

# Übung 4 - Implementierung

## 1 PID-Realisierung

Das Folgeverhalten eines PID-Reglers durch die Einführung von setpoint weights ( $a$ ,  $b$  und  $c$ ) verbessert werden kann.

1. **P**: Sollwertgewichtung  $a$ , oft  $0 < a < 1$ ;
2. **I**: Sollwertgewichtung  $b$ , oft  $b = 1$  (sodass  $e_\infty \rightarrow 0$ );
3. **D**: Sollwertgewichtung  $c$ , oft  $c = 0$  (kein Überschwingen).

Mathematisch ergibt sich:

$$U(s) = k_p \cdot (a \cdot R(s) - Y(s)) + \frac{k_p}{T_i \cdot s} \cdot (b \cdot R(s) - Y(s)) + k_p \cdot T_d \cdot s \cdot (c \cdot R(s) - Y(s)). \quad (1.1)$$

Die Kreisverstärkung wird jedoch durch das Einführen von diesen Parameter nicht beeinflusst, darum bleiben die Stabilitätseigenschaften erhalten.

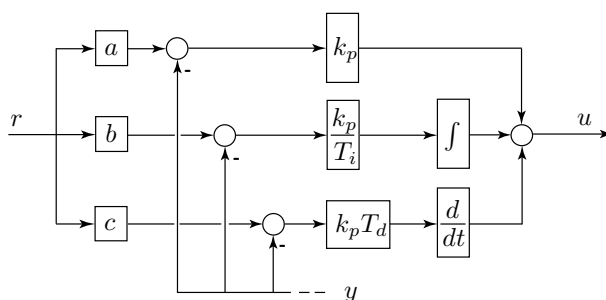


Abbildung 1: PID-Realisierung.

## 2 Feed Forward

Das Einführen eines Feed-Forward erhöht die Geschwindigkeit des Systems, da  $r$  direkt einen Einfluss auf  $u$  hat. Der Feed-Forward kann ein Gain (statisch) oder auch dynamisch sein, je nach den Anforderungen des Systems. Die Vorsteuerung hat keinen Einfluss auf die Stabilitätseigenschaften des geschlossenen Regelkreises.

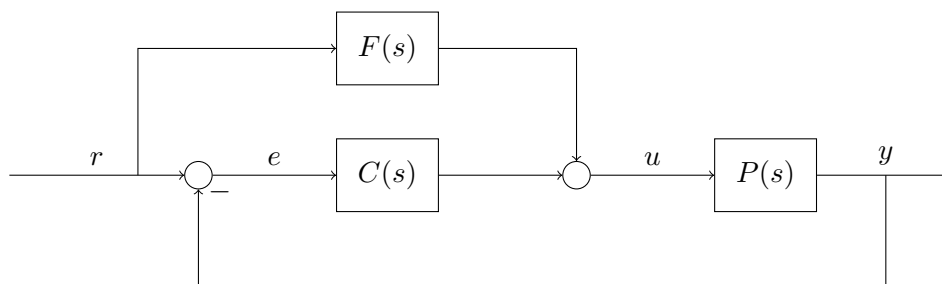


Abbildung 2: Regelkreis mit Feed Forward.

### 3 Anti-Reset Windup

Viele physikalische Signale können nicht beliebige Werte annehmen: Die sind begrenzt. Biespiele davon sind das maximale Drehmoment eines Motors ( $|T| \leq T_{\max}$ ) und Öffnungsfläche eines Ventils ( $0 \leq A \leq A_{\max}$ ). Wenn diese Begrenzungen relevant für das System sind, muss der Regelkreis entsprechend angepasst werden.

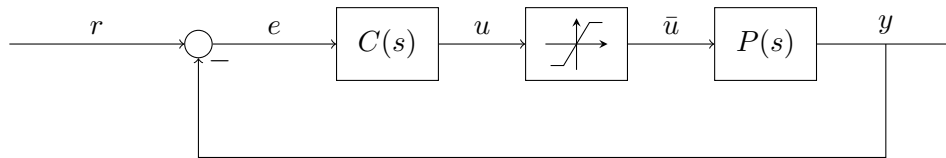


Abbildung 3: Sättigung eines Systems.

Die kritischste Situation kommt bei integrierenden Regeln vor und kann mithilfe eines Anti-Reset Windups verbessert werden. Ziel dieses Komponentes ist das ‐Leeren‐ des Integrators. Qualitativ kann man die Situation so vorstellen: Da der Fehler grösser ist, integriert der Regler weiter (aber die  $u(t)$  wird aufgrund der Sättigung nicht grösser). Um den Integrator zu leeren, d.h.

$$\int_0^t e(\tau) d\tau \approx 0, \quad (3.1)$$

braucht man dann für eine grössere Zeit einen negativen Fehler. Das ist Ursache eines grösseren (unerwünschten) Überschingers. Mit dem Anti-Reset Windup wird das Integrator (schneller) geleert, d.h. das Verhalten des Systems wird besser.

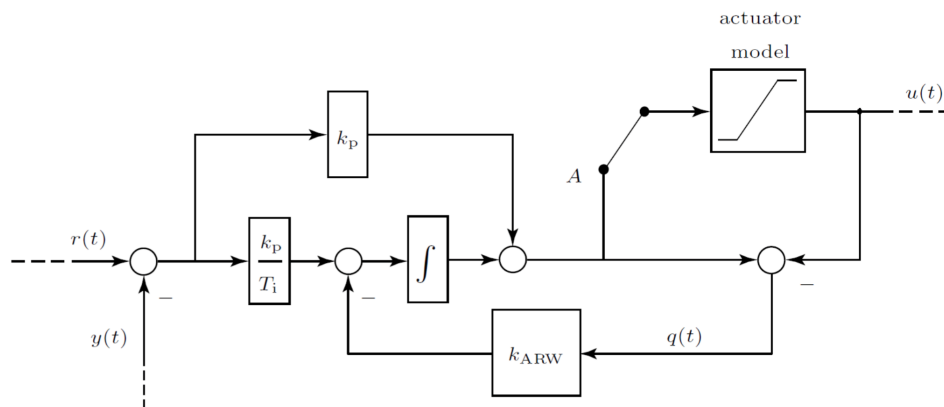


Abbildung 4: Struktur des Anti-Reset Windups.

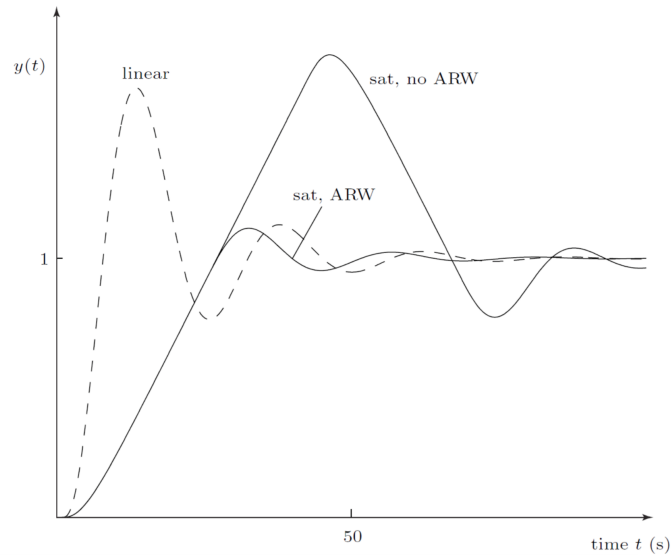


Abbildung 5: Effekte des Anti-Reset Windups.

## 4 Digitale Regelung

### 4.1 Grundlagen

Zeitkontinuierliche Signale werden durch eine Funktion  $x(t)$  dargestellt. Zeitdiskrete Signale werden durch die Folge  $x[n] = x(n \cdot T_s)$  beschrieben, wobei  $T_s$  die Abtastzeit ist. Die Abtastfrequenz  $f_s$  ist  $\frac{1}{T_s}$ .

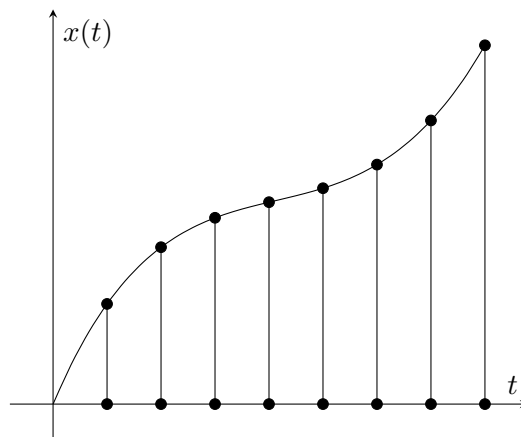


Abbildung 6: Zeitdiskrete und Zeitkontinuierliche Darstellung eines Signals.

#### 4.1.1 Vorteile der diskreten Regelung

- Einfachere Berechnungen.
- Implementierung von komplexen Algorithmen möglich.

#### 4.1.2 Nachteile der diskreten Regelung

- Es fügt eine Totzeit ins System ein ( $\approx \exp(-s \cdot T_s/2)$ ).

- Informationen zwischen  $x[n]$  und  $x[n + 1]$  sind verloren.

## 4.2 Zustandsraumdarstellung

Zeitdiskrete System werden durch folgende Zustandsraumdarstellung beschrieben:

$$x[k + 1] = A \cdot x[k] + b \cdot u[k], \quad (4.1)$$

$$y[k] = c \cdot x[k] + d \cdot u[k]. \quad (4.2)$$

### 4.2.1 Stabilität

Ein zeitdiskretes System ist asymptotisch stabil wenn alle Eigenwerte in dem Einheitskreis liegen, d.h.

$$|\lambda_i| < 1 \quad \forall i. \quad (4.3)$$

## 4.3 Aliasing

Wenn die Abtastfrequenz eines Signals zu klein ist, verliert man “zu viele” Informationen über das Signal. Ab einer gewissen Frequenz kann ein Signal dann nicht mehr eindeutig rekonstruiert werden. Nehmen wir an, das Signal

$$x_1(t) = \cos(\omega \cdot t) \quad (4.4)$$

sei gegeben. Die Diskretisierung mit Abtastzeit  $T_s$  liefert

$$x_1[n] = \cos(\omega \cdot T_s \cdot n) = \cos(\Omega \cdot n), \quad \Omega = \omega \cdot T_s. \quad (4.5)$$

Sei nun ein zweites Signal

$$x_2(t) = \cos((\omega + 2\pi/T_s) \cdot t) \quad (4.6)$$

mit Frequenz  $\omega_2 = \omega + \frac{2\pi}{T_s}$  gegeben. Die entsprechende Diskretisierung ist

$$x_2[n] = \cos((\omega + 2\pi/T_s) \cdot T_s \cdot n) = \cos(\omega \cdot T_s \cdot n + 2\pi \cdot n) = \cos(\omega \cdot T_s \cdot n) = x_1[n]. \quad (4.7)$$

Obwohl die zwei Signale nicht die gleiche Frequenz aufweisen, haben sie die gleiche diskrete Darstellung. Aus diesem Grund müssen allen Frequenzen in einem gewissen Intervall liegen, so dass dieses Phänomen, das Aliasing genannt wird, nicht vorkommt. Insbesondere muss es gelten

$$|\omega| < \frac{\pi}{T_s} \quad (4.8)$$

oder

$$f < \frac{1}{2 \cdot T_s} \quad \Leftrightarrow \quad f_s > 2 \cdot f_{\max}. \quad (4.9)$$

Die maximal erlaubte Frequenz im System  $f = \frac{1}{2 \cdot T_s}$  wird auch als Nyquist Frequenz bezeichnet. In der Praxis, benutzt man oft einen Faktor 10, d.h.

$$f < \frac{1}{10 \cdot T_s} \quad \Leftrightarrow \quad f_s > 10 \cdot f_{\max}. \quad (4.10)$$

Für Regelsysteme muss die Durchtrittsfrequenz  $\omega_c$  folgende Faustregel:

$$f_s \geq 10 \cdot \frac{\omega_c}{2\pi}. \quad (4.11)$$

Die unerwünschten Frequenz können mit einem AAF gedämpft werden.

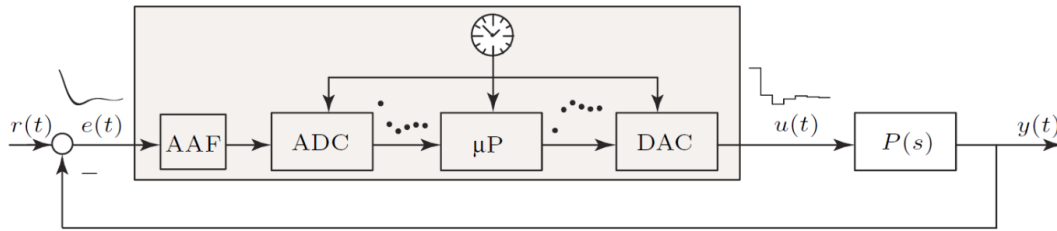


Abbildung 7: Regelkreis mit Anti-Aliasing Filter.

#### 4.3.1 Anti-Aliasing Filter

Aliasing kann durch das Zuschaltung von einem Anti-Aliasing Filter vermieden werden, um hohe Frequenzen zu eliminieren. Der resultierende Regelkreis in Abbildung 7 dargestellt.

#### 4.4 Regleremulation

Zeitdiskrete Reglern können durch Einführen der Variable  $z$  beschrieben werden. Es gilt:

$$y(t + T_s) = y[n + 1] \Leftrightarrow z \cdot Y(z) \quad (4.12)$$

$$y(t + T_s) \Leftrightarrow Y(s) \cdot e^{s \cdot T_s} \quad (4.13)$$

Daraus folgen die folgende Konversionen. In der Praxis wird am meisten die Tustin Transformation angewendet, aber es gibt Fälle, bei denen andere Transformationen bessere Verhalten aufweisen.

<b>Exakt</b>	$s = \frac{1}{T_s} \cdot \ln(z)$	$z = e^{s \cdot T_s}$
<b>Euler forward</b>	$s = \frac{z - 1}{T_s}$	$z = s \cdot T_s + 1$
<b>Euler backward</b>	$s = \frac{z - 1}{z \cdot T_s}$	$z = \frac{1}{1 - s \cdot T_s}$
<b>Tustin</b>	$s = \frac{2}{T_s} \cdot \frac{z - 1}{z + 1}$	$z = \frac{1 + s \cdot T_s/2}{1 - s \cdot T_s/2}$

Die drei Approximationen können auch mit der Taylor-Entwicklung<sup>1</sup> motiviert werden:

- Forward Euler:

$$z = e^{s \cdot T_s} \approx 1 + s \cdot T_s. \quad (4.14)$$

- Backward Euler:

$$z = e^{s \cdot T_s} = \frac{1}{e^{-s \cdot T_s}} \approx \frac{1}{1 - s \cdot T_s}. \quad (4.15)$$

- Tustin:

$$z = \frac{e^{s \cdot T_s/2}}{e^{-s \cdot T_s/2}} \approx \frac{1 + s \cdot T_s/2}{1 - s \cdot T_s/2}. \quad (4.16)$$

Die Inverse Transformationen  $s = f(z)$  können dann direkt aus diesen Gleichungen berechnet werden.

<sup>1</sup>Zur Erinnerung:  $e^x \approx 1 + x$ .

## 4.5 Stabilität

Werden die Stabilitätseigenschaften bei der Emulation verloren?

Zeitkontinuierliche Systeme sind stabil wenn alle Pole einen negativen Realteil besitzen und zeitdiskrete System sind stabil wenn alle Pole in dem Einheitskreis liegen. Bei Euler Backward and Tustin werden die Stabilitätseigenschaften gleich; bei Forward Euler ist es dagegen möglich, dass stabile System durch die Emulation instabil werden (siehe Abbildung 8).

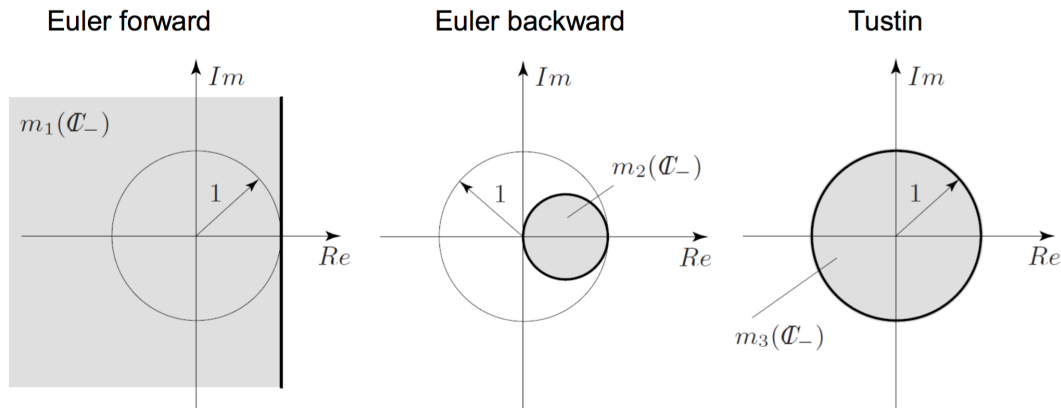


Abbildung 8: Pole mapping.

**Beispiel.** Die Stabilitätseigenschaften des Systems bleiben unverändert, falls stabile zeitkontinuierliche Pole zu stabilen zeitdiskreten Polen gemappt werden. Bei Tustin ist

$$z = \frac{1 + s \cdot T_s/2}{1 - s \cdot T_s/2} = \frac{1 + (x + i \cdot y) \cdot T_s/2}{1 - (x + i \cdot y) \cdot T_s/2} = \frac{1 + x \cdot T_s/2 + i \cdot y \cdot T_s/2}{1 - x \cdot T_s/2 - i \cdot y \cdot T_s/2}. \quad (4.17)$$

Der Betrag von  $z$  ist

$$|z| = \sqrt{\frac{(1 + x \cdot T_s/2)^2 + y^2 \cdot (T_s/2)^2}{(1 - x \cdot T_s/2)^2 + y^2 \cdot (T_s/2)^2}}. \quad (4.18)$$

Da das zeitkontinuierliche System stabil ist, muss  $x$  kleiner als 0 sein. Da  $x < 0$  gilt es

$$(1 + x \cdot T_s/2)^2 + y^2 \cdot (T_s/2)^2 < (1 - x \cdot T_s/2)^2 + y^2 \cdot (T_s/2)^2, \quad (4.19)$$

d.h.  $|z| < 1$ . Das zeitdiskrete System ist darum stabil.