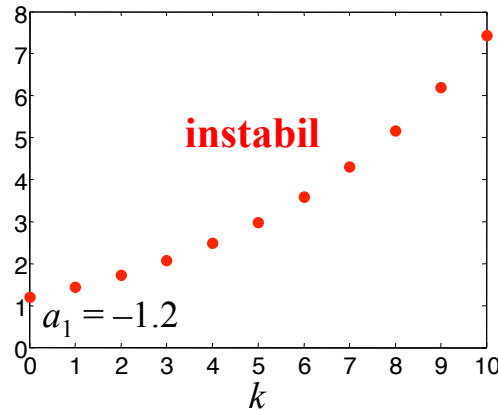
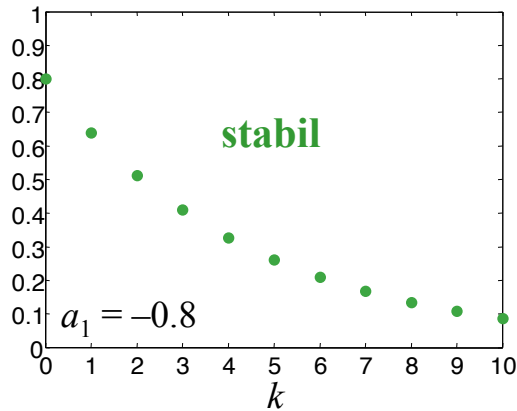
Stabil:	$ a_1 < 1$
Instabil:	$ a_1 > 1$
Grenzstabil:	$ a_1 = 1$

Für Differenzengleichungen n . Ordnung mit $n > 1$ läuft die Berechnung entsprechend, allerdings werden dann die n Anfangsbedingungen $y(-1), y(-2), \dots, y(-n)$ benötigt, weil $y(k)$ von $y(k-1), y(k-2), \dots, y(k-n)$ abhängt.

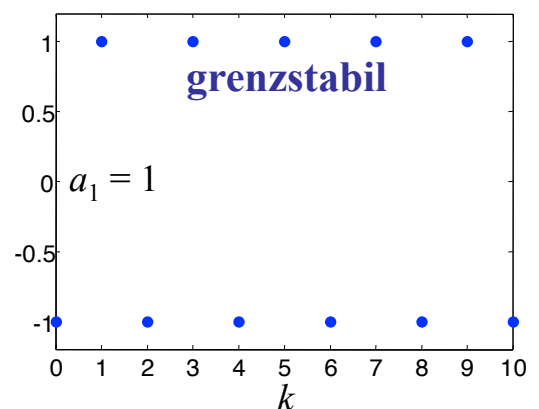
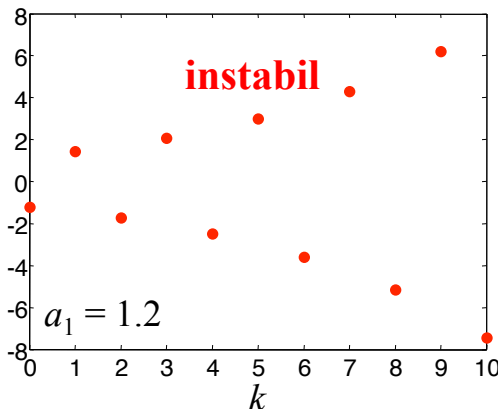
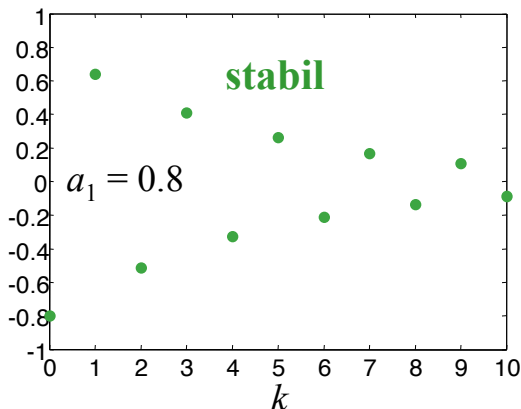
3.2 Differenzengleichungen

Stabilität einer Differenzengleichung 1. Ordnung

An der obigen homogenen Lösung lassen sich leicht folgende 3 Fälle unterscheiden:



Sollte $a_1 > 0$ sein, ergeben sich alternierende (zwischen positiv und negativ springende) Zeitverläufe, zu denen keine Entsprechung bei den zeitkontinuierlichen Systemen existiert:



3.2 Differenzengleichungen

$$u(k) = \delta_K(k) = \begin{cases} 1 & k = 0 \\ 0 & \text{sonst} \end{cases}$$

Impulsantwort

Für $u(k) = \delta_K(k)$ ergibt sich für den Ausgang $y(k)$ die Impulsantwort, die wie im Zeitkontinuierlichen, mit der Gewichtsfolge $g(k)$ identisch ist. Wir nehmen der Einfachheit halber an, dass alle Anfangsbedingungen = 0 sind und somit die homogene Lösung wegfällt. Als einfaches Beispiel betrachten wir eine Differenzengleichung erster Ordnung mit $b_1 = 0$:

$$y(k) = b_0 u(k) - a_1 y(k-1)$$

Wenn die Anfangsbedingung $y(-1) = 0$ ist, kann der Zeitverlauf der Ausgangsgröße $y(k)$ für alle k wie folgt berechnet werden:

$$k = 0 : y(0) = b_0 u(0) - a_1 y(-1) = b_0$$

$$k = 1 : y(1) = b_0 u(1) - a_1 y(0) = -a_1 b_0$$

$$k = 2 : y(2) = b_0 u(2) - a_1 y(1) = (-a_1)^2 b_0$$

⋮

$$k : y(k) = (-a_1)^k b_0$$

Es ergibt sich also ein qualitativ ähnlicher zeitlicher Verlauf wie bei der homogenen Lösung.

3.2 Differenzengleichungen

$$u(k) = \sigma(k) = \begin{cases} 1 & k \geq 0 \\ 0 & k < 0 \end{cases}$$

Sprungantwort

Für $u(k) = \sigma(k)$ ergibt sich für den Ausgang $y(k)$ die Sprungantwort. Wir nehmen wieder der Einfachheit halber an, dass alle Anfangsbedingungen = 0 sind und somit die homogene Lösung wegfällt. Als einfaches Beispiel betrachten wir wieder eine Differenzengleichung erster Ordnung mit $b_1 = 0$:

$$y(k) = b_0 u(k) - a_1 y(k-1)$$

Wenn die Anfangsbedingung $y(-1) = 0$ ist, kann der Zeitverlauf der Ausgangsgröße $y(k)$ für alle k wie folgt berechnet werden:

$$k = 0 : y(0) = b_0 u(0) - a_1 y(-1) = b_0 \quad (\text{identisch mit Impulsantwort})$$

$$k = 1 : y(1) = b_0 u(1) - a_1 y(0) = b_0 - a_1 b_0 = b_0 (1 - a_1)$$

$$k = 2 : y(2) = b_0 u(2) - a_1 y(1) = b_0 - a_1 b_0 (1 - a_1) = b_0 (1 - a_1 + a_1^2)$$

⋮

$$k : y(k) = b_0 (1 - a_1 + a_1^2 - \dots - (-a_1)^k) = b_0 \sum_{i=0}^k (-a_1)^i$$

3.2 Differenzgleichungen

Beziehungen zwischen Impuls- und Sprungantwort

Erinnerung: Im *Zeitkontinuierlichen* galten zwischen der Impulsantwort $g(t)$ und der Sprungantwort $h(t)$ die Beziehungen:

$$\delta(t) = \frac{d}{dt}\sigma(t) \quad \rightarrow \quad g(t) = \frac{d}{dt}h(t)$$

$$\sigma(t) = \int_0^t \delta(\tau)d\tau \quad \rightarrow \quad h(t) = \int_0^t g(\tau)d\tau$$

Aus den zuvor berechneten zeitdiskreten Impuls- und Sprungfolgen ergeben sich im *Zeitdiskreten* vergleichbare Zusammenhänge:

$$\delta_K(k) = \sigma(k) - \sigma(k-1) \quad \rightarrow \quad g(k) = h(k) - h(k-1)$$

$$\sigma(k) = \sum_{i=0}^k \delta_K(k-i) \quad \rightarrow \quad h(k) = \sum_{i=0}^k g(k-i)$$

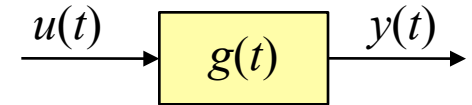
D.h. Differenzen ersetzen die Ableitungen, Summen ersetzen die Integrale. Im Zeitdiskreten lässt sich also wesentlich leichter rechnen als im Zeitkontinuierlichen, insbesondere mit Hilfe eines Computers. Die Anzahl der Summanden steigt allerdings mit der Simulationszeit k an!

3.2 Differenzgleichungen

Faltungssumme

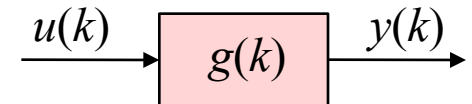
Die Gewichtsfolge (Impulsantwort) kennzeichnet alle Eigenschaften eines linearen, zeitdiskreten, dynamischen Systems, genauso wie die Gewichtsfunktion dies für zeitkontinuierliche Systeme tut. Die Systemantwort auf ein beliebiges Eingangssignal $u(t)$ lässt sich für zeitkontinuierliche System mit Hilfe des *Faltungsintegrals* bestimmen:

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



Im Zeitdiskreten lässt sich entsprechend die Antwort auf jedes beliebige Eingangssignal $u(k)$ mit Hilfe der *Faltungssumme* bestimmen:

$$y(k) = \sum_{i=-\infty}^{\infty} g(i)u(k - i) = \sum_{i=-\infty}^{\infty} g(k - i)u(i)$$



Wir setzen normalerweise voraus, dass $u(k) = 0$ für $k < 0$; dann muss die erste Summe nur bis $i = k$ und die zweite Summe erst ab $i = 0$ laufen. Ist zudem das System kausal, d.h. $g(k) = 0$ für $k < 0$, dann muss die erste Summe erst ab $i = 0$ und die zweite Summe nur bis $i = k$ laufen.

3.2 Differenzgleichungen

Faltungssumme (vereinfacht)

Mit diesen Vereinfachungen ergibt sich für die erste Formulierung der Faltungssumme:

$$y(k) = \sum_{i=0}^k g(i)u(k-i) = g(0)u(k) + g(1)u(k-1) + \dots + g(k)u(0)$$

In der zweiten Formulierung der Faltungssumme ist die Reihenfolge umgekehrt:

$$y(k) = \sum_{i=0}^k g(k-i)u(i) = g(k)u(0) + g(k-1)u(1) + \dots + g(0)u(k)$$

Offensichtlich sind beide Formulierungen identisch!

Die Faltungssumme ist, insbesondere mit Computer-Hilfe, wesentlich einfach zu berechnen als das Faltungsintegral.

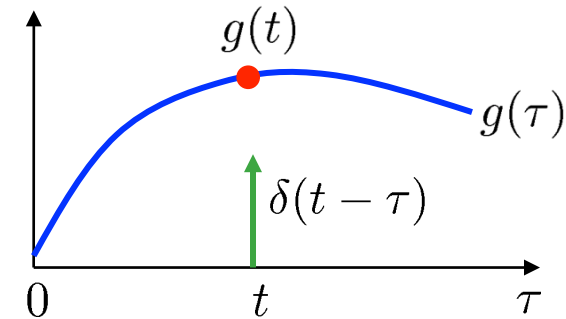
ACHTUNG: Mit fortschreitender Simulationszeit $k \rightarrow \infty$, wächst die Anzahl der Summanden linear an. Wenn also die Gewichtsfolge unendlich lang ist (IIR), dann wächst auch der Rechenaufwand und Speicherbedarf unendlich stark an. D.h. für die Simulation von IIR-Systemen müssen wir uns eine geschicktere Alternative überlegen. Bei Gewichtsfunktionen der Länge L (FIR) bleibt die Anzahl der Summanden hingegen auf L beschränkt.

3.2 Differenzgleichungen

Faltung mit einem Impuls

Im Zeitkontinuierlichen hat der Dirac-Impuls $\delta(t)$ die Ausblendeigenschaft, d.h. die Faltung eines Zeitsignals mit einem Dirac-Impuls liefert wieder das Zeitsignal. Der Dirac-Impuls ist also bzgl. der Faltung das neutrale Element (wie die „0“ bei der Addition und die „1“ bei der Multiplikation). Zur Berechnung der Impulsantwort wählen wir $u(t) = \delta(t)$ und berechnen als Impulsantwort mittels des Faltungsintegrals und der Ausblendeigenschaft:

$$y(t) = \int_{-\infty}^{\infty} g(\tau)\delta(t - \tau)d\tau = g(t)$$

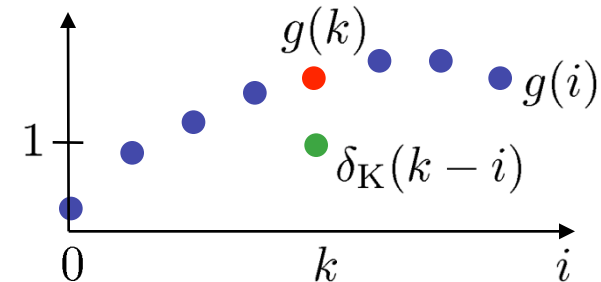


D.h. die Impulsantwort eines linearen, kontinuierlichen, dynamischen Systems ist identisch mit dessen Gewichtsfunktion!

Im Zeitdiskreten wählen wir $u(k) = \delta_K(k)$ und berechnen mittels der Faltungssumme:

$$y(k) = \sum_{i=-\infty}^{\infty} g(i)\delta_K(k - i) = g(k) \quad \text{= 1 für } k = i$$

Auch hier gilt also die Ausblendeigenschaft und die Äquivalenz von Impulsantwort und Gewichtsfolge.



3.2 Differenzgleichungen

Geometrische Reihe

In den vorherigen Folien tauchte immer wieder eine geometrische Folge oder Reihe auf. Eine geometrische Reihe ist die Summe einer exponentiell verlaufenden Folge von Zahlen:

$$\sum_{k=0}^{\infty} x^k = x^0 + x^1 + x^2 + x^3 + \dots$$

Mit folgendem Trick kann man den Wert dieser unendlichen Reihe berechnen:

$$q = x^0 + x^1 + x^2 + x^3 + \dots$$

$$xq = x^1 + x^2 + x^3 + x^4 + \dots$$

$$q - xq = x^0 \rightarrow q(1 - x) = 1 \rightarrow q = \frac{1}{1 - x}$$

Daraus folgt für $|x| < 1$ (für $|x| \geq 1$ divergiert die Reihe gegen unendlich):

$$\frac{1}{1 - x} = \sum_{k=0}^{\infty} x^k = x^0 + x^1 + x^2 + x^3 + \dots$$

Eine erweiterte Formel lässt sich für die endliche Reihe herleiten:

$$\frac{1 - x^{n+1}}{1 - x} = \sum_{k=0}^n x^k$$

3.3 Z-Transformation

Abkürzung: $u(k) = u_c(kT_0)$

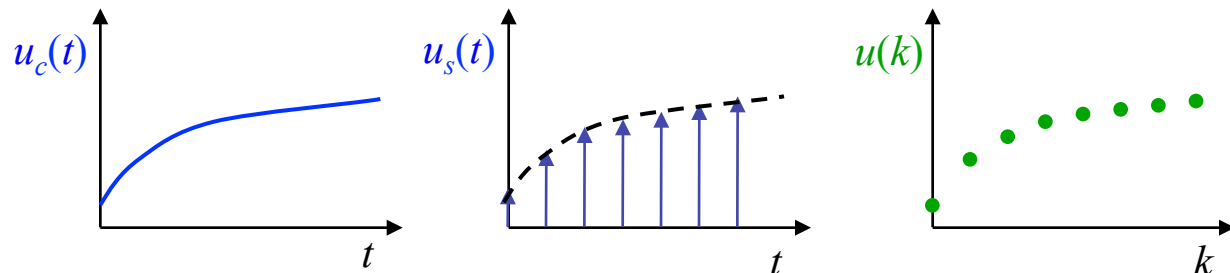
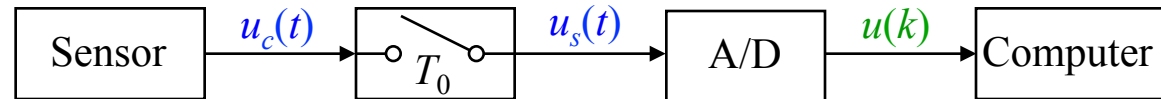
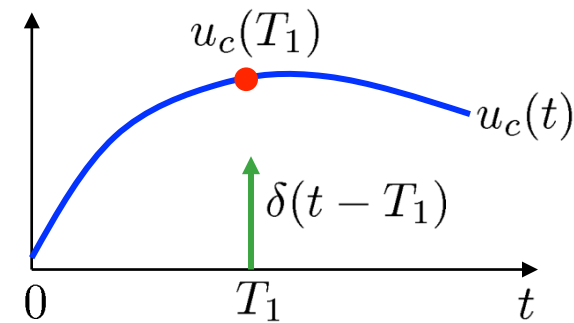
Beschreibung abgetasteter Signale

Ein Abtaster (*Sampler*) mit A/D-Wandler erzeugt aus dem zeitkontinuierlichen Signal $u_c(t)$ das zeitdiskrete Signal $u(k) = u_c(kT_0)$. Eine Abtastung des kontinuierlichen Signals zum Zeitpunkt T_1 wird als Multiplikation von $u_c(t)$ mit einem Dirac-Impuls zum Zeitpunkt T_1 , also mit $\delta(t-T_1)$, beschrieben:

$$u_s(t) = u_c(t)\delta(t - T_1) = u_c(T_1)\delta(t - T_1)$$

Wird nun periodisch zu den Zeitpunkten kT_0 abgetastet, so muss $u_c(t)$ mit einer ganz Impulsfolge (*impulse train*) multipliziert (moduliert) werden:

$$\begin{aligned} u_s(t) &= u_c(t) \sum_{k=-\infty}^{\infty} \delta(t - kT_0) \\ &= \sum_{k=-\infty}^{\infty} u_c(kT_0)\delta(t - kT_0) \\ &= \sum_{k=-\infty}^{\infty} u(k)\delta(t - kT_0) \end{aligned}$$



3.3 Z-Transformation

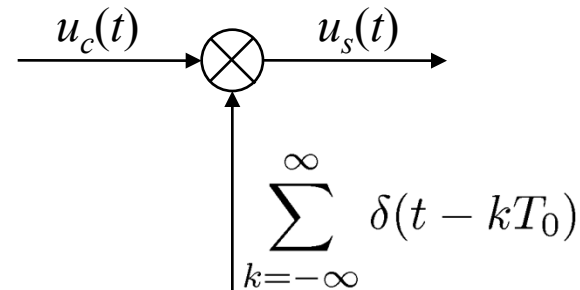
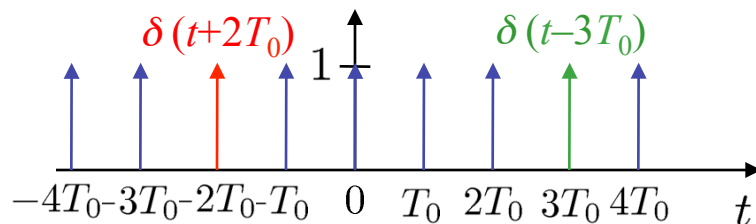
Interpretation der modulierten Impulsfolge

Die zeitkontinuierliche Beschreibung des abgetasteten Signals als modulierte Impulsfolge

$$u_s(t) = \sum_{k=-\infty}^{\infty} u(k)\delta(t - kT_0) \quad \text{bzw.} \quad u_s(t) = \sum_{k=0}^{\infty} u(k)\delta(t - kT_0), \text{ wenn } u(k) = 0 \text{ f\u00fcr } k < 0$$

stellt nur eine mathematische Modellvorstellung dar. Die unendlich hohen und schmalen Dirac-Impulse existieren in Wirklichkeit nicht. Sie sind aber ein elegantes Mittel, dem abgetasteten Signal, da ja aus isolierten Punkten besteht, eine Energiemenge zuzuordnen (ein Dirac-Impuls hat die Fl\u00e4che 1 und das entspricht auch seiner Energiemenge). Deswegen macht auch eine Multiplikation eines Dirac-Impulses mit einem Signal, hier $u(k)$, Sinn. Obwohl $u(k) \cdot \infty = \infty$, \u00e4ndert sich doch der Energiegehalt von 1 auf $u(k)$.

Mathematisches Modell des Abtastvorgangs:



3.3 Z-Transformation

Laplace-Transformation eines abgetasteten Signals

Wendet man die Laplace-Transformation auf abgetastete Signale an, so entsteht die sog. *z-Transformation*. Die Laplace-Transformierte $U(s)$ eines zeitkontinuierlichen Signals $u(t)$ ist definiert als:

Laplace-Transformation:
$$U(s) = \int_0^{\infty} u(t)e^{-st} dt$$

Wählen wir für $u(t)$ ein abgetastetes Signal, d.h. $u(t) = u_s(t)$, dann ergibt sich:

$$\begin{aligned} U_s(s) &= \int_0^{\infty} u_s(t)e^{-st} dt = \int_0^{\infty} \sum_{k=0}^{\infty} u(k)\delta(t - kT_0)e^{-st} dt \\ &= \sum_{k=0}^{\infty} u(k) \underbrace{\int_0^{\infty} \delta(t - kT_0)e^{-st} dt}_{\mathcal{L}\{\delta(t - kT_0)\}} \end{aligned}$$

Damit ergibt sich:

$$U_s(s) = \sum_{k=0}^{\infty} u(k)e^{-skT_0} = \sum_{k=0}^{\infty} u(k) (e^{sT_0})^{-k}$$

Zur Erinnerung:

$$\mathcal{L}\{\delta(t)\} = 1$$

$$\mathcal{L}\{\delta(t - kT_0)\} = e^{-skT_0}$$

3.3 Z-Transformation

Laplace-Transformation \rightarrow z-Transformation

Mit der Abkürzung

$$z = e^{sT_0}$$

nennt man die Laplace-Transformierte eines abgetasteten Signals die *z-Transformierte* (der Index „s“ entfällt, weil durch die Variable „z“ schon klar ist, dass es sich hierbei um abgetastete Signale handelt):

z-Transformation:

$$U(z) = \sum_{k=0}^{\infty} u(k)z^{-k}$$

Frequenzgang

Zur Ermittlung des Frequenzgangs werten wir die Laplace-Transformierte auf der imaginären Achse der s -Ebene aus, indem wir $s = i\omega$ für $\omega = 0 \dots \infty$ setzen. Der Frequenzgang einer z -Transformierten ergibt sich dann entsprechend aus $z = e^{i\omega T_0}$ für $\omega = 0 \dots \infty$. In der z -Ebene laufen wir zur Auswertung des Frequenzgangs also entlang des *Einheitskreises*. Es fällt auf, dass der Einheitskreis unendlich oft umlaufen wird, der Frequenzgang also periodisch ist. Ab $\omega T_0 = \pi$ wiederholt sich das Ganze! (Beachte die Symmetrie bzgl. $\pm\omega$!)

3.3 Z-Transformation

Herleitung der Periodizität des Frequenzgangs

Wir wollen die eben festgestellte Periodizität des Frequenzgangs näher untersuchen. Der Frequenzgang eines abgetasteten Systems ergibt sich für $z = e^{i\omega T_0}$:

$$U(i\omega) = \sum_{k=0}^{\infty} u(k) (e^{i\omega T_0})^{-k}$$

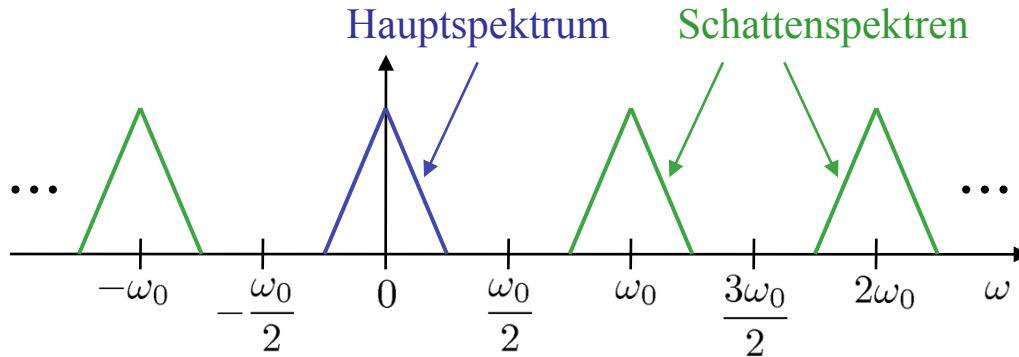
Mit den Zusammenhängen $e^{in2\pi} = 1$ für $n = 0, \pm 1, \pm 2, \dots$ und $\omega_0 T_0 = 2\pi$ lässt sich zeigen

$$\begin{aligned} U(i\omega) &= \sum_{k=0}^{\infty} u(k) (e^{i\omega T_0} e^{in2\pi})^{-k} = \sum_{k=0}^{\infty} u(k) \left(e^{i(\omega T_0 + n2\pi)} \right)^{-k} \\ &= \sum_{k=0}^{\infty} u(k) \left(e^{i(\omega T_0 + n\omega_0 T_0)} \right)^{-k} = \sum_{k=0}^{\infty} u(k) \left(e^{i(\omega + n\omega_0) T_0} \right)^{-k} \end{aligned}$$

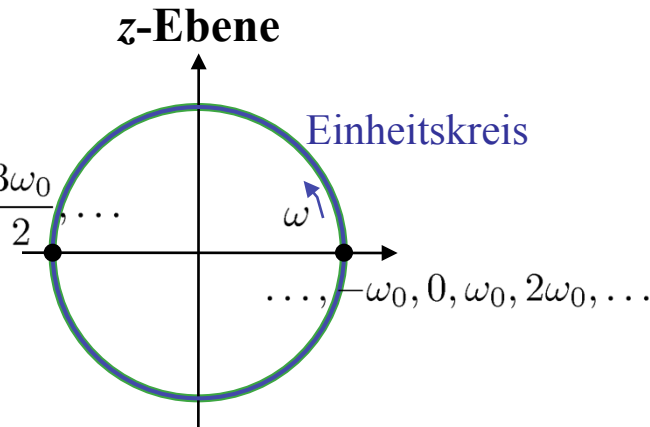
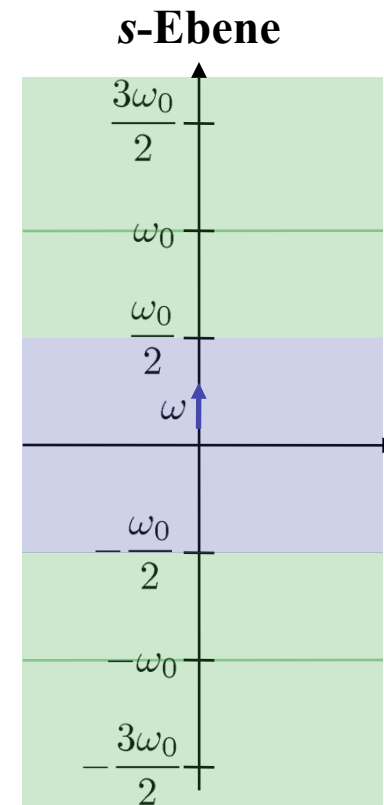
dass sich der Frequenzgang alle Vielfache von ω_0 wiederholt (je eine Umdrehung auf dem Einheitskreis). D.h. der Frequenzgang ist eine periodische Funktion und damit identisch für Frequenzen: $\omega, \omega \pm \omega_0, \omega \pm 2\omega_0, \omega \pm 3\omega_0, \dots$

3.3 Z-Transformation

Illustration der Periodizität des Frequenzgangs



- Die Schattenspektren rund um die Vielfachen von ω_0 werden durch die periodische Abtastung erzeugt.
- Die Im-Achse zwischen $-i\omega_0/2$ und $i\omega_0/2$ der s -Ebene wird auf den Einheitskreis der z -Ebene abgebildet.
- Die gesamte Information eines zeitdiskreten Systems ist im Frequenzgang entlang des Einheitskreises von $\omega = 0$ bis $\omega = \omega_0/2$ enthalten; der negative Halbkreis für $\omega = -\omega_0/2 \dots 0$ ist symmetrisch; alle weiteren Umkreisungen sind redundant!

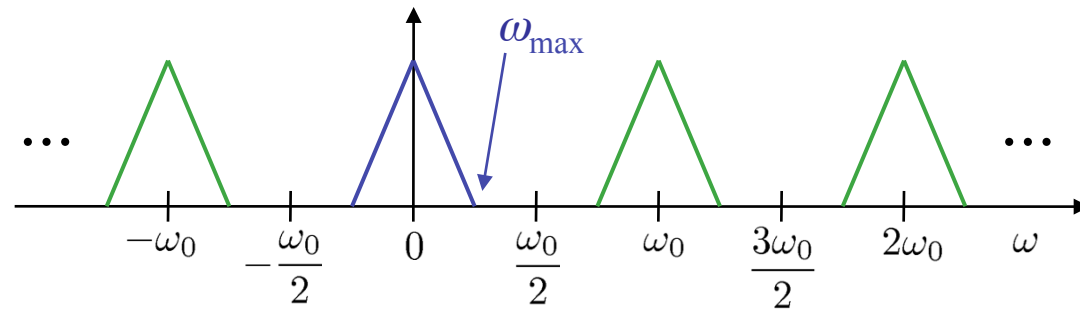


3.3 Z-Transformation

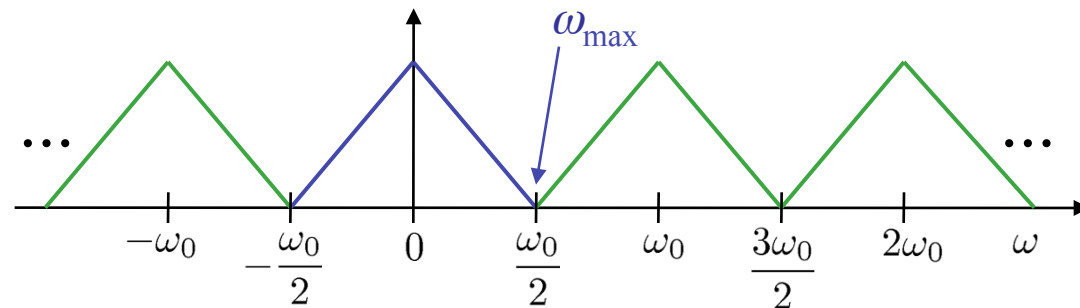
Abtasttheorem und Aliasing

- Solange die maximale Signalfrequenz ω_{\max} kleiner als die halbe Abtastfrequenz $\omega_0/2$ ist, kann das zeitkontinuierliche Signal aus dem abgetasteten perfekt rekonstruiert werden. Es tritt kein Informationsverlust auf, denn Haupt- und Schattenspektren sind getrennt (kein Aliasing).
- Sobald $\omega_{\max} > \omega_0/2$ überlappen sich Haupt- und Schattenspektren. Es tritt Aliasing auf, welches das Originalsignal verzerrt. Eine perfekte Rekonstruktion ist nicht mehr möglich (Informationsverlust!).

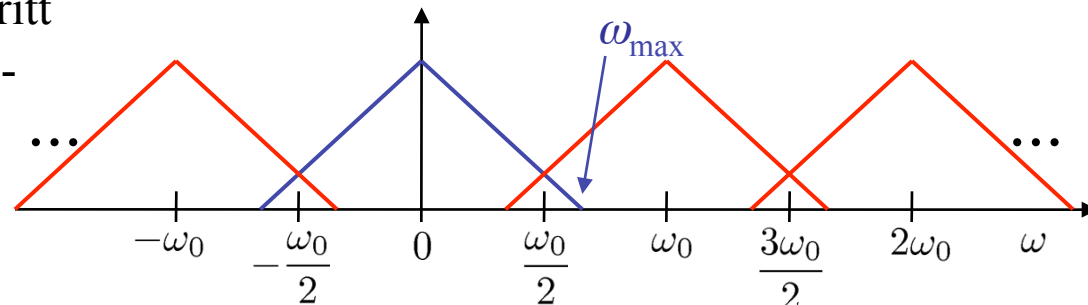
Abtastung schnell genug: kein Aliasing!



Grenzfall



Abtastung zu langsam: Aliasing!



3.3 Z-Transformation

Z-Transformation von Delta-Impulsen und Einheitssprüngen

Der Delta-Impuls $u(k) = \delta_K(k)$ hat folgende z -Transformierte:

$$u(0) = 1, u(1) = 0, u(2) = 0, \dots \rightarrow U(z) = 1z^0 + 0z^{-1} + 0z^{-2} + \dots \rightarrow U(z) = 1$$

Ein um 1 Zeitschritt verzögerter Delta-Impuls $u(k) = \delta_K(k-1)$ hat folgende z -Transformierte:

$$u(0) = 0, u(1) = 1, u(2) = 0, \dots \rightarrow U(z) = 0z^0 + 1z^{-1} + 0z^{-2} + \dots \rightarrow U(z) = z^{-1}$$

Ein um d Zeitschritte verzögerter Delta-Impuls $u(k) = \delta_K(k-d)$ hat folgende z -Transformierte:

$$u(0) = 0, \dots, u(d-1) = 0, u(d) = 1, u(d+1) = 0, \dots \rightarrow U(z) = z^{-d}$$

Der Einheitssprung $u(k) = \sigma(k)$ hat folgende z -Transformierte:

$$u(0) = 1, u(1) = 1, u(2) = 1, \dots \rightarrow U(z) = 1z^0 + 1z^{-1} + 1z^{-2} + \dots \rightarrow U(z) = \frac{1}{1 - z^{-1}}$$

Ein um d Zeitschritte verzögerter Einheitssprung $u(k) = \sigma(k-d)$ hat folgende z -Transformierte:

$$u(0) = 0, \dots, u(d-1) = 0, u(d) = 1, u(d+1) = 1, \dots \rightarrow U(z) = \frac{z^{-d}}{1 - z^{-1}}$$

Folgende Schreibweisen sind gleichwertig:
$$\frac{z^{-d}}{1 - z^{-1}} = \frac{z^{-d+1}}{z - 1} = \frac{1}{(1 - z^{-1})z^d} = \frac{1}{(z - 1)z^{d-1}}$$

3.3 Z-Transformation

Z-Transformation geometrischer Folgen

Die geometrische Folge $u(k) = a^k$ mit irgendeiner Zahl a kommt häufig vor, da sie einen exponentiellen Zeitverlauf darstellt. Diese Folge hat folgende z -Transformierte:

$$u(0) = a^0, u(1) = a^1, u(2) = a^2, u(3) = a^3, \dots \rightarrow U(z) = a^0 z^0 + a^1 z^{-1} + a^2 z^{-2} + a^3 z^{-3} + \dots$$

Weitere Umformung führt auf die Standardform der geometrischen Reihe:

$$U(z) = (az^{-1})^0 + (az^{-1})^1 + (az^{-1})^2 + (az^{-1})^3 + \dots = \sum_{k=0}^{\infty} (az^{-1})^k$$

Mit Hilfe der Summenformel für die unendliche geometrische Reihe ergibt sich:

$$U(z) = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}$$

Auf diese Art lassen sich unendliche Gewichtsfolgen in einfacher und geschlossener Weise als Übertragungsfunktionen schreiben.

Von der gebrochen rationalen Funktion zurück zur unendlichen Reihe gelangt man mit Hilfe der Polynomdivision.

Polynomdivision:

$$z : (z-a) = 1 + az^{-1} + a^2z^{-2} + \dots$$

$$\begin{array}{r} z-a \\ \underline{z-a} \\ a \\ \underline{a-a^2z^{-1}} \\ a^2z^{-1} \\ \underline{a^2z^{-1}-a^3z^{-2}} \\ a^3z^{-2} \end{array}$$

3.3 Z-Transformation

$$z = e^{sT_0}$$

$$u(k) \quad \circ \text{---} \bullet \quad U(z)$$

$$U(z) = \mathcal{Z}\{u(k)\}$$

Wichtigste Eigenschaften der z-Transformation

Für Grenzbetrachtungen werden häufig die Fälle $t \rightarrow 0$ ($k \rightarrow 0$) oder $t \rightarrow \infty$ ($k \rightarrow \infty$) untersucht. Dies bedeutet für den Bildbereich, d.h. die s - bzw. z -Variable:

$$t \rightarrow 0: s \rightarrow \infty$$

$$t \rightarrow \infty: s \rightarrow 0$$

$$k \rightarrow 0: z \rightarrow \infty$$

$$k \rightarrow \infty: z \rightarrow 1$$

Anfangswert

Der Anfangswert einer zeitdiskreten Folge kann ihrer z -Transformierten berechnet werden:

$$u(k=0) = \lim_{z \rightarrow \infty} U(z)$$

Endwert

Der Endwert (falls existent!) einer zeitdiskreten Folge kann ihrer z -Transformierten berechnet werden:

$$u(k \rightarrow \infty) = \lim_{z \rightarrow 1} (z-1)U(z)$$

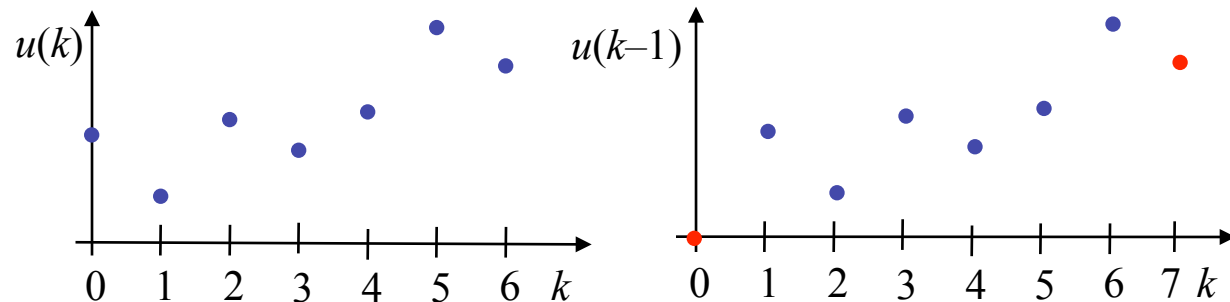
3.3 Z-Transformation

Rückwärtsverschiebung (nach rechts)

Eine Totzeit von $T_t = dT_0$ ist äquivalent mit einer Rückwärtsverschiebung (Verschiebung nach rechts) um d Abtastschritte. Diese Operation entspricht der Laplace-Transformierten e^{-sT_t} .

Dies bedeutet für den z -Bereich:

$$u(k-d) \quad \circ \rightarrow \bullet \quad z^{-d}U(z)$$

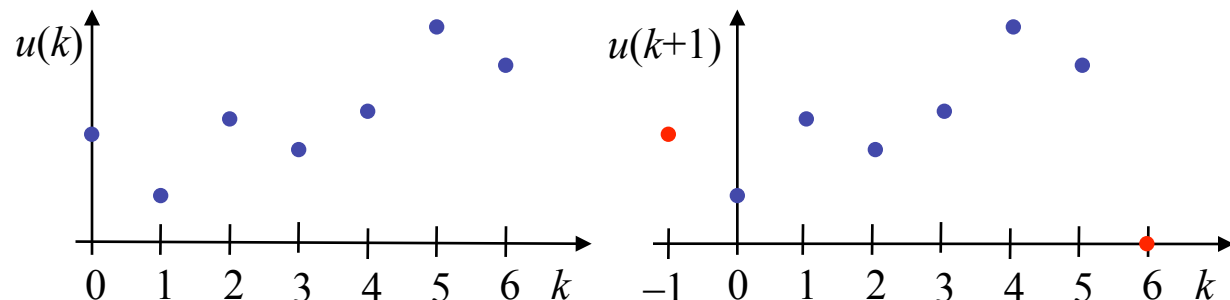


Vorwärtsverschiebung (nach links)

Eine Prädiktion (Vorhersage) um die Zeit $T_p = dT_0$ ist äquivalent mit einer Vorwärtsverschiebung (Verschiebung nach links) um d Abtastschritte. Diese Operation entspricht der Laplace-Transformierten e^{sT_p} .

Dies bedeutet für den z -Bereich:

$$u(k+d) \quad \circ \rightarrow \bullet \quad z^dU(z)$$

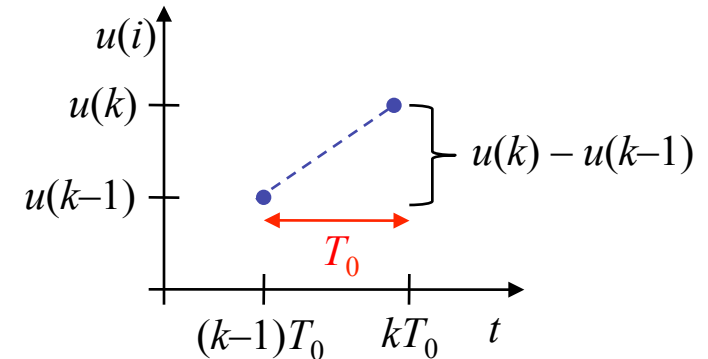


3.3 Z-Transformation

Differenzen / Differentiation

Die Differenz zweier aufeinanderfolgenden Abtastwerte ist, geteilt durch die Abtastzeit, gleich dem Differenzenquotienten 1. Ordnung und entspricht damit näherungsweise einer *Differentiation*. Im s -Bereich wird diese durch eine *Multiplikation mit s* realisiert. Im z -Bereich gilt:

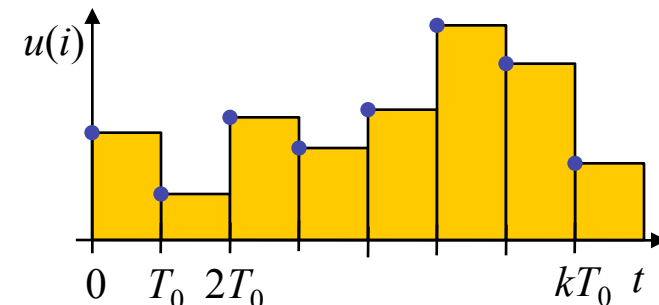
$$u(k) - u(k-1) \quad \circ \bullet \quad \frac{z-1}{z} U(z)$$



Summation / Integration

Die Summe aller Abtastwerte vom Zeitpunkt 0 ab ist, multipliziert mit der Abtastzeit, gleich der Untersumme-Approximation der Fläche unterhalb der Abtastwerte und entspricht damit näherungsweise einer *Integration*. Im s -Bereich wird diese durch eine *Division durch s* realisiert. Im z -Bereich gilt entsprechend:

$$\sum_{i=0}^k u(i) \quad \circ \bullet \quad \frac{z}{z-1} U(z)$$



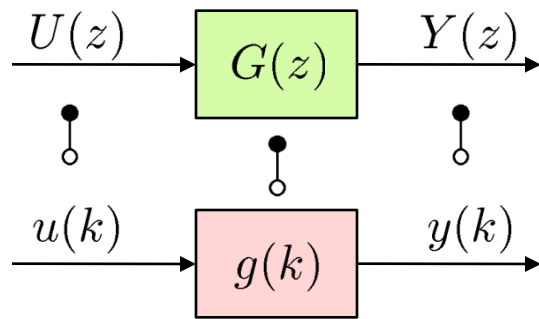
3.4 Übertragungsfunktionen

Z-Übertragungsfunktion und Gewichtsfolge

Bei zeitdiskreten Systemen besteht zwischen Übertragungsfunktion und Gewichtsfolge eine vergleichbare Beziehung wie im Zeitkontinuierlichen: die Übertragungsfunktion $G(z)$ ist die z -Transformierte der Gewichtsfolge $g(k)$:

$$G(z) \longleftrightarrow g(k)$$

Sowohl in $G(z)$ als auch in $g(k)$ sind alle Eigenschaften eines linearen, zeitdiskreten, dynamischen Systems enthalten. Zur Berechnung des Systemausgangs wird nur der zeitliche Verlauf des Eingang und $G(z)$ oder $g(k)$ benötigt.



$$Y(z) = G(z)U(z)$$

Multiplikation

$$y(k) = \sum_{i=0}^k g(i)u(k - i)$$

Faltung

Die Multiplikation tritt also im Bildbereich an die Stelle der Faltungssumme, genau wie sie bei zeitkontinuierlichen Systemen an die Stelle des Faltungsintegrals tritt.

3.4 Übertragungsfunktionen

Z-Übertragungsfunktion und Gewichtsfolge

Wählen wir als Eingangssignal einen Delta-Impuls $u(k) = \delta_K(k)$ bzw. $U(z) = 1$, dann ergibt sich am Ausgang die Gewichtsfolge bzw. die Übertragungsfunktion:

$$y(k) = \sum_{i=0}^k g(i)u(k-i) = \sum_{i=0}^k g(i)\delta_K(k-i) = g(k)$$

bzw.

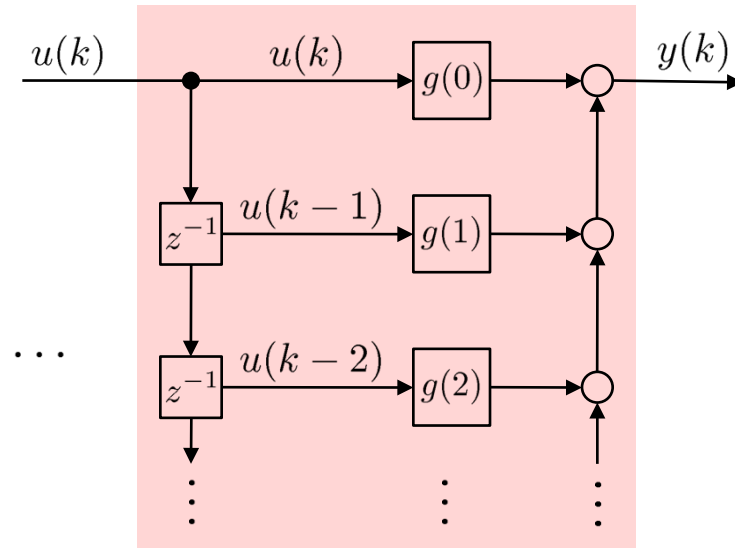
$$Y(z) = G(z)U(z) = G(z) \cdot 1 = G(z)$$

Für eine allgemeine Gewichtsfolge

$$g(k) = g(0)\delta_K(k) + g(1)\delta_K(k-1) + g(2)\delta_K(k-2) + \dots$$

ergibt sich eine entsprechende Übertragungsfunktion:

$$G(z) = g(0)z^0 + g(1)z^{-1} + g(2)z^{-2} + \dots$$

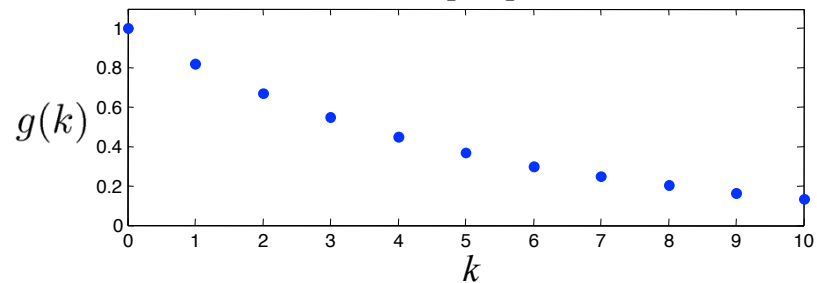
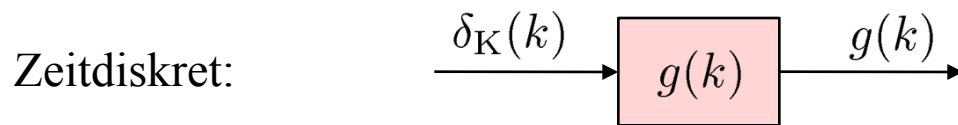
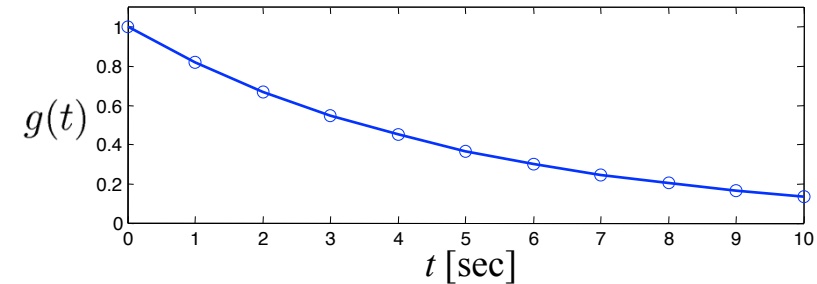
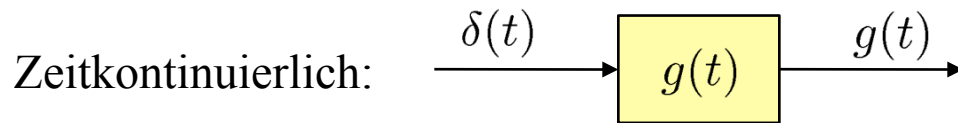


Ist die Gewichtsfolge $g(k)$ endlich lang, so gilt dies auch für die Anzahl der Terme in $G(z)$. Ist die Gewichtsfolge $g(k)$ hingegen unendlich lang, so gilt dies ebenfalls für die Anzahl der Terme in $G(z)$ und eine besser handhabbare Form muss gefunden werden, da sich mit unendlich vielen Termen in der Übertragungsfunktion schlecht rechnen lässt.

3.4 Übertragungsfunktionen

Beispiel: Transformation mittels Invarianz der Impulsantwort (Gewichtsfunktion)

Eine, insbesondere beim Filterentwurf, beliebte Methode, ein zeitkontinuierliches System in ein zeitdiskretes zu transformieren, besteht darin, identische Impulsantworten beider Systeme zu fordern. Genauer: Wir fordern, dass die abgetasteten Werte der zeitkontinuierlichen Impulsantwort identisch zur Impulsfolge des zeitdiskreten Systems sein sollen.



Für ein System PT_1 -System ergibt sich z.B. folgenden Gewichtsfunktion:

$$G(s) = \frac{K}{1 + Ts} = \frac{K/T}{s + 1/T} \rightarrow g(t) = \frac{K}{T} e^{-t/T}$$

3.4 Übertragungsfunktionen

Für $K = 5$ und $T = 5$ sec ergibt sich: $g(t) = e^{-t/5}$. Wählen wir z.B. die Abtastzeit $T_0 = 1$ sec, dann lautet die Forderung nach Invarianz der Impulsantworten (Gewichtsfunktionen):

$$g(t = kT_0) = e^{-kT_0/5} = e^{-k/5} = \left(e^{-1/5}\right)^k = 0.82^k = g(k)$$

D.h. die Gewichtsfolge des transformierten zeitdiskreten Systems ist eine *geometrische Folge*.

Die Gewichtsfolge lässt sich auch mit Hilfe zeitverschobener Delta-Impulse schreiben:

$$g(k) = 0.82^0 \delta_K(k) + 0.82^1 \delta_K(k - 1) + 0.82^2 \delta_K(k - 2) + 0.82^3 \delta_K(k - 3) + \dots$$

Dies kann nun leicht in den z -Bereich überführt werden, um die Übertragungsfunktion zu erhalten:

$$G(z) = 0.82^0 z^0 + 0.82^1 z^{-1} + 0.82^2 z^{-2} + 0.82^3 z^{-3} + \dots = \sum_{k=0}^{\infty} 0.82^k z^{-k} = \sum_{k=0}^{\infty} (0.82 z^{-1})^k$$

Da eine Übertragungsfunktion in Form einer unendlichen Reihe sehr unhandlich ist, verwenden wir die Summenformel für unendliche geometrische Reihen mit $x = 0.82 z^{-1}$:

$$G(z) = \frac{1}{1 - 0.82 z^{-1}} = \frac{z}{z - 0.82} \quad \text{Ver}G(z = 1) = \frac{1}{1 - 0.82} = 5.56 \neq 5$$

Dieses $G(z)$ entspricht also dem genannten $G(s)$ im Sinne einer identischen *Impulsantwort*.

3.4 Übertragungsfunktionen

Beispiel: Transformation mittels Invarianz der Sprungantwort

Eine andere, insbesondere in der Regelungstechnik, beliebte Methode, ein zeitkontinuierliches System in ein zeitdiskretes zu transformieren, besteht darin, identische Sprungantworten beider Systeme zu fordern. Diese Forderung liefert ein anderes Ergebnis als im vorherigen Beispiel. Es ist nämlich unmöglich sowohl die Impuls- als auch die Sprungantwort identisch zu machen! So liefert die Forderung nach Invarianz der *Sprungantwort* eine andere Übertragungsfunktion als im vorherigen Beispiel:

$$G(z) = \frac{0.9z^{-1}}{1 - 0.82z^{-1}} = \frac{0.9}{z - 0.82} \quad \text{Verstärkung: } G(z = 1) = \frac{0.9}{1 - 0.82} = 5$$

Bei der Wahl des Kriteriums für die Transformation vom Zeitkontinuierlichen ins Zeitdiskrete müssen wir also eine Entscheidung treffen, auf welche Eigenschaften es in der konkreten Anwendung besonders ankommt. Beispielsweise wird eine Transformation mittel Invarianz der Impulsantwort alle Frequenz gleich stark gewichten, da im verwendeten Anregungssignal $\delta(t)$ bzw. $\delta_K(k)$ alle Frequenzen gleich stark vertreten sind. Hingegen wird die Forderung nach Invarianz der Sprungantwort die niedrigen Frequenzen betonen. Das erkennt man auch an obigem Ergebnis, denn $G(z)$ nach Sprungantwortinvarianz liefert die korrekte Verstärkung, $G(z)$ nach Impulsantwortinvarianz aber nicht!

3.4 Übertragungsfunktionen

Übertragungsfunktion → Differenzgleichung

Gegeben sei eine allgemeine z -Übertragungsfunktion mit Zählergrad m und Nennergrad n :

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

Der Koeffizient a_0 kann durch Kürzen immer auf 1 gebracht werden. Daraus lässt sich leicht, wie folgt, die Differenzgleichung im Zeitbereich herleiten:

$$(1 + a_1 z^{-1} + \dots + a_n z^{-n}) Y(z) = (b_0 + b_1 z^{-1} + \dots + b_m z^{-m}) U(z)$$

↓

$$y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_0 u(k) + b_1 u(k-1) + \dots + b_m u(k-m)$$

Eine Totzeit von $T_t = dT_0$ entspricht einer Rückwärtsverschiebung um d Schritte:

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

$$\rightarrow y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_0 u(k-d) + b_1 u(k-1-d) + \dots + b_m u(k-m-d)$$

Im Unterschied zum s -Bereich stellt eine Totzeit also im z -Bereich kein „Fremdkörper“ dar; die Übertragungsfunktion bleibt gebrochen rational, d.h. ein Bruch zweier Polynome.

3.4 Übertragungsfunktionen

Liegt die Übertragungsfunktion in Form positiver Potenzen von z vor sollte dies zunächst in Potenzen von z^{-1} umgewandelt werden; dann kann das Standardvorgehen angewandt werden.

$$G(z) = \frac{b'_m z^m + \dots + b'_1 z^1 + b'_0}{a'_n z^n + \dots + a'_1 z^1 + a'_0}$$

ACHTUNG: Die n und m hier unterscheiden sich von denen auf der vorherigen Folie!

Erweitern mit z^{-n} (Annahme $n \geq m$) und Kürzen durch a'_n führt auf:

$$\begin{aligned} G(z) &= \frac{\frac{b'_m}{a'_n} z^{m-n} + \frac{b'_{m-1}}{a'_n} z^{m-n-1} + \dots + \frac{b'_0}{a'_n} z^{-n}}{1 + \frac{a'_{n-1}}{a'_n} z^{-1} + \dots + \frac{a'_0}{a'_n} z^{-n}} \\ &= \frac{b_{m-n} z^{m-n} + b_{n-m+1} z^{m-n-1} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}} \end{aligned}$$

Für $n = m$ ist diese Übertragungsfunktion mit der Standardform auf der vorherigen Folie identisch. Für $n > m$ lässt sich hier aus dem Zähler eine Totzeit ausklammern:

$$G(z) = \frac{b_{m-n} + b_{n-m+1} z^{-1} + \dots + b_n z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}} z^{m-n} = \frac{\tilde{b}_0 + \tilde{b}_1 z^{-1} + \dots + \tilde{b}_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}} z^{-d}$$

mit $d = n - m$. **Der Fall $m > n$ kann daher nicht auftreten (negative Totzeit \rightarrow akausal)!**

3.4 Übertragungsfunktionen

Kausalität und Realisierbarkeit

Für Übertragungsfunktion der Form

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

sind Zählerordnung m und Nennerordnung n beliebige positive Zahlen. $G(z)$ ist immer kausal.

Für Übertragungsfunktion der Form

$$G(z) = \frac{b'_m z^m + \dots + b'_1 z^1 + b'_0}{a'_n z^n + \dots + a'_1 z^1 + a'_0}$$

muss gelten: **Nennerordnung \geq Zählerordnung** bzw.: $n \geq m$. Ist diese Bedingung erfüllt, dann ist **$G(z)$ kausal**. Gilt hingegen $m > n$, dann ist $G(z)$ akausal, weil eine negative Totzeit entsteht, d.h. Werte aus der Zukunft prädiziert werden müssen.

Die Bedingung „Nennerordnung \geq Zählerordnung“ ist aus dem s -Bereich her bekannt. Dort war dies allerdings eine Bedingung für die praktische Realisierbarkeit einer Übertragungsfunktion (keine reinen D-Anteile). Damit gibt es im Zeitdiskreten keine zusätzlichen Schwierigkeiten. **Im z -Bereich ist jedes kausale System auch automatisch realisierbar!**

3.4 Übertragungsfunktionen

Sprungfähigkeit

Im Zeitkontinuierlichen kann man die Sprungfähigkeit eines Systems direkt an der s -Übertragungsfunktion ablesen: Gilt Zählergrad = Nennergrad, dann ist das System sprungfähig. Gilt Zählergrad < Nennergrad, dann ist das System nicht sprungfähig.

Im Zeitdiskreten ist ein System *sprungfähig*, wenn

- für die Übertragungsfunktion in z -Form (nur positive z -Potenzen) gilt:

$$\text{Zählergrad } m = \text{Nennergrad } n$$

- für die Übertragungsfunktion in z^{-1} -Form (nur negative z -Potenzen) gilt: $b_0 \neq 0$

Nur wenn b_0 existiert, beeinflusst der Eingang $u(k)$ den Ausgang $y(k)$ direkt. Ist $b_0 = 0$, dann muss eine Änderung im Eingang erst bis $u(k-1)$ oder weiter durchwandern bis sie sich im Ausgang $y(k)$ bemerkbar macht.

Terminologie: Ein System folge der Differenzengleichung $y(k) = b_1 u(k-1) - a_1 y(k-1)$. Je nach Sichtweise kann man es als nicht sprungfähig oder als totzeitbehaftet interpretieren:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_1 z^{-1}}{1 - a_1 z^{-1}} = \frac{\tilde{b}_0}{1 - a_1 z^{-1}} z^{-1} \quad \text{mit} \quad b_1 = \tilde{b}_0$$

3.4 Übertragungsfunktionen

Differenzgleichung → Übertragungsfunktion

Um eine Differenzgleichung in den z -Bereich zu transformieren, verschiebt man zunächst die Zeitvariable so, dass der „neueste“ Ausgangswert in der Form $y(k)$ auftaucht. Dann lässt sich die Gleichung mit Hilfe reiner Verzögerungsoperatoren z^{-i} transformieren.

Beispiel: $2y(k-1) + 4y(k) + 3y(k+3) - u(k) = -u(k-1)$

- 1.) Suche den „neuesten“ Ausgangswert: $y(k+3)$
- 2.) Zeitverschiebung, so dass dieser Wert auf $y(k)$ abgebildet wird: $k := k-3$
→ $2y(k-4) + 4y(k-3) + 3y(k) - u(k-3) = -u(k-4)$
- 3.) Transformation in den z -Bereich, Trennung von $Y(z)$ und $U(z)$, Division zur Übertragungsfunktion:

$$2z^{-4}Y(z) + 4z^{-3}Y(z) + 3Y(z) - z^{-3}U(z) = -z^{-4}U(z)$$

$$(2z^{-4} + 4z^{-3} + 3)Y(z) = (z^{-3} - z^{-4})U(z)$$

$$\frac{Y(z)}{U(z)} = G(z) = \frac{z^{-3} - z^{-4}}{3 + 4z^{-3} + 2z^{-4}} = \frac{\frac{1}{3}(z^{-3} - z^{-4})}{1 + \frac{4}{3}z^{-3} + \frac{2}{3}z^{-4}} = \frac{\frac{1}{3}(1 - z^{-1})}{1 + \frac{4}{3}z^{-3} + \frac{2}{3}z^{-4}} z^{-3}$$

3.4 Übertragungsfunktionen

IIR (*Infinite Impulse Response*)

Alle kontinuierlichen Impulsantworten $g(t)$ sind unendlich lang; typischerweise klingen sie exponentiell gegen 0 ab. Durch Abtastung entsteht damit ebenfalls eine unendlich lange Impulsfolge $g(k)$. Solche Systeme werden als IIR-Systeme (*infinite impulse response*) bezeichnet.

IIR-Systeme haben eine z -Übertragungsfunktion mit nichttrivialem Nenner, d.h. der Nenner ist komplexer als z^n . Eine solcher (komplexerer) Nenner führt nämlich dazu, dass $y(k)$ in mindestens zwei verschiedenen Zeitverzögerungen in der zugehörigen Differenzgleichung auftritt, die sich damit nur *rekursiv* berechnen lässt!

Beispiele:

$$G(z) = \frac{1}{1 - 0.9z^{-1}}$$

$$G(z) = \frac{z^{-1}}{1 - 0.8z^{-1}}$$

$$G(z) = \frac{z}{1 - 0.7z^{-1}} \quad (\text{akausal!})$$

$$G(z) = \frac{0.4 + 0.5z^{-1} + 0.6z^{-2} + 0.7z^{-3} + 0.8z^{-4}}{1 - 0.9z^{-1}}$$

$$G(z) = \frac{0.4 + 0.5z^{-1} + 0.6z^{-2} + 0.7z^{-3} + 0.8z^{-4}}{(1 - 0.8z^{-1})^2(2 - z^{-1} + 0.3z^{-2} + 0.5z^{-3})}$$

$$G(z) = \frac{z^2 + 0.7z + 0.4}{z^3 + 0.8z + 0.2}$$

$$G(z) = \frac{z^3 + 0.8z + 0.2}{z^2 + 0.7z + 0.4} \quad (\text{akausal!})$$

3.4 Übertragungsfunktionen

FIR (*Finite Impulse Response*)

Systeme mit endlich langer Impulsfolge $g(k)$ nennt man FIR-Systeme (*finite impulse response*). Sie existieren nur im Zeitdiskreten und haben keine (exakte) Entsprechung im Zeitkontinuierlichen. Ist die Länge eines FIR-Systems aber relativ groß, kann es ein stabiles IIR-System näherungsweise beschreiben. Für grenzstabile und instabile IIR-Systeme ist dies allerdings nicht möglich, da deren Gewichtsfolge nicht gegen 0 konvergiert.

FIR-Systeme haben eine z -Übertragungsfunktion ohne Nenner oder mit trivialem Nenner z^m . Daraus folgt, dass in der Differenzengleichung nur ein y -Term auftaucht und sich daher der Ausgang nur aus verschobenen Eingangssignalen berechnet (*feedforward*).

Beispiele:
$$G(z) = \frac{z^2 - z + 0.25}{z - 0.5} z^{-2} = \frac{(z - 0.5)^2}{z - 0.5} z^{-2} = (z - 0.5) z^{-2} = z^{-1} - 0.5 z^{-2}$$

$$G(z) = 1 - z^{-1} \qquad G(z) = \frac{z^3 + 4z^2 + 3z + 1}{z^5} = z^{-2} + 4z^{-3} + 3z^{-4} + z^{-5}$$

$$G(z) = \sum_{i=0}^{10} b_i z^{-i} \qquad G(z) = \frac{z^3 + 4z^2 + 3z + 1}{z^2} = z + 4 + 3z^{-1} + z^{-2} \quad (\text{akausal!})$$

$$G(z) = z^{-2} \qquad G(z) = z^3 \quad (\text{akausal!})$$

3.4 Übertragungsfunktionen

Pol-Nullstellen-Form einer Übertragungsfunktion

Bisher haben wir die Zähler- und Nennerpolynome einer z -Übertragungsfunktion in ausmultiplizierter Form betrachtet. Wie im s -Bereich ist hier auch eine Faktor-Schreibweise möglich und sinnvoll, da man daraus sofort die Pole und Nullstellen ablesen kann. Es ist einfacher zunächst die Übertragungsfunktion in positiven z -Potenzen zu schreiben:

$$G(z) = \frac{b'_m z^m + \dots + b'_1 z^1 + b'_0}{a'_n z^n + \dots + a'_1 z^1 + a'_0} = k \frac{(z - n_1) \cdot (z - n_2) \cdot \dots \cdot (z - n_m)}{(z - p_1) \cdot (z - p_2) \cdot \dots \cdot (z - p_n)}$$

Die Verstärkung von $G(z)$ berechnet sich gemäß dem Endwertsatz der z -Transformation, indem man $z = 1$ setzt:

Verstärkung:
$$K = \frac{b'_m + \dots + b'_1 z + b'_0}{a'_n + \dots + a'_1 z + a'_0} = k \frac{\prod_{i=1}^m (1 - n_i)}{\prod_{i=1}^n (1 - p_i)}$$

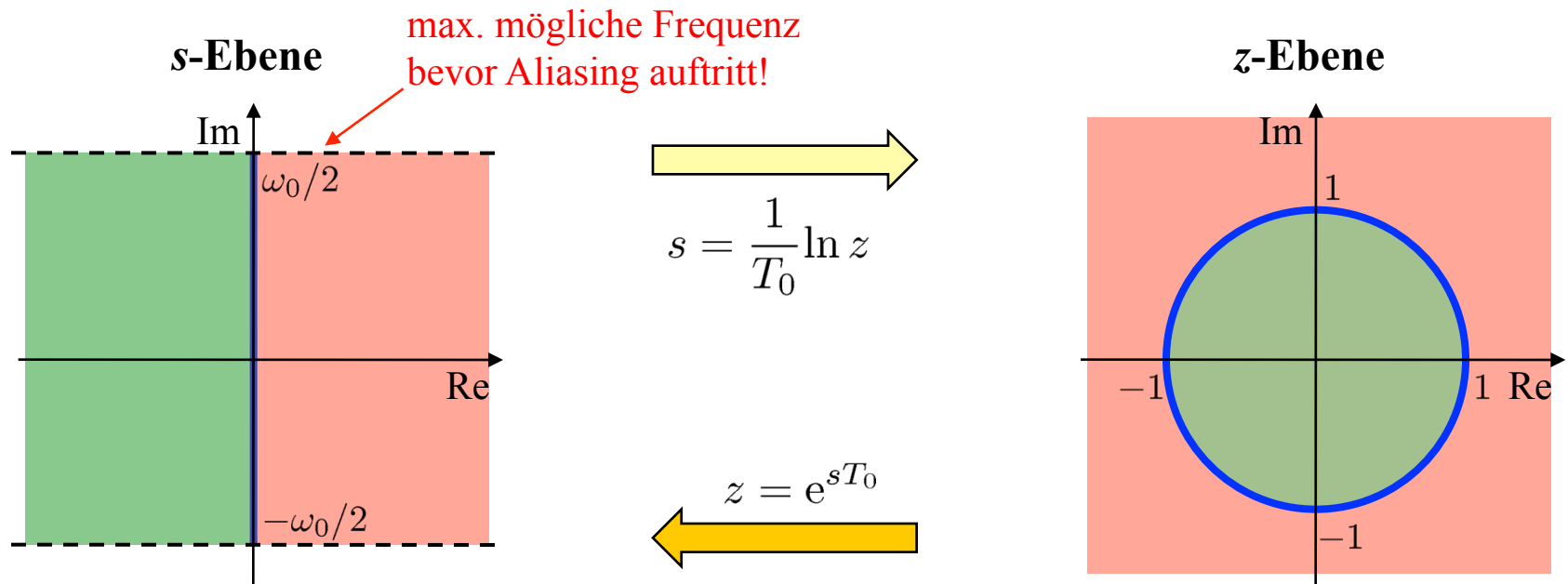
Die Pole p_i und Nullstellen n_i können mit der Formel $s = \frac{1}{T_0} \ln z$ in den s -Bereich transformiert und dort auf bekannte Weise interpretiert werden.

Damit ergeben sich sofort die entsprechenden Bedingungen für **Stabilität** und **Phasenminimalität** für Pole und Nullstellen im z -Bereich!

3.4 Übertragungsfunktionen

Beziehung zwischen s -Ebene und z -Ebene

- Das Stabilitätsgebiet „linke s -Halbebene“ wird in das „Innere des Einheitskreises“ der z -Ebene transformiert.
- Die Imaginärachse der s -Ebene wird auf den Einheitskreis der z -Ebene transformiert.
- Das Instabilitätsgebiet „rechte s -Halbebene“ wird in das „Äußere des Einheitskreises“ der z -Ebene transformiert.



3.4 Übertragungsfunktionen

Vergleich: Frequenzgang im Analogen und Digitalen

Ist eine Übertragungsfunktion $G_a(s)$ im s -Bereich gegeben, so kann sie mittels der bilinearen Transformation näherungsweise in den z -Bereich überführt werden:

$$G_a(s) = G_d \left(\frac{2}{T_0} \frac{1 - z^{-1}}{1 + z^{-1}} \right)$$

Den Frequenzgang erhält man im Analogen für $s = i\omega_a$ und im Digitalen, wenn man z auf dem Einheitskreis auswertet, also für $z = e^{i\omega_d T_0}$. Weil die bilineare Transformation nur eine Näherungsformel ist, unterscheidet sich die analoge Frequenz ω_a von der digitalen Frequenz ω_d .

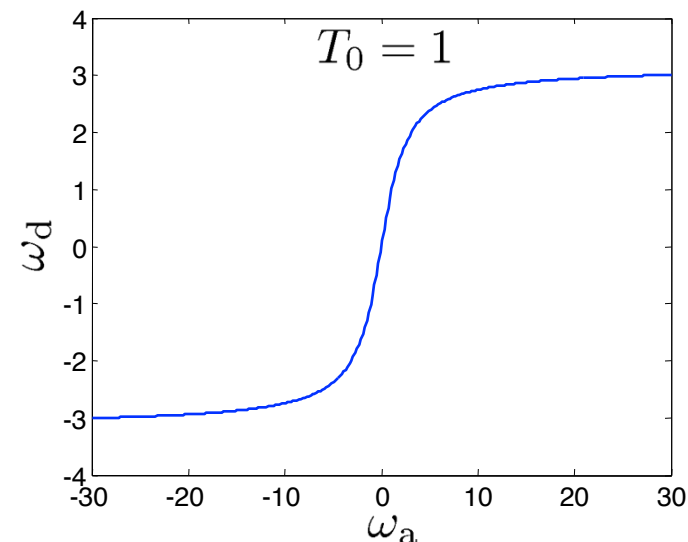
$$i\omega_a = \frac{2}{T_0} \frac{1 - e^{-i\omega_d T_0}}{1 + e^{-i\omega_d T_0}} = i \frac{2}{T_0} \tan \frac{\omega_d T_0}{2}$$

$$\omega_a = \frac{2}{T_0} \tan \frac{\omega_d T_0}{2}$$

$$\omega_d = \frac{2}{T_0} \arctan \frac{\omega_a T_0}{2}$$

Die maximal vorkommende digitale Frequenz ist:

$$\omega_{d,\max} = \frac{\pi}{T_0} = \pi f_0 = \frac{\omega_0}{2}$$



3.4 Übertragungsfunktionen

Bilineare Transformation (Tustin-Formel) = Trapez-Regel zur Integration

Im Zeitdiskreten kann die zeitkontinuierliche Integration auf verschiedene Weisen approximiert werden. Genauer als die Bildung der Untersumme oder Obersumme (siehe nächste Folie) ist die Trapez-Regel:

$$y(k) = y(k-1) + T_0 \frac{u(k) + u(k-1)}{2}$$

← **mittlere Höhe**
↑ **Breite**

Im z -Bereich ergibt sich damit:

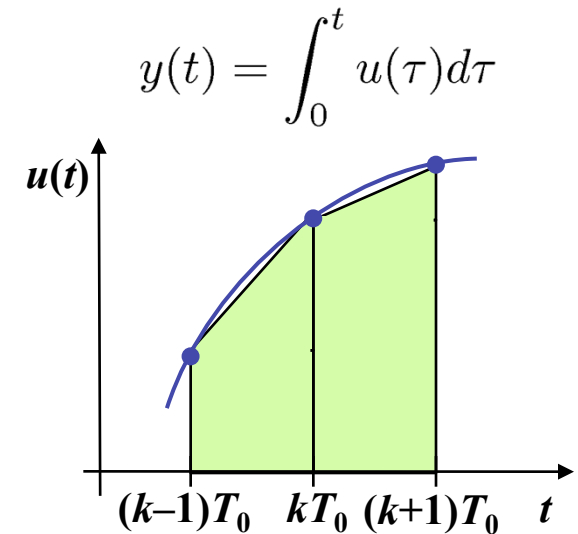
$$Y(z) = \frac{T_0}{2} \frac{1 + z^{-1}}{1 - z^{-1}} U(z)$$

Diese Formel soll im s -Bereich einer Integration entsprechen:

$$Y(s) = \frac{1}{s} U(s)$$

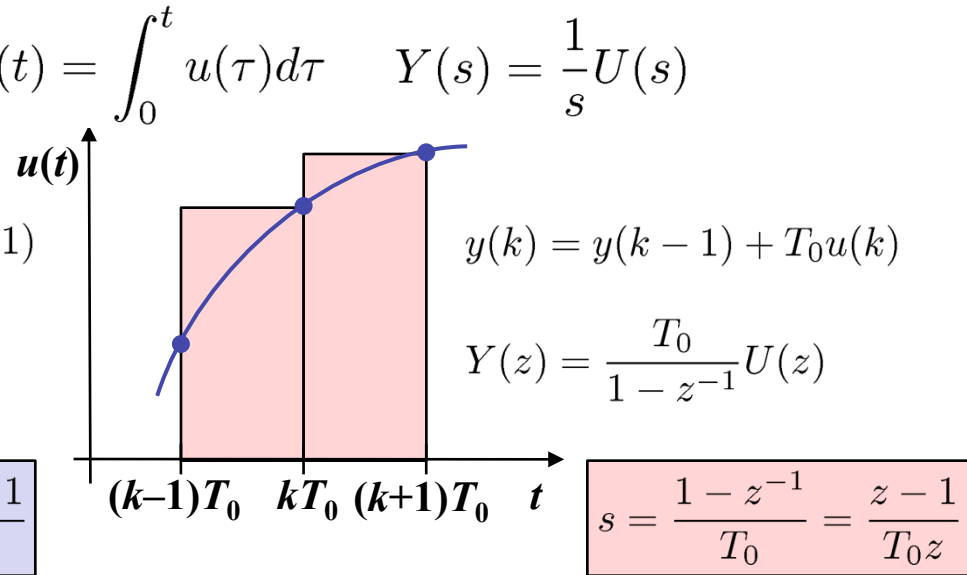
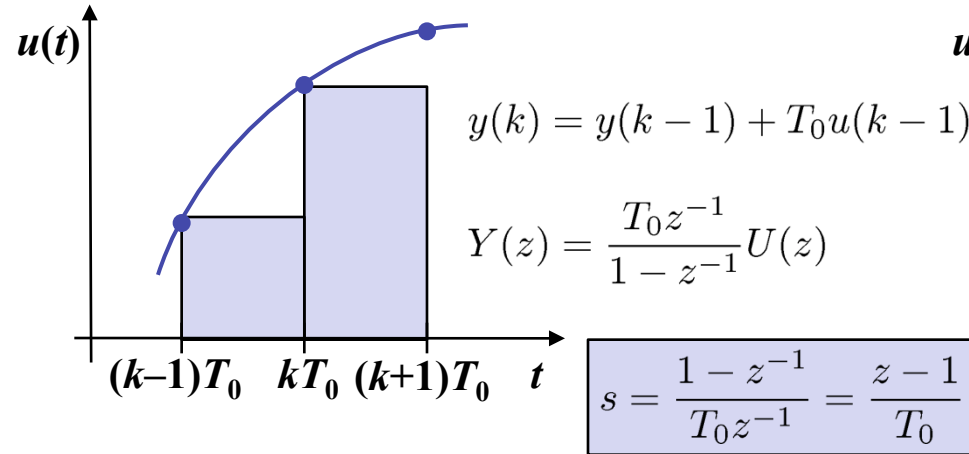
Damit erhält man genau die bilineare Transformation:

$$\frac{1}{s} = \frac{T_0}{2} \frac{1 + z^{-1}}{1 - z^{-1}} \rightarrow s = \frac{2}{T_0} \frac{1 - z^{-1}}{1 + z^{-1}}$$

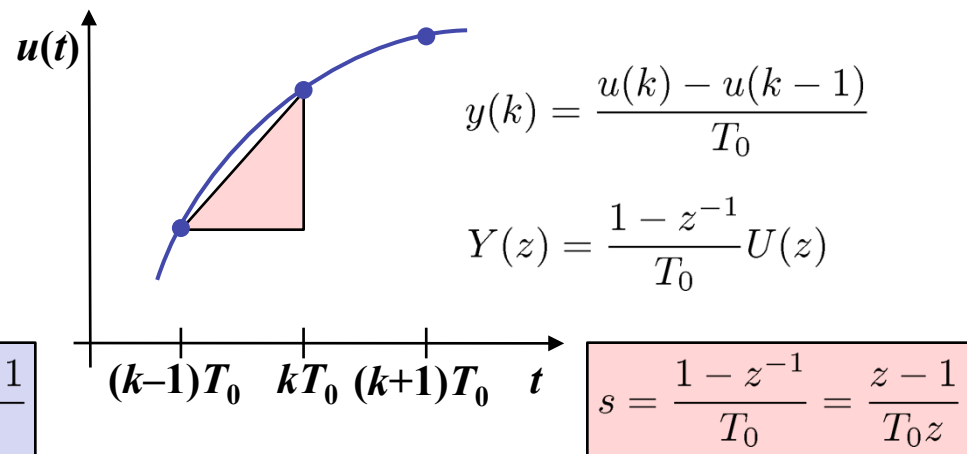
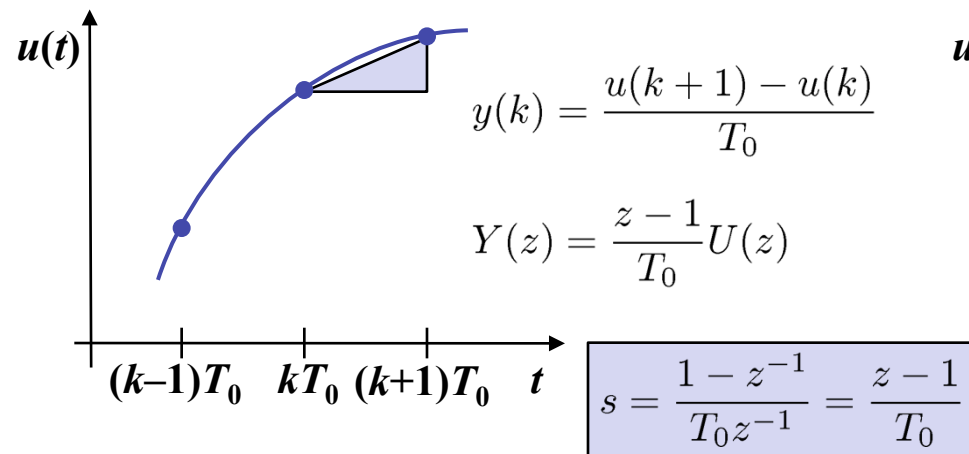


3.4 Übertragungsfunktionen

Integration mit Unter- und Obersumme $y(t) = \int_0^t u(\tau) d\tau$ $Y(s) = \frac{1}{s}U(s)$



Differentiation mit Vorwärts- und Rückwärtsdifferenzen $y(t) = \frac{d}{dt}u(t)$ $Y(s) = sU(s)$



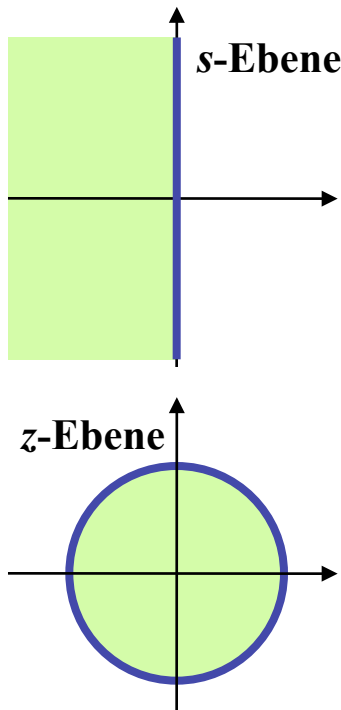
3.4 Übertragungsfunktionen

Vergleich

Bilineare Transformation (Trapez-Integration)

$$s = \frac{2}{T_0} \frac{1 - z^{-1}}{1 + z^{-1}}$$

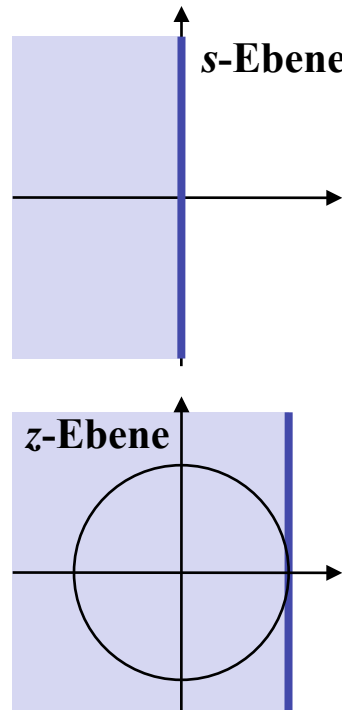
$$z = \frac{1 + \frac{T_0}{2}s}{1 - \frac{T_0}{2}s}$$



Vorwärtsdifferenzen (Untersumme -Integration)

$$s = \frac{1 - z^{-1}}{T_0 z^{-1}}$$

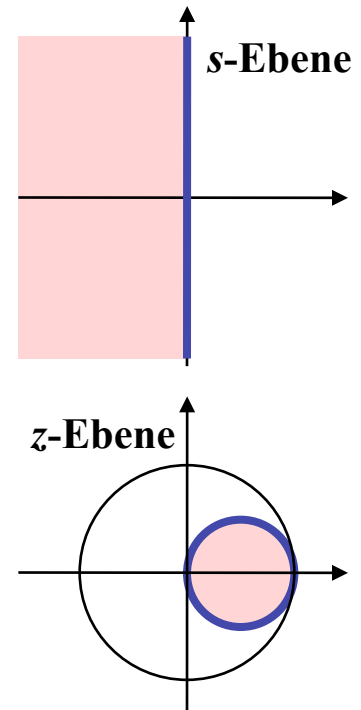
$$z = 1 + T_0 s$$



Rückwärtsdifferenzen (Obersumme-Integration)

$$s = \frac{1 - z^{-1}}{T_0}$$

$$z = \frac{1}{1 - T_0 s}$$



Stabilitätsgebiet wird zu
groß klein
abgebildet!

3.4 Übertragungsfunktionen

Stabilität

- Eine z -Übertragungsfunktion ist *stabil*, wenn alle Pole *innerhalb* des Einheitskreises liegen.
- Liegen ein oder mehrere Pole *auf dem* Einheitskreis (aber keine Mehrfachpole!) und alle restlichen Pole innerhalb des Einheitskreises, dann ist das System *grenzstabil*.
- Existiert mindestens ein Pol *außerhalb* des Einheitskreises oder ein Mehrfachpol auf dem Einheitskreis dann ist das System *instabil*.

Die Stabilitätseigenschaften einer s -Übertragungsfunktion bleiben bei der Transformation in den z -Bereich *erhalten*, weil sich die Pole entsprechend $z = e^{sT_0}$ mit transformieren.

Phasenminimalität

- Ein System ist phasenminimal, wenn es ausschließlich *stabile* und grenzstabile Pole und *Nullstellen* aufweist.
- Deshalb bleibt auch die Eigenschaft Phasenminimalität *nicht* notwendigerweise erhalten!
- Im Zeitkontinuierlichen sind nicht phasenminimale Systeme die Ausnahme, im Zeitdiskreten die Regel.

3.4 Übertragungsfunktionen

Nullstellen

- Die Nullstellen einer s -Übertragungsfunktion *ändern* sich üblicherweise bei einer Transformation in den z -Bereich.
- Die Nullstellen hängen von der Transformationsmethode ab. `c2d` in Matlab bietet verschiedene Transformationsmethode
 - Zero order hold (Invarianz der Sprungantwort): Standard in Regelungstechnik
 - First order hold
 - Invarianz der Impulsantwort
 - Tustin oder bilineare Transformation
 - Matched: Nullstellen werden wie Polstellen mit $z = e^{sT_0}$ transformiert.

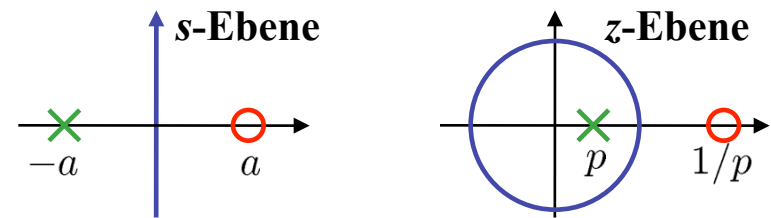
Nullstellen bei Zero Order Hold

- Die z -Übertragungsfunktion weist in der Regel Nullstellen auf, auch wenn die s -Übertragungsfunktion keine Nullstellen hat.
- Schnelle Abtastung $T_0 \rightarrow 0$ führt bei einem relativen Grad $n - m > 1$ zu instabilen Nullstellen im z -Bereich [Åström] ($n =$ Nennergrad, $m =$ Zählergrad). **Typischer Fall!**

3.4 Übertragungsfunktionen

Beispiel: Allpass im z -Bereich

Ein Allpass zeichnet sich durch einen Amplitudengang = 1 aus. Dies entsteht dadurch, dass alle Pole und Nullstellen betragsmäßig identisch sind, so dass sich ihre Wirkung Amplitudengang aufhebt. Allerdings sind alle Pole stabil und alle Nullstellen instabil (sonst würden sie sich ja gegenseitig wegkürzen). Im s -Bereich sieht z.B. ein Allpass 1. Ordnung so aus:



$$G(s) = \frac{-Ts + 1}{Ts + 1} = \frac{-s + a}{s + a} \quad \text{mit } a = \frac{1}{T} > 0$$

Pol: $s = -a$ (stabil)
Nullstelle: $s = a$ (instabil)

Bei einem Allpass im z -Bereich ist der Pol ebenfalls stabil und die Nullstelle instabil; allerdings entspricht die Nullstelle dem *inversen* Pol (nicht dem gespiegelten, wie im s -Bereich)!

$$G(z) = \frac{pz - 1}{z - p} = \frac{p - z^{-1}}{1 - pz^{-1}} \quad \text{mit } |p| < 1$$

Pol: $z = p$ (stabil)
Nullstelle: $z = 1/p$ (instabil)

Der Amplitudengang ergibt sich für $z = e^{i\omega T_0}$:

$$|G(i\omega)| = \frac{|pe^{i\omega T_0} - 1|}{|e^{i\omega T_0} - p|} = \frac{\sqrt{(p \cos \omega T_0 - 1)^2 + p^2 \sin^2 \omega T_0}}{\sqrt{(\cos \omega T_0 - p)^2 + \sin^2 \omega T_0}} = \frac{\sqrt{p^2 - 2p \cos \omega T_0 + 1}}{\sqrt{1 - 2p \cos \omega T_0 + p^2}} = 1$$



Veränderung der Abtastrate:

`decimate(x,r);1`

*% Reduziert die Abtastrate des Signals
% x um den Faktor r mit Hilfe eines Tiefpass-
% Filters.*

`upsample(x,n);1`

*% Erhöht die Abtastrate um den Faktor n, indem
% Nullen zwischen den Abtastschritten eingefügt
% werden.*

% Z.B.: x = [1 2 3];

% y = upsample(x,3);

% y =

% 1 0 0 2 0 0 3 0 0

`downsample(x,n);1`

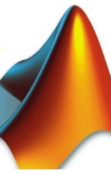
*% Herabsetzen der Abtastrate. Nur jeder n-te
% Abtastschritt wird übernommen.*

% Z.B.: x = [1 2 3 4 5 6 7 8 9 10];

% y = downsample(x,3);

% y =

% 1 4 7 10



`resample(x,p,q);1` *% Verändert die Abtastrate von Signalvektor x*
% um den rationalen Faktor p/q

Impulsantwort und Sprungantwort:

`impulse;2` *% Berechnet die Impulsantwort eines linearen*
% Systems

`step;2` *% Berechnet die Sprungantwort eines linearen*
% Systems

Partialbruchzerlegung:

`[r,p,k] = residuez(b,a);1` *% Führt eine Partialbruchzerlegung*
% unter Vorgabe des Zählerpolynoms $b(z)$
% und Nennerpolynoms $a(z)$ aus. Die
% umgekehrte Operation ist ebenfalls
% möglich

¹ : *Signal Processing Toolbox*

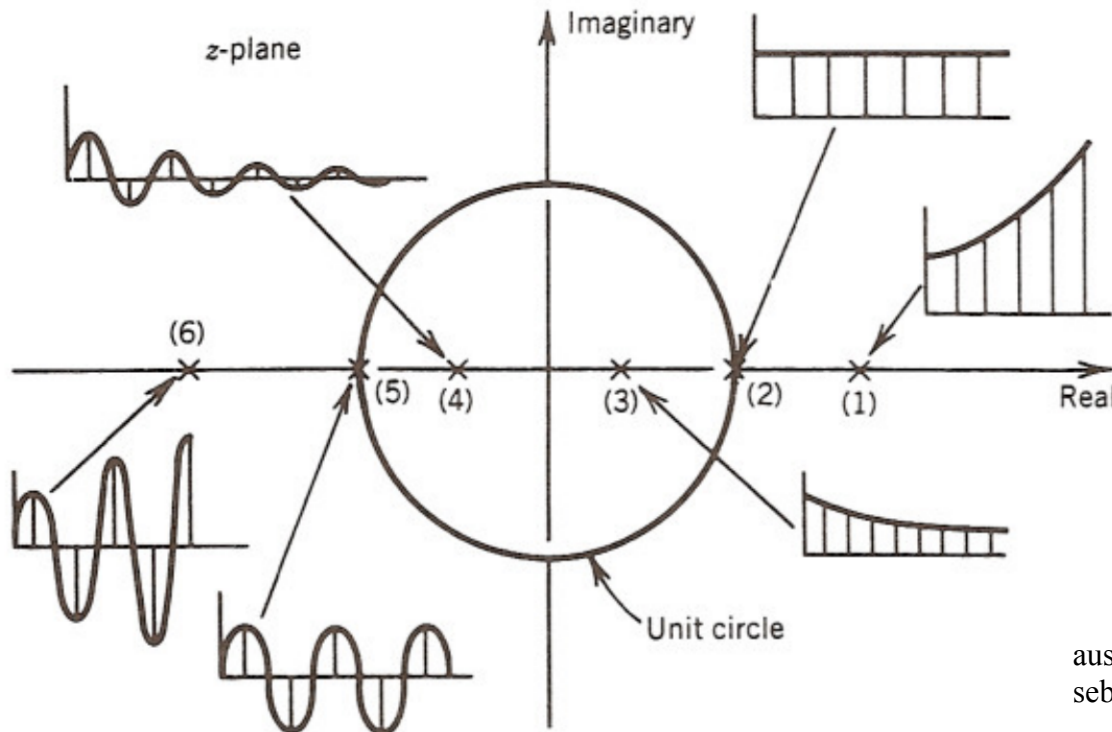
² : *Control System Toolbox*

4. Stabilität zeitdiskreter Systeme

4.1 Stabilität

Pole der z-Übertragungsfunktion

- Stabil: im Einheitskreis
- Grenzstabil: auf dem Einheitskreis
- Instabil: außerhalb des Einheitskreises



aus http://higheredbcs.wiley.com/legacy/college/seborg/0471000779/dig_control/ch24.pdf

4.1 Stabilität

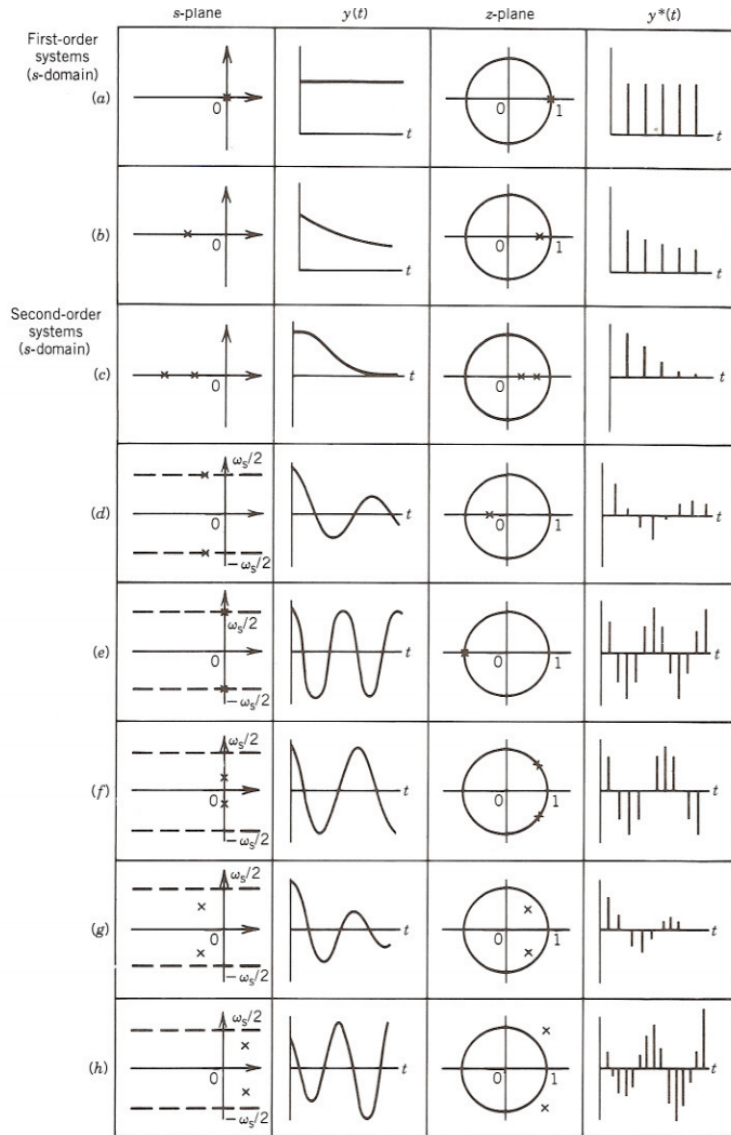


Figure 24.6 Effect of pole locations on impulse response.

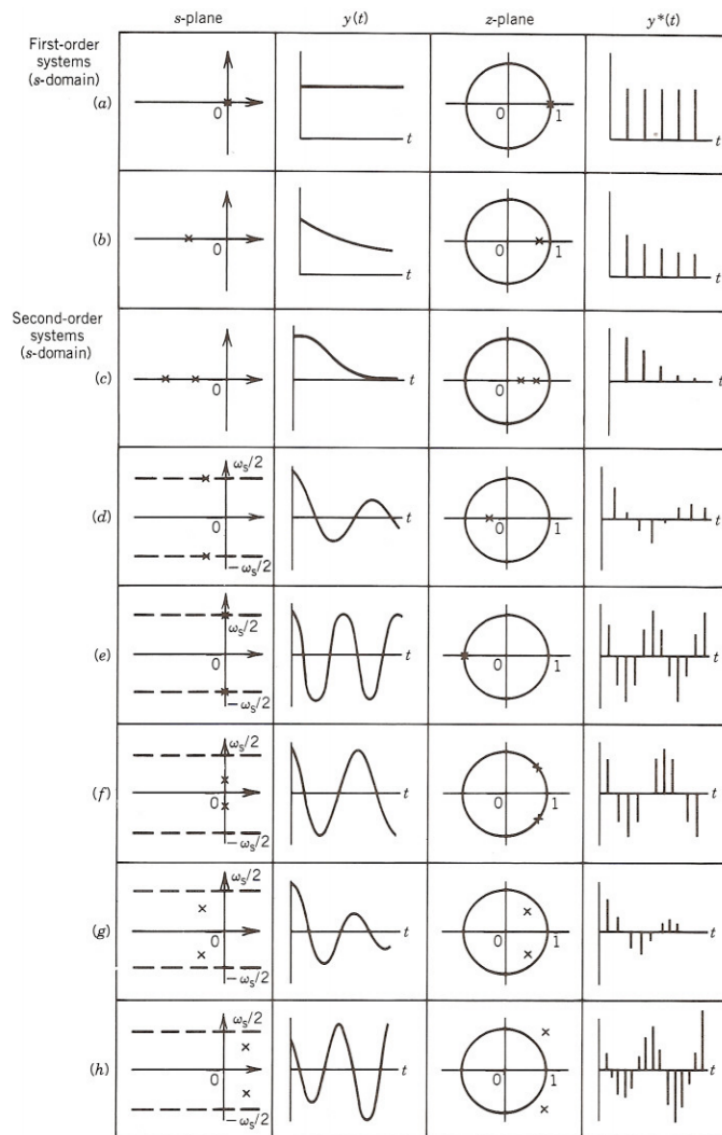


Figure 24.6 Effect of pole locations on impulse response.

aus http://higheredbcs.wiley.com/legacy/college/seborg/0471000779/dig_control/ch24.pdf

4.2 Schur-Cohn-Jury-Kriterium

Ähnlich dem Hurwitz-Kriterium für den zeitdiskreten Fall

Charakteristisches Polynom in z (nicht z^{-1}): $P(z) = a_n z^n + a_{n-1} z^{n-1} + \dots + a_1 z + a_0$
 mit $a_n > 0$ (kann mittels Multiplikation mit -1 immer erreicht werden). Meist dividiert man durch a_n , so dass $a_n = 1$. Alle Nullstellen von $P(z)$ sind im Einheitskreis, wenn

$P(z = 1) > 0$ und folgende n Bedingungen gelten:

$P(z = -1) > 0$ für n gerade bzw. $P(z = -1) < 0$ für n ungerade

$$|a_0| < a_n$$

$$|b_0| < b_{n-1}$$

$$|c_0| < c_{n-2}$$

$$|d_0| < d_{n-3}$$

⋮

$$|q_0| < q_2$$

Tabelle 11.6-2: Stabilitätstest nach JURY

Zeile	z^0	z^1	z^2	$\dots z^{n-k} \dots$	z^{n-2}	z^{n-1}	z^n
1	a_0	a_1	a_2	$\dots a_{n-k} \dots$	a_{n-2}	a_{n-1}	a_n
2	a_n	a_{n-1}	a_{n-2}	$\dots a_k \dots$	a_2	a_1	a_0
3	b_0	b_1	b_2	$\dots b_{n-1-k} \dots$	b_{n-2}	b_{n-1}	—
4	b_{n-1}	b_{n-2}	b_{n-3}	$\dots b_k \dots$	b_1	b_0	—
5	c_0	c_1	c_2	$\dots c_{n-2-k} \dots$	c_{n-2}	—	—
6	c_{n-2}	c_{n-3}	c_{n-4}	$\dots c_k \dots$	c_0	—	—
⋮	⋮	⋮	⋮	⋮	—	—	—
$2n - 5$	p_0	p_1	p_2	p_3	—	—	—
$2n - 4$	p_3	p_2	p_1	p_0	—	—	—
$2n - 3$	q_0	q_1	q_2	—	—	—	—

Die Elemente der Tabelle sind aus den Determinanten zu bilden:

$$b_k = \begin{vmatrix} a_0 & a_{n-k} \\ a_n & a_k \end{vmatrix}, \quad c_k = \begin{vmatrix} b_0 & b_{n-1-k} \\ b_{n-1} & b_k \end{vmatrix}, \quad d_k = \begin{vmatrix} c_0 & c_{n-2-k} \\ c_{n-2} & c_k \end{vmatrix}, \quad \dots$$

$$q_0 = \begin{vmatrix} p_0 & p_3 \\ p_3 & p_0 \end{vmatrix}, \quad q_1 = \begin{vmatrix} p_0 & p_2 \\ p_3 & p_1 \end{vmatrix}, \quad q_2 = \begin{vmatrix} p_0 & p_1 \\ p_3 & p_2 \end{vmatrix}.$$

aus Lutz, Wendt: „Taschenbuch der Regelungstechnik“, Deutsch, 2012

4.2 Schur-Cohn-Jury-Kriterium

Systeme 1. Ordnung: $P(z) = z + a_0$

Stabil für $|a_0| < 1$

Pol bei $z = -a_0$ muss betragsmäßig kleiner als 1 sein.

Systeme 2. Ordnung: $P(z) = z^2 + a_1z + a_0$

Stabil für

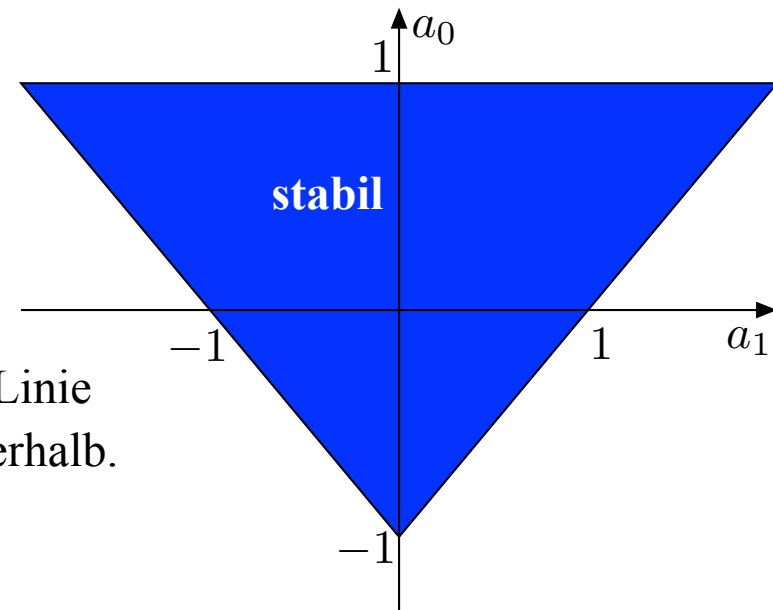
$$|a_0| < 1$$

$$P(z = 1) = 1 + a_1 + a_0 > 0 \quad \rightarrow \quad a_0 > -1 - a_1$$

$$P(z = -1) = 1 - a_1 + a_0 > 0 \quad \rightarrow \quad a_0 > -1 + a_1$$

Stabilitätsgebiet ist konvex, d.h. alle Punkte auf einer Linie zwischen zwei Punkten innerhalb liegen ebenfalls innerhalb.

Ab 3. Ordnung gilt das nicht mehr!



4.2 Schur-Cohn-Jury-Kriterium

Systeme 3. Ordnung: $P(z) = z^3 + a_2z^2 + a_1z + a_0$

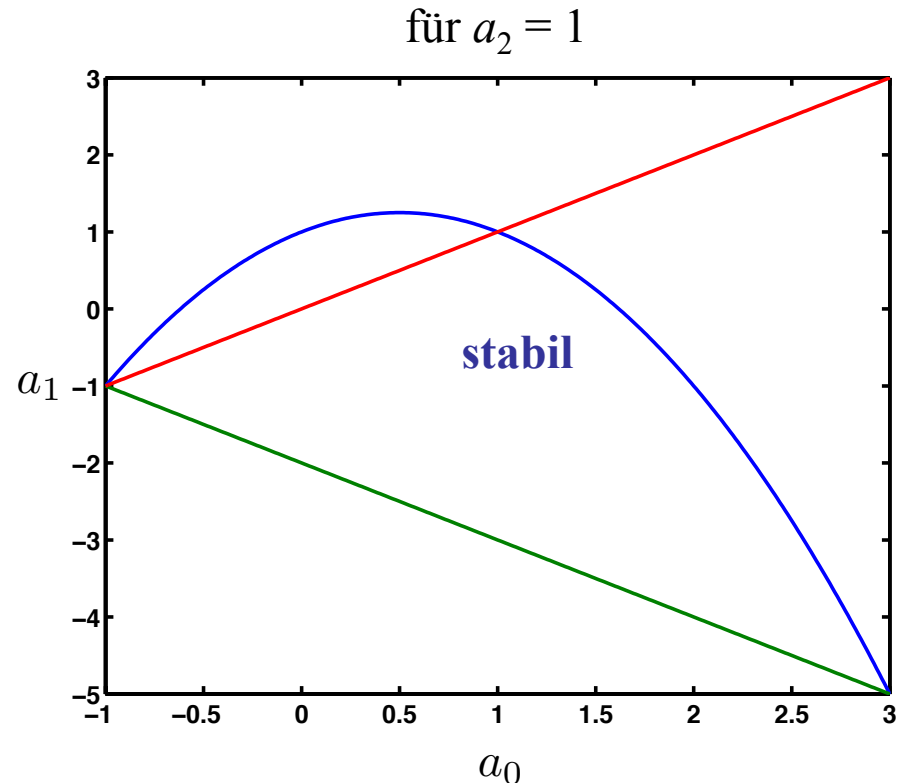
Stabil für

$$|a_0| < 1$$

$$P(z = 1) = 1 + a_2 + a_1 + a_0 > 0$$

$$P(z = -1) = -1 + a_2 - a_1 + a_0 < 0$$

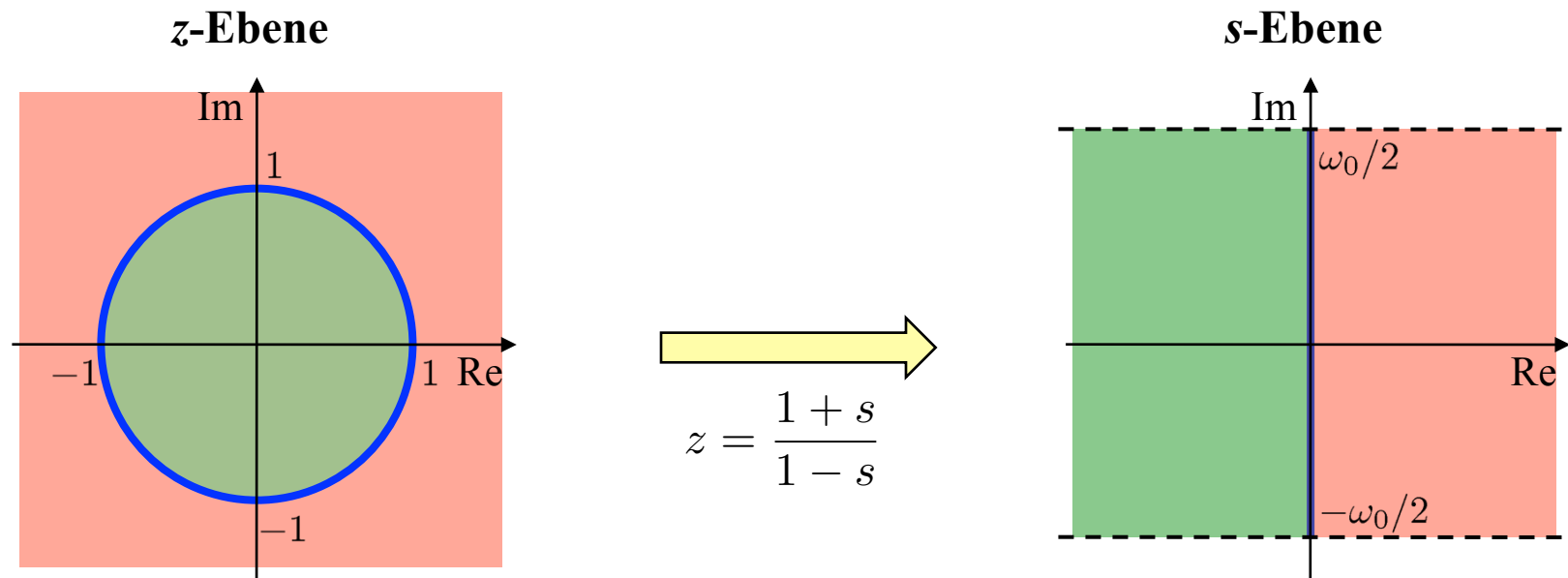
$$a_1 - a_0a_2 < 1 - a_0^2$$



4.2 Bilineare Transformation & Hurwitz

Alternative:

1. Bilineare bzw. Tustin- oder w -Transformation:
 - Nähert die exakte Transformation an: Polynom \rightarrow Polynom
 - Stabilitätsinformation bleibt erhalten
 - Gebiete im Einheitskreis \rightarrow linke s -Halbebene
2. Hurwitz-Kriterium zur Stabilitätsuntersuchung

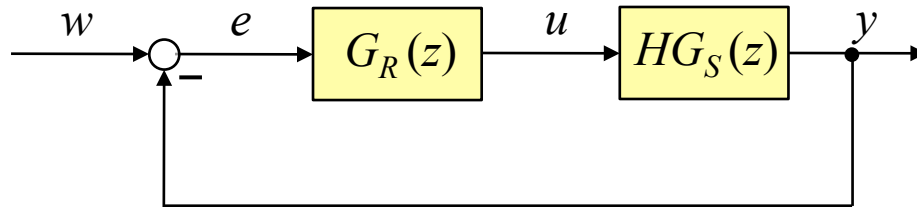


5. Deadbeat-Regler

5.1 Entwurf

Deadbeat-Regler: Regler mit endlicher Einschwingzeit

Man kann im Zeitdiskreten einen Regler entwerfen, der den Regelkreis in endlicher Zeit (FIR-Verhalten) einschwingen lässt, d.h. Stellgröße $u(k)$ und Regelgröße $y(k)$ befinden sich dann in Ruhe.



Der Entwurf basiert auf einem Kompensationsregler, d.h. es werden die Streckenpole und -nullstellen der Regelstrecke mit Halteglied $HG_S(z)$ gekürzt.

$$G_R(z) = \frac{1}{HG_S(z)} \frac{G_w(z)}{1 - G_w(z)}$$

Daher kann ein Deadbeat-Regler nur für **phasenminimale** Regelstrecken entworfen werden!

5.1 Entwurf

Man kann ein Einschwingen in n Schritten erreichen, wenn n die Ordnung der Regelstrecke ist. Für die Regelstrecke mit Halteglied wird folgende (nicht sprungfähige, d.h. $b_0 = 0$) Übertragungsfunktion angenommen:

$$HG_S(z) = \frac{b_1 z^{-1} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

Die Führungsgröße $w(k)$ wird als Einheitssprung angenommen:

$$w(z) = \frac{1}{1 + z^{-1}}$$

Die Stellgröße und die Regelgröße sollen ab dem n . Wert eingeschwungen sein:

$$u(z) = u(0) + u(1)z^{-1} + u(2)z^{-2} + \dots + u(n)z^{-n} + u(n)z^{-(n+1)} + \dots$$

$$y(z) = y(1)z^{-1} + y(2)z^{-2} + \dots + 1 \cdot z^{-n} + 1 \cdot z^{-(n+1)} + \dots$$



Da $y(n) = y(n+1) = \dots = 1$, denn die Regelgröße soll nach n Schritten der Führungsgröße entsprechen.

5.1 Entwurf

Das bedeutet wir fordern FIR-Verhalten von Führungsgröße auf Stellgröße $w(k) \rightarrow u(k)$ und von Führungsgröße auf Regelgröße $w(k) \rightarrow y(k)$. D.h.

$$\frac{u(z)}{w(z)} = Q(z) = q_0 + q_1 z^{-1} + q_2 z^{-2} + \dots + q_n y(n) z^{-n}$$

$$\frac{y(z)}{w(z)} = P(z) = p_1 z^{-1} + p_2 z^{-2} + \dots + p_n y(n) z^{-n} = G_w(z)$$

Da $w(k)$ ein Einheitssprung ist, also $w(z) = \frac{1}{1 + z^{-1}}$, folgt daraus:

$$q_0 = u(0)$$

$$q_1 = u(1) - u(0)$$

$$\vdots$$

$$q_n = u(n) - u(n - 1)$$

$$p_1 = y(1)$$

$$p_2 = y(2) - y(1)$$

$$\vdots$$

$$p_n = 1 - y(n - 1)$$

Die Übertragungsfunktionen hängen wie folgt mit der Regelstrecke zusammen:

$$\frac{y(z)}{w(z)} \frac{w(z)}{u(z)} = \frac{P(z)}{Q(z)} = \frac{y(z)}{u(z)} = HG_S(z) = \frac{b_1 z^{-1} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

5.1 Entwurf

Durch Koeffizientenvergleich ergeben sich die Reglerparameter

$$\frac{p_1 z^{-1} + \dots + p_n z^{-n}}{q_0 + q_1 z^{-1} + \dots + q_n z^{-n}} = \frac{\frac{p_1}{q_0} z^{-1} + \dots + \frac{p_n}{q_0} z^{-n}}{1 + \frac{q_1}{q_0} z^{-1} + \dots + \frac{q_n}{q_0} z^{-n}} = \frac{b_1 z^{-1} + \dots + b_n z^{-n}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}$$

für den Deadbeat-Regler:

$$G_R(z) = \frac{1}{HG_S(z)} \frac{G_w(z)}{1 - G_w(z)} = \frac{Q(z)}{P(z)} \frac{P(z)}{1 - P(z)} = \frac{Q(z)}{1 - P(z)}$$

zu

$$q_i = a_i q_0$$

$$p_i = b_i q_0$$

$$\text{mit } q_0 = \frac{1}{b_1 + b_2 + \dots + b_n}$$

da der geschlossene Regelkreis $G_w(z) = P(z)$ die Verstärkung $P(z=1) = 1$ hat, also

$$p_1 + p_2 + \dots + p_n = (b_1 + b_2 + \dots + b_n)q_0 = 1$$

5.1 Entwurf

Es ergibt sich also folgender Deadbeat-Regler:

$$G_R(z) = \frac{q_0 A(z^{-1})}{1 - q_0 B(z^{-1})}$$

$$\text{mit } q_0 = \frac{1}{b_1 + b_2 + \dots + b_n}$$

ACHTUNG: Die Polynome müssen in z^{-1} geschrieben sein!

Hier sieht man deutlich, dass das Nennerpolynom der Strecke gekürzt wird und daher stabil sein muss.

Dies führt auf folgenden geschlossenen Regelkreis:

$$G_w(z) = \frac{q_0 B(z)}{z^n}$$

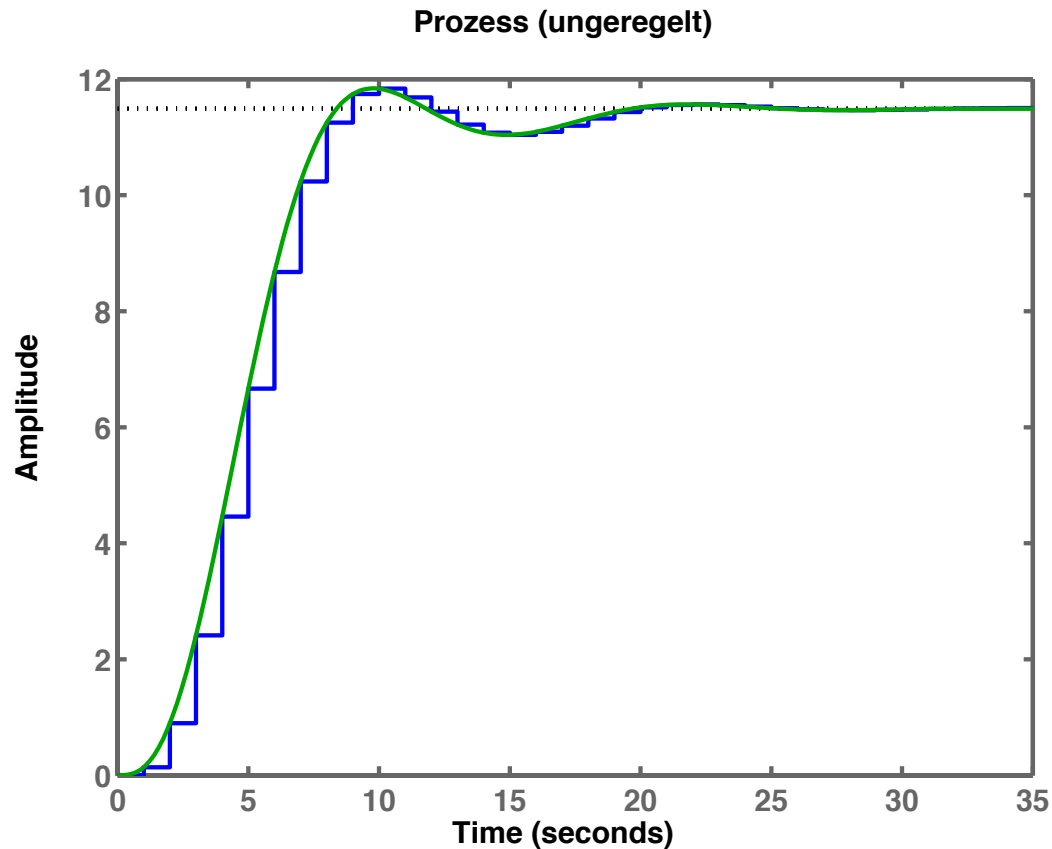
Der n -fache Pol bei 0 charakterisiert ein FIR-Verhalten n . Ordnung.

Eine Totzeit von d Abtastschritten in der Strecke kann berücksichtigt werden, indem man den Zähler $B(z^{-1})$ der Regelstrecke durch $B(z^{-1})z^{-d}$ ersetzt.

5.2 Beispiele

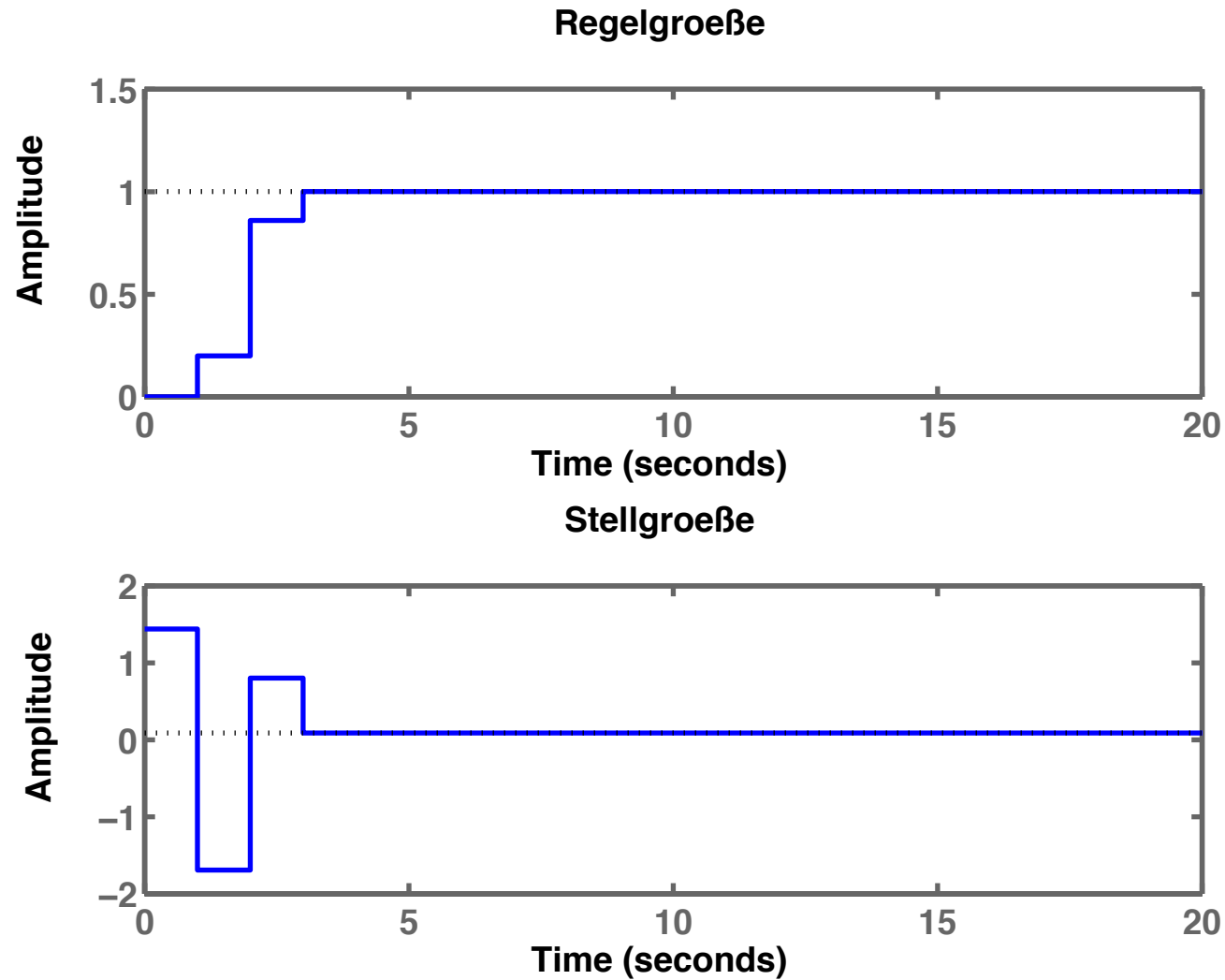
Regelstrecke 3. Ordnung: $G_S(s) = \frac{1}{(s + 0.2 + 0.5i)(s + 0.2 - 0.5i)(s + 0.3)}$

Abtastzeit: $T_0 = 1 \text{ sec}$



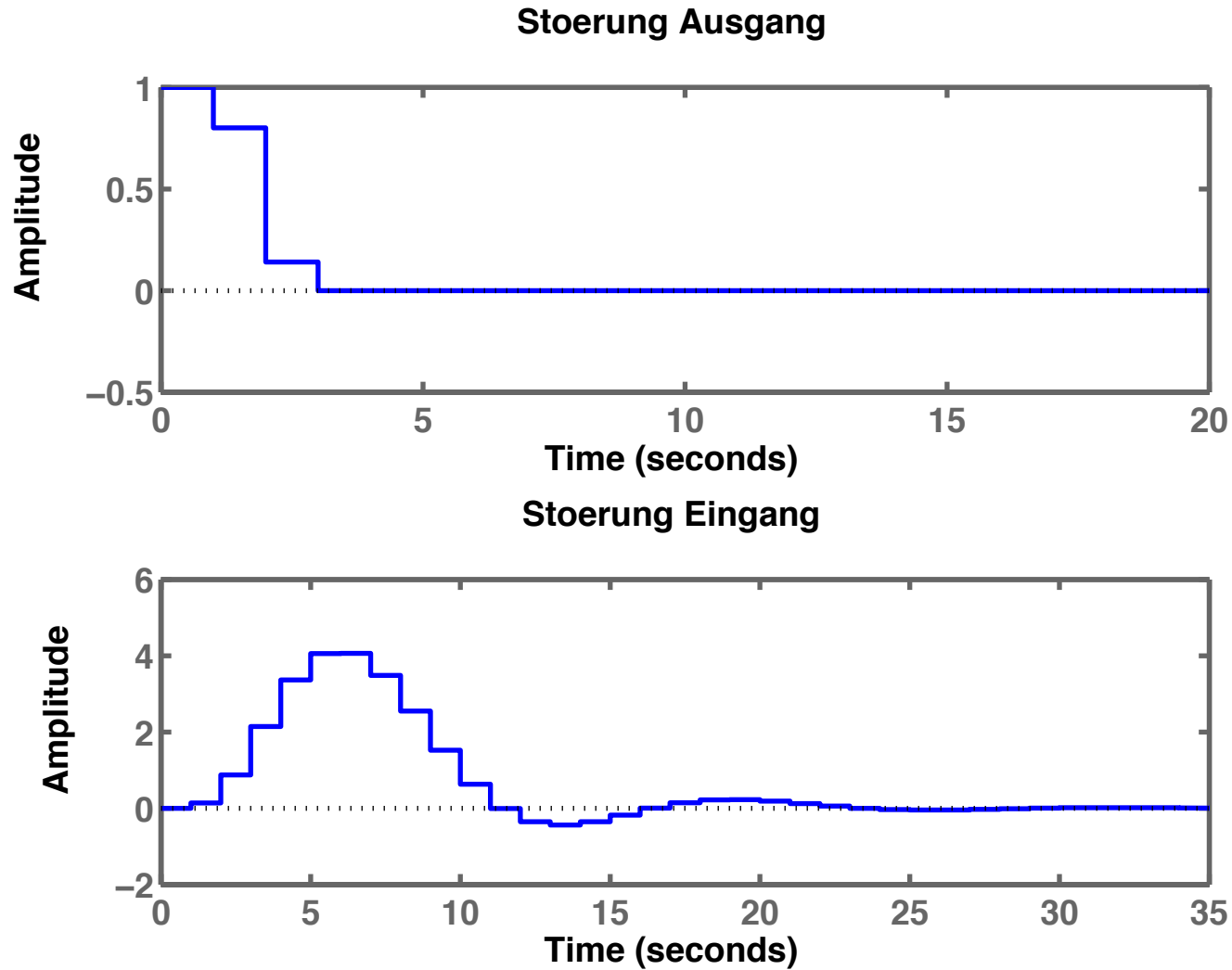
5.2 Beispiele

$$T_0 = 1 \text{ sec}$$



5.2 Beispiele

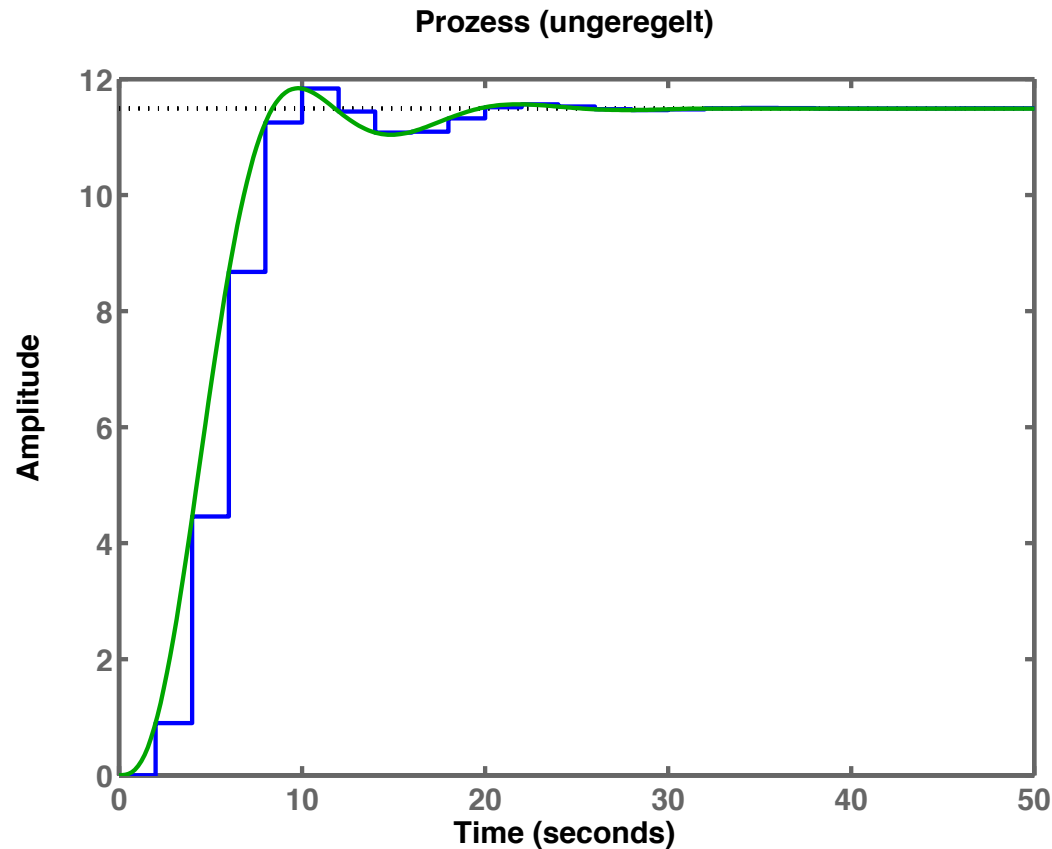
$$T_0 = 1 \text{ sec}$$



5.2 Beispiele

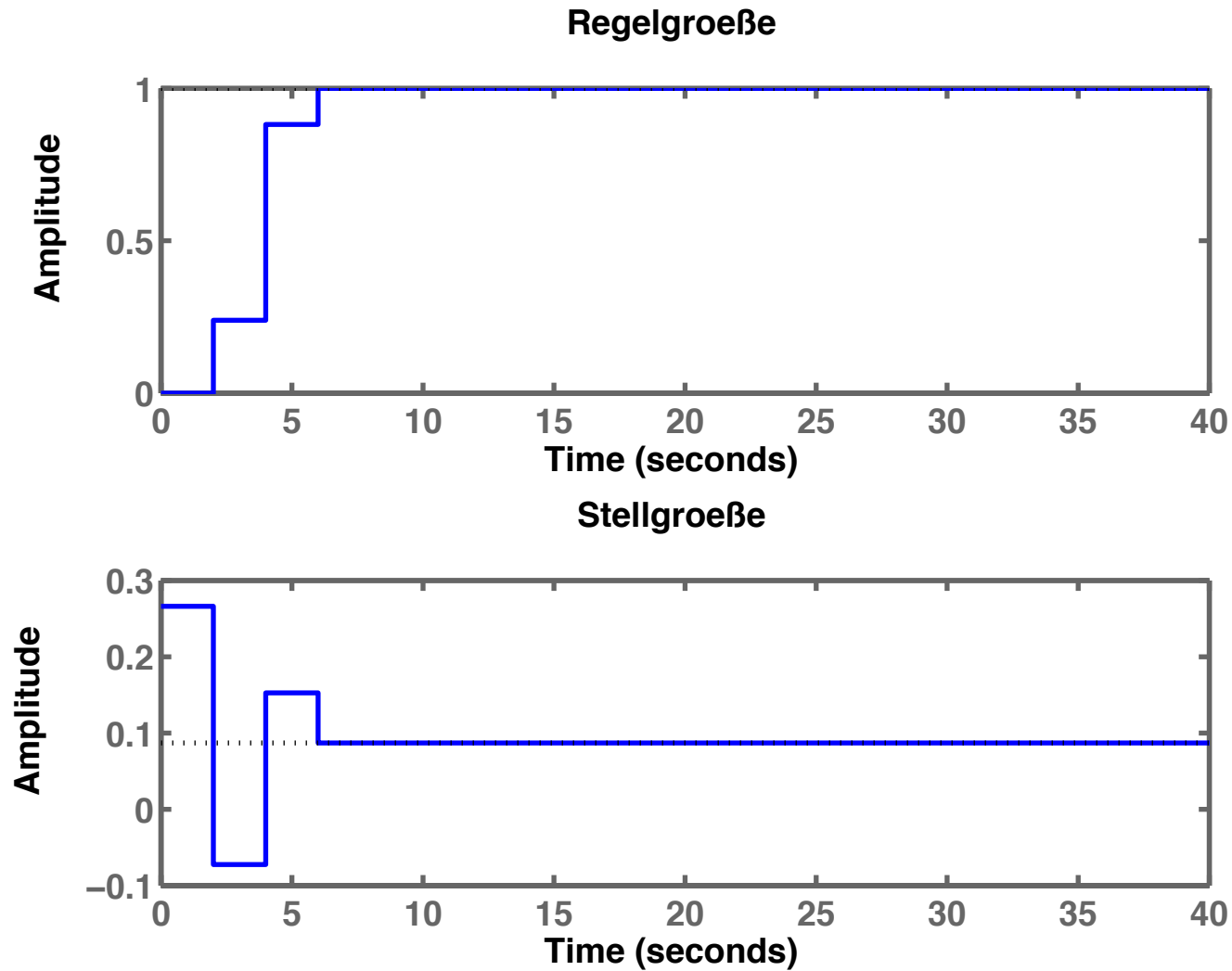
Regelstrecke 3. Ordnung: $G_S(s) = \frac{1}{(s + 0.2 + 0.5i)(s + 0.2 - 0.5i)(s + 0.3)}$

Doppelte Abtastzeit: $T_0 = 2 \text{ sec}$



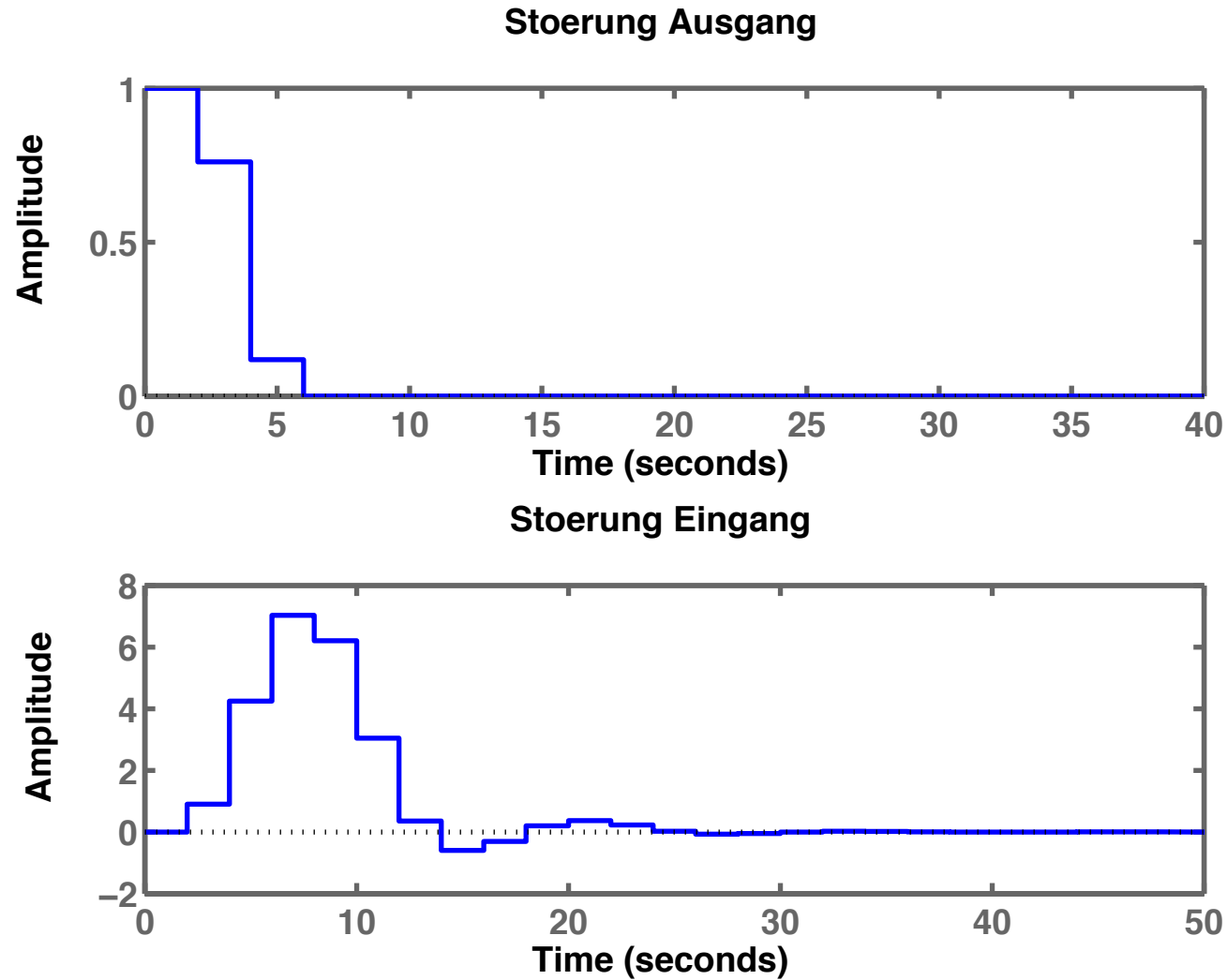
5.2 Beispiele

$$T_0 = 2 \text{ sec}$$



5.2 Beispiele

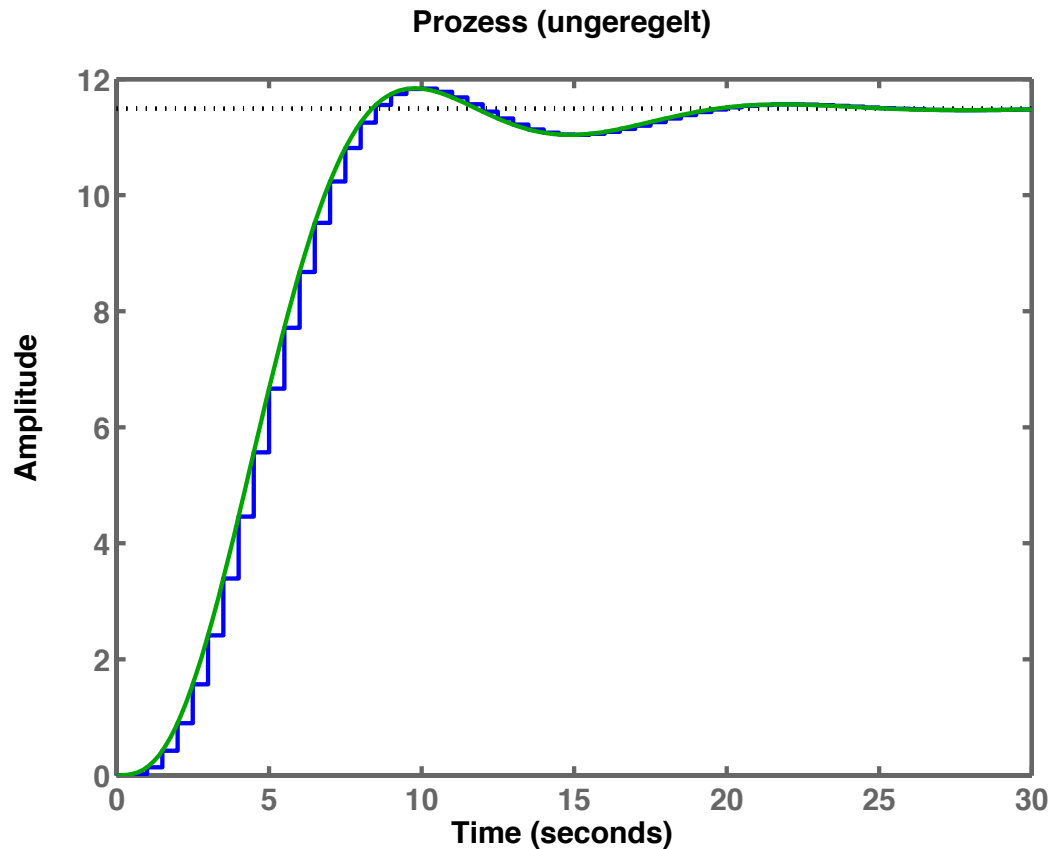
$$T_0 = 2 \text{ sec}$$



5.2 Beispiele

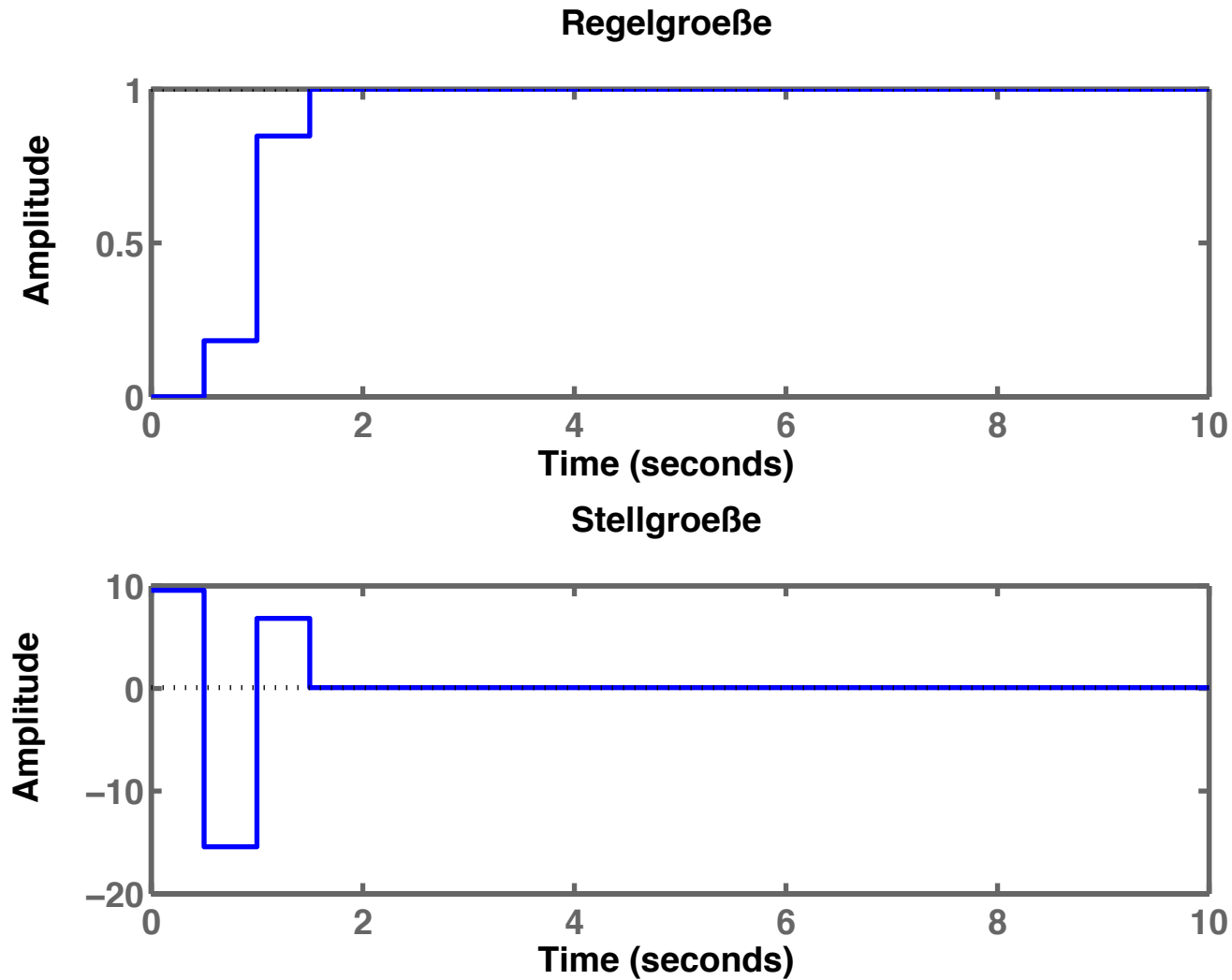
Regelstrecke 3. Ordnung: $G_S(s) = \frac{1}{(s + 0.2 + 0.5i)(s + 0.2 - 0.5i)(s + 0.3)}$

Halbe Abtastzeit: $T_0 = 0.5 \text{ sec}$



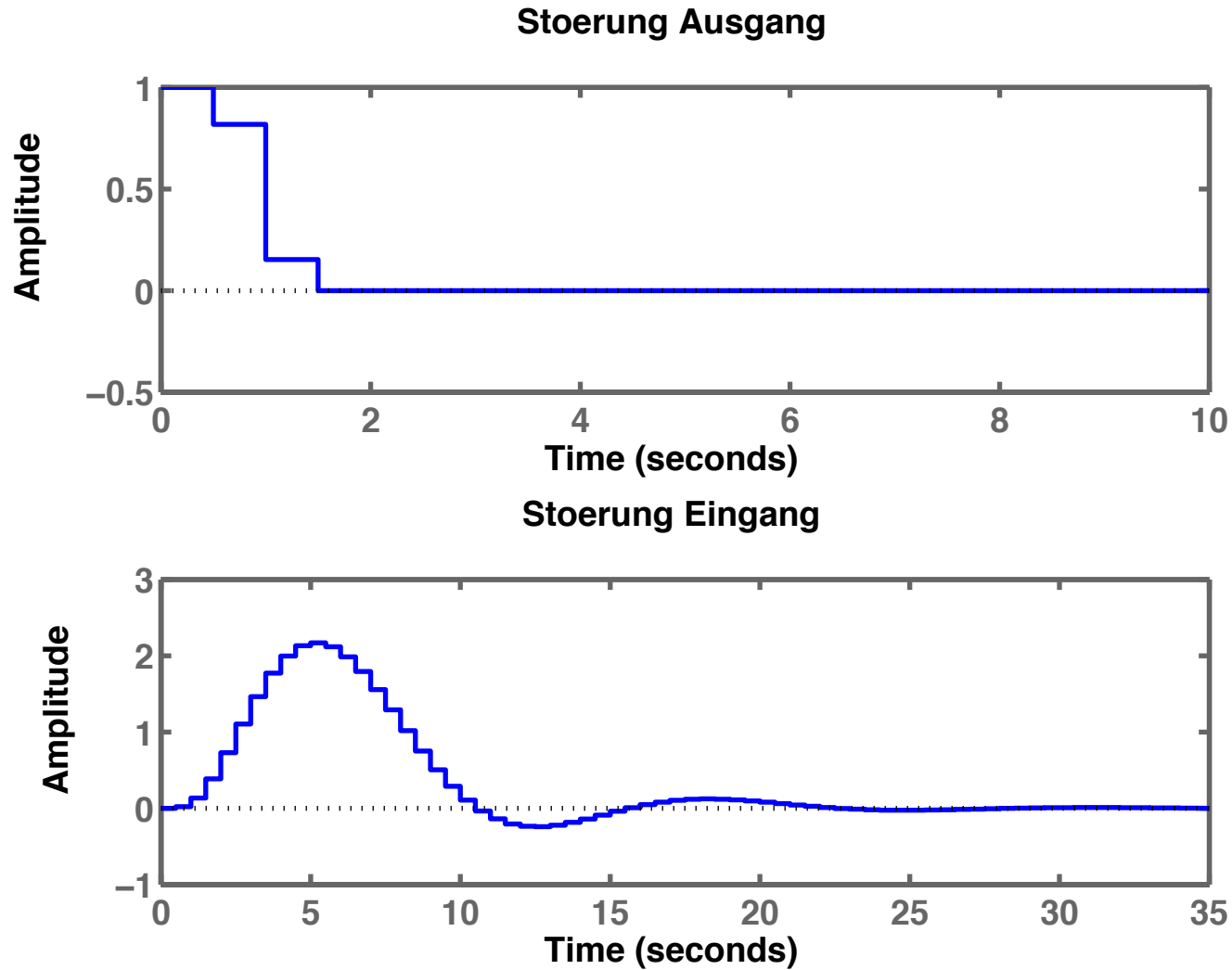
5.2 Beispiele

$$T_0 = 0.5 \text{ sec}$$



5.2 Beispiele

$$T_0 = 0.5 \text{ sec}$$



5.3 Erweiterungen

Eigenschaften des Deadbeat-Reglers

- Einfach und wenig Rechenaufwand.
- Nur für stabile und phasenminimale Regelstrecken, da es ein Kompensationsregler ist und damit die Regelstrecke invertiert werden muss. Keine instabilen Pol/Nullstellenkürzungen erlaubt!
- Hat die Regelstrecke eine Totzeit dT_0 , braucht der Deadbeat-Regler natürlich zusätzlich diese d Schritte (und dies muss beim Entwurf berücksichtigt werden).
- Da ein Deadbeat-Regler in n Schritten einschwingt, ist die endliche Einschwingzeit nT_0 , d.h. sie hängt von der Abtastzeit ab. Da diese oft sehr kurz gewählt wird, ist dies eine sehr (zu) anspruchsvolle Forderung in Bezug auf die Bandbreite des Regelkreises. Außerdem führt es auf sehr große (aggressive) Stelleingriffe.
- Daher gibt es erweiterte Deadbeat-Regler, denen ein zusätzlicher Zeitschritt gewährt wird. Dieser Freiheitsgrad kann genutzt werden, um die erste Stellgröße vorzugeben.
- Die Sprungantwort ist einem PID-Regler ähnlich. Daher kann er auch zur Initialisierung einer PID-Regler-Optimierung genutzt werden, was für eine adaptive Regelung nützlich ist.

6. Adaptive Regelung

6.1 Überblick

Was ist ein adaptive Regler?

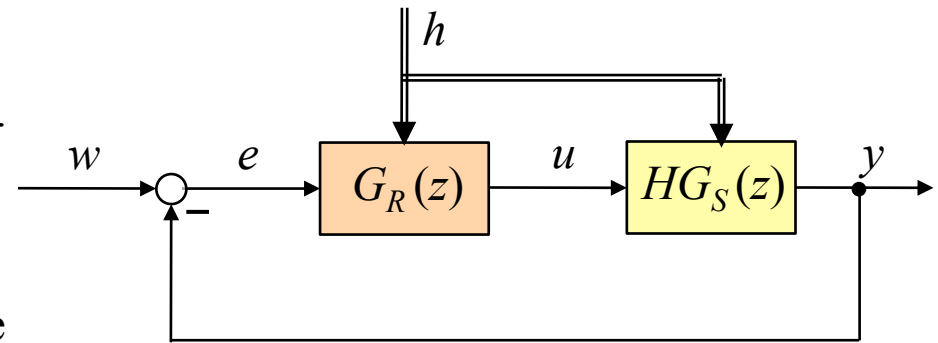
Ein Regler, der sich an Änderungen des Prozesses anpasst. Die Änderungen können sich sowohl auf die Dynamik der Regelstrecke (im einfachsten Fall: Verstärkung, Zeitkonstanten) als auf den Charakter der Störungen beziehen.

- Konstante Regler können schon sehr gut mit Prozessänderungen zurechtkommen. Dies ist ein Hauptvorteil der Rückkopplung.
- Adaptive Regler sind nur dann sinnvoll bzw. notwendig, wenn die Änderungen sehr stark sind. Dann lohnt es sich, die höhere Komplexität in Kauf zu nehmen.
- Gesteuert adaptive Regelungen sind weit verbreitet und zuverlässig einsetzbar. Komplexere Ansätze benötigen in der Regel eine aufwändige Überwachungsebene und stoßen nur in Nischen auf Akzeptanz.
- Eine Änderung des linearen Verhaltens wird oft nur durch eine Änderung des Arbeitspunktes bei *nichtlinearem* Prozessverhalten verursacht. Dann ist eine nichtlineare Regelung eine (vielleicht bessere) Alternative zu einer adaptiven Regelung.
- Echte *zeitliche* Veränderungen kann man allerdings nur mittels adaptiver Regelung in den Griff bekommen.

6.2 Gesteuert Adaptive Regelung (Gain Scheduling)

Eigenschaften

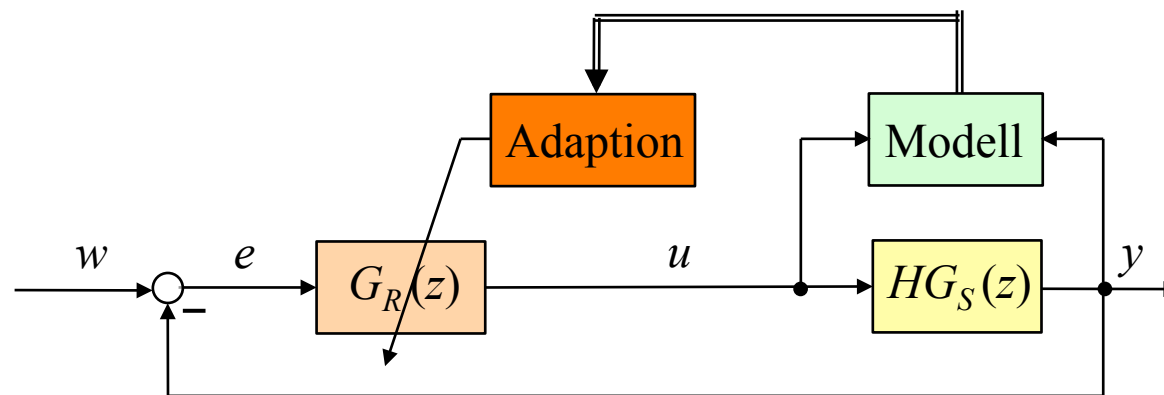
- Eine externe Größe steuert die Adaption.
- Typisches Beispiel: Die Flugdynamik eines Flugzeuges hängt stark von dessen Flughöhe ab. Man kann für verschiedene Flughöhen h ein lineares Modell aufstellen, einen linearen Regler entwerfen und in dieser Tabelle zwischen den Reglern umschalten bzw. interpolieren.
- Im einfachsten Fall unterscheiden sich nur (oder im Wesentlichen) die Verstärkungen der Modelle und in Folge der Regler. Daher der Name **Gain Scheduling**. Oft wird dieser Ausdruck auch verwendet, wenn sich die Dynamik unterscheidet. Das Prinzip bleibt das selbe.
- Die Adaption darf nur von externen Größen h abhängen und nicht von Zuständen des Systems. Die Größen h müssen messbar oder beobachtbar sein.
- Die Schedulinggröße h sollte sich nicht zu schnell ändern (was z.B. für die Flughöhe der Fall ist), weil sonst komplizierte nichtlinear dynamische Effekte auftreten können.



6.3 Self-Tuning Regulator (STR)

Eigenschaften

- Auch bezeichnet als **indirekte adaptive Regelung**.
- Ein Modell der Regelstrecke wird online identifiziert (rekursive Algorithmen).
- Dieses Modell wird im geschlossenen Regelkreis identifiziert (closed loop)
- Dieses Modell dient online dem Reglerentwurf.
- Problem: Eine gute Regelung führt auf sehr ruhiges Signalverhalten (Strich fahren). Eine gute Identifikation benötigt eine gute Anregung also unruhiges Signalverhalten (*persistent excitation*). Teilweise werden Testsignale aufgeschaltet, um gute Identifizierbarkeit sicher zu stellen.

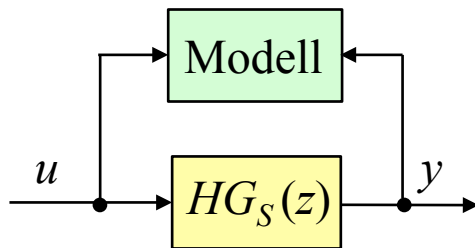


6.3 Self-Tuning Regulator (STR)

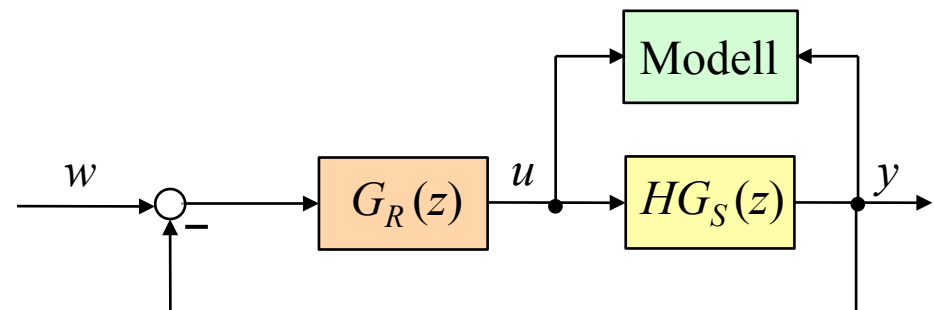
Identifikation eines Modells der Regelstrecke

- Einfacher im offenen Regelkreis, da
 - das Eingangssignal frei wählbar ist und nicht durch den Regler vorgegeben wird und
 - Störungen unkorreliert mit den Signalen sind und nicht durch die Rückkopplung mit den Signalen korrelieren.
- ABER: Man kann zeigen, dass für den Reglerentwurf am besten in geschlossenen Regelkreis identifiziert wird, da dann die dafür wichtigen Frequenzen besonders stark vertreten sind (Gewichtung mit der Empfindlichkeitsfunktion $S(s)$).
→ *iterative identification and control!*

Identifikation im offenen Regelkreis (open-loop identification)



Identifikation im geschlossenen Regelkreis (closed-loop identification)



6.4 Model-Reference Adaptive System (MRAS)

Eigenschaften

- Auch bezeichnet als **direkte adaptive Regelung**.
- Ein Referenzmodell des Regelkreises (*nicht* der Regelstrecke!) wird vorgegeben.
- Der Regler wird adaptiert, um mit dem Regelkreis dieses Verhalten nachzubilden.
- Stabilitätsnachweis mittels Ljapunov-Funktion(en) relativ einfach.
- Typischerweise wird der Regler mit einem Gradienten-basierten Verfahren adaptiert.
- Problem: Das Referenzmodell kann für manche Prozesse zu anspruchsvoll (zu schnell) für andere zu unambitioniert (zu langsam) sein.

