

Regelungstechnik II FS 2020

PID Praxiseinsatz

Autoren: C. Küttel, Dozent: L. Guzzella, Vorlesungsnummer: 151-0591-00

Zusammenfassung Vorlesung 4

Buch Kapitel 14, Kapitel 7.5

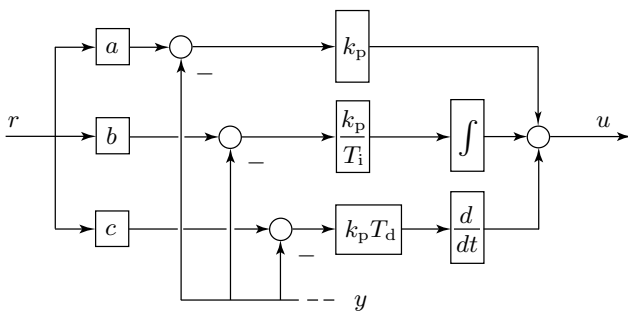
Bei Fragen: hrappael@ethz.ch, pduhr@ethz.ch, 12. März 2020

1 PID Regler in der Praxis

Der praktische Einsatz von Reglern bringt diverse bisher unerwähnte Schwierigkeiten mit sich. Die wichtigsten Einschränkungen werden im Folgenden diskutiert.

Set Point Weights

Die Standardregelstruktur dient vorallem zur Unterdrückung von Störungen. Natürlich werden Regler auch verwendet um spezifischen Referenzen $r(t)$ zu folgen. Schnelle Änderungen im Referenzsignal kann sehr hohe Eingänge $u(t)$ und ungewollte Transienten produzieren. Mit sogenannten *set point weights* (Abb. 1) kann das Regelverhalten verbessert werden.

Abb. 1: PID Regler mit Setpointverstärkungen a, b, c .

Das Referenzsignal wird für jeden Teil des Reglers (P, I, D) mit einer separaten Verstärkung (a, b, c) multipliziert. Das heisst, die Referenz wird "vormassiert", bevor sie durch den Regler geht.

- Um keinen statischen Nachlauffehler zu produzieren, falls die Referenz plötzlich konstant werden soll, wird normalerweise $b = 1$ gesetzt.
- Schnelle Referenzen erzeugen grosse Signale auf dem Differential-Pfad des Reglers ($\frac{d}{dt}(r - y) \approx \frac{d}{dt}r$). Normalerweise will man nur Änderungen aufgrund des Ausgangssignals dämpfen. Demnach setzt man oftmals $c = 0$, um hohe Ableitungen zu umgehen.
- Für die Verstärkung a auf dem proportionalen Signal kann man die Åström und Hägglund Regeln anwenden:

$$a = \alpha_{0,x} \cdot e^{\alpha_{1,x} \cdot \kappa + \alpha_{2,x} \cdot \kappa^2}$$

a	$\mu_{\min} = 0.7$			$\mu_{\min} = 0.5$		
	$\alpha_{0,x}$	$\alpha_{1,x}$	$\alpha_{2,x}$	$\alpha_{0,x}$	$\alpha_{1,x}$	$\alpha_{2,x}$
PI	1.10	-0.0061	1.8	0.48	0.40	-0.17
PID	0.58	-1.3000	3.5	0.25	0.56	-1.20

Tabelle 1: Åström-Hägglund Koeffizient für PI und PID Regler.

Typischerweise ist $0 < a < 1$. Ein gut gewählter Wert von a kann das closed-loop Regelverhalten häufig verbessern.

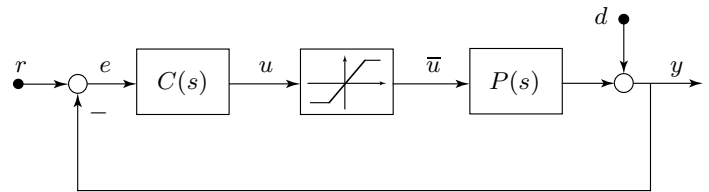
Saturationen und Anti Reset Windup (ARW)

Eine Saturation ist eine Nichtlinearität, die in jedem Regelsystem vorhanden ist. Aktuatoren können in Realität nie beliebig kleine oder beliebig grosse Sollsignale $u(t)$ umsetzen.

Die Nichtlinearität einer Saturation ist wie folgt definiert:

$$\bar{u}(t) = \begin{cases} u_{\min} & \text{if } u(t) < u_{\min} \\ u_{\max} & \text{if } u(t) > u_{\max} \\ u(t) & \text{else} \end{cases}$$

Falls der vom Regler $C(s)$ geforderte Ausgang u grösser ist als der maximal produzierbare Eingang u_{\max} des Aktuators, saturiert der Eingang bei $\bar{u} = u_{\max}$. Die gleiche Logik gilt für den minimal erreichbaren Eingang u_{\min} .

Abb. 2: Regelsystem mit Saturation auf u .

Beispiel: Das System $P(s)$ soll mit einem PI-Regler $C_{PI}(s)$ geregelt werden:

$$P(s) = \frac{1}{s(3s^2 + 4s + 1)}, \quad C_{PI}(s) = k_p \cdot \left(1 + \frac{1}{T_i \cdot s}\right)$$

Eine Simulation mit $k_p = 0.3$ und $T_i = 13$ s führt zum gestrichelten Systemverhalten:

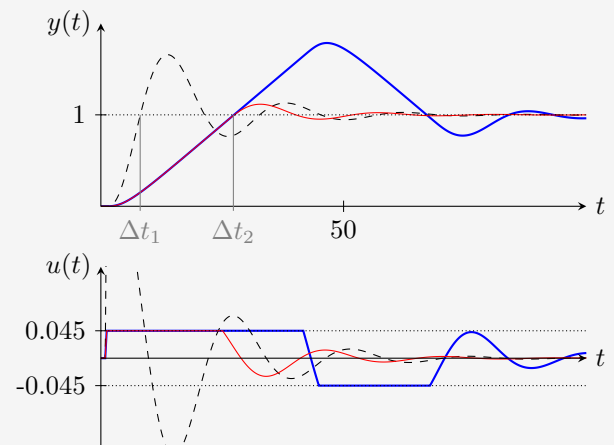


Abb. 3: Saturierendes Regelsystem.

Beim Auslegen wurde jedoch vernachlässigt, dass der Aktuator bei $|u(t)| = u_{\text{sat}} = 0.045$ saturiert. Mit der Aktuator-Sättigung resultiert effektiv die blaue Linie und damit eine schlechtere Reglerperformance. Mit einem ARW (siehe folgenden Abschnitt) resultiert die viel bessere Performance der roten Linie.

Um mit Saturationen umzugehen, muss man zuerst verstehen, was bei einem saturierten Regelsystem überhaupt passiert. Dadurch, dass das System langsamer reagiert als erwartet, kann sich der Integrator über eine längere Zeit t füllen ($\Delta t_1 < t < \Delta t_2$). Sobald sich das Vorzeichen des Fehlers ändert, überschiesst die Antwort das Ziel stärker, da sich der Integrator zuerst leeren muss.

Um die Saturation zu umgehen, kann man den Integrator durch ein *anti-reset-windup* (ARW) schwächen, sobald man erkennt, dass der Eingang saturiert (siehe Abb. 4, rote Box). In der roten Box sieht man eine mögliche Struktur eines anti reset-windup. Dabei ist $q(t) = u(t) - \bar{u}(t)$. Falls $u(t)$ nicht saturiert, ist $q(t) = 0$ und der ARW macht nichts. Falls man saturiert, wird die zu integrierende Grösse reduziert, sodass sich der Integrator in der Saturationsphase nicht zu stark füllt. Das heisst je stärker die Aktuatoren saturieren, desto mehr wirkt man der Integration entgegen. Dazu muss man die Saturationspunkte der Nichtlinearität kennen.

Bumpless Transfer

Um ein System zu testen macht es oft Sinn einen Manuellen Modus (M) und einen Automatischen Modus (A) zu haben. Wenn man von einem Modus in den anderen wechselt, will man dies so reibungslos wie möglich machen (Bumpless Transfer). Dazu verwendet man folgende Regelstruktur:

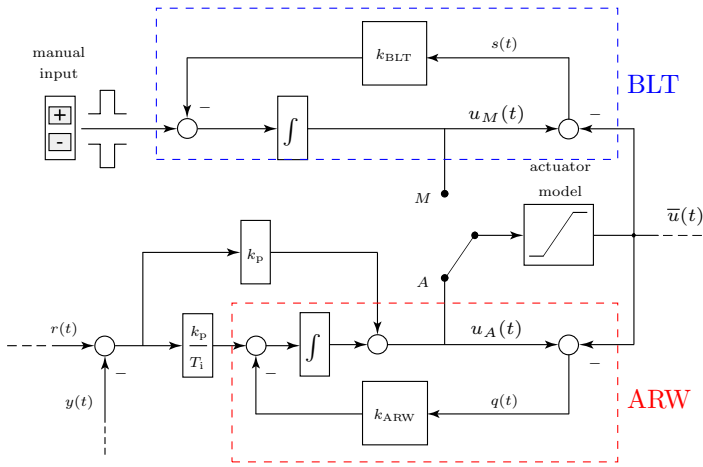


Abb. 4: Anti Reset-windup (ARW) und Bumpless Transfer (BLT) Regelung.

Während man im Modus A ist, regelt man $u_M(t)$ auf $u_A(t)$, sodass beim Wechsel von A nach M kein Sprung in der Eingangsgrösse auftritt. Umgekehrt regelt man im Modus M mit Hilfe der ARW-Rückführung das Signal $u_A(t)$ auf $u_M(t)$.

Gain Scheduling

Gain scheduling wird meist für nichtlineare Systeme verwendet, die um verschiedene Betriebspunkte linearisiert werden. Für jeden Betriebspunkt kann ein anderer linearer Regler ausgelegt werden.

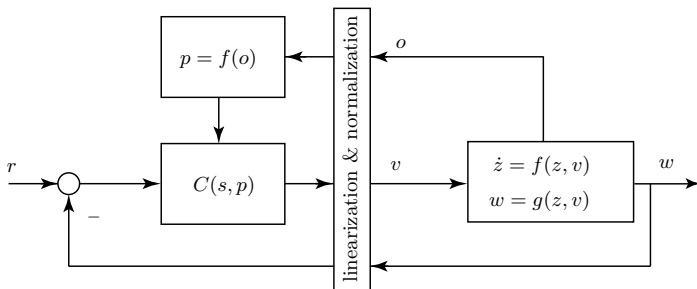


Abb. 5: Diagramm zu Gain scheduling.

Beispielsweise werden für einen PI-Regler die Verstärkungen $k_p(o)$, $T_i(o)$ zu Funktionen des Betriebspunktes o .

Analoge Realisierung

Mit analogen Schaltungen kann man eine Eingangsspannung U_e in eine Ausgangsspannung U_a umwandeln. Dabei kann man die Grundstruktur in Abb. 6 verwenden.

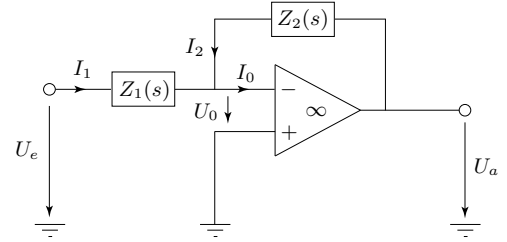


Abb. 6: Grundstruktur mit idealem analogem Opamp.

Die Übertragungsfunktion von $U_e(s)$ nach $U_a(s)$ lautet

$$\Sigma(s) = \frac{U_a(s)}{U_e(s)} = -\frac{Z_2(s)}{Z_1(s)},$$

wobei $Z_1(s)$ und $Z_2(s)$ aus Widerständen (R), Induktoren (L) und Kapazitäten (C) aufgebaut sind.

Die Impedanzen der Standard RLC Elemente lauten:

$$Z_R(s) = R, \quad Z_L(s) = sL, \quad Z_C(s) = \frac{1}{sC}.$$

Des Weiteren gelten die Kirchhoffschen Regeln für Serien- und Parallelschaltungen dieser Elemente.

Zeitdiskrete Realisierung

In diesem Kapitel wird gezeigt, wie man ein continuous-time Regler $C(s)$, in einen discrete-time Regler $C(z)$ umwandelt. Die Diskretisierung bezieht sich in dem Fall auf eine Zeitdiskretisierung. Bis jetzt wurde angenommen, dass ein Regler $C(s)$ in jedem Zeitpunkt t berechnen kann was $u(t)$ sein soll. In der Realität läuft die Rechnung in einer fixen Taktfrequenz f_s . D.h. es können in Wirklichkeit nur zu fixen Zeitpunkten t_n neue Eingänge $u(n \cdot T_s) := u[k]$ berechnet werden:

$$t_n = \frac{n}{f_s} = n \cdot T_s, \quad n \in 0, 1, \dots \quad (1)$$

Die Diskretisierung findet statt, da der Mikroprozessor (μP), der die Rechnungen ausführt, auf einer fixen Frequenz läuft (dargestellt durch die Uhr in Abb. 7).

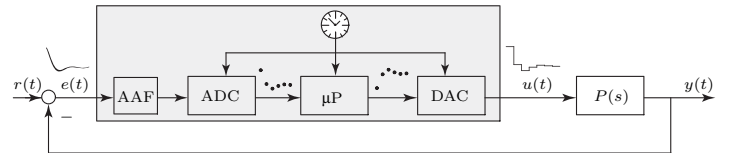


Abb. 7: Digitales Regelsystem.

- ADC : Steht für Analog-to-digital Converter. Konvertiert zeit-kontinuierliche Signale in zeit-diskrete Signale.
- μP : Steht für Mikroprozessor. Berechnet den zeitdiskreten Eingang $u[k]$.
- DAC : Steht für Digital-to-Analog Converter. Konvertiert zeit-diskrete Signale in zeit-kontinuierliche Signale.

Der DAC wird meistens mit einem *Zero-order Hold* (ZOH) Element realisiert:

$$u(t) = u[k] \quad \forall t \in [k \cdot T_s, (k+1) \cdot T_s),$$

Das heisst der Eingang $u[k]$ wird zwischen zwei Berechnungsschritten konstant gehalten.

Beispiel: Ein Signal $x(t)$ wird zuerst abgetastet, und dann wird ein zero-order hold (ZOH) angewandt.

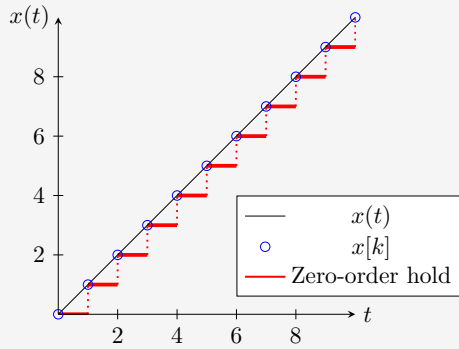


Abb. 8: Gesamples Signal in blauen Kreisen, ZOH in rot.

Das Sampling geschieht mit einem ADC. Der zero-order hold wird mit dem DAC realisiert.

In Abb. 7 ist in der grauen Box noch ein weiteres Element: der analoge Anti-aliasing Filter (AAF). Der AAF ist nötig, da ein zeitdiskretes Signal nicht beliebig hohe Frequenzen repräsentieren kann. Energien der Frequenzen im analogen Signal, die höher als die maximal repräsentierbare Frequenz sind, werden bei der Diskretisierung ungewollt auf das gesamte Frequenzspektrum verteilt und verzerren dieses somit. Deshalb wählt man für den AAF einen Tiefpassfilter (System erster Ordnung) um hohe Frequenzen vor der Diskretisierung zu eliminieren.

Beispiel: Du sollst ein diskretes Signal zeichnen, dass für eine fixe Rate $f_s = \frac{1}{T_s}$ die höchstmögliche Frequenz hat.

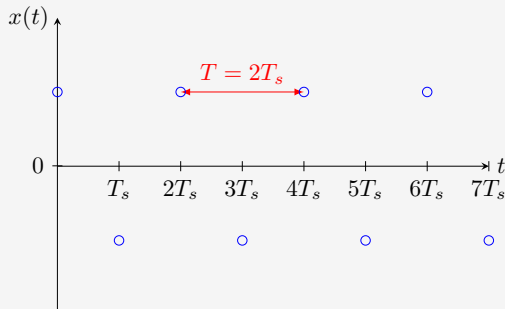


Abb. 9: Signal mit höchstmöglicher Frequenz für ein gegebenes T_s .

Die höchstmögliche Frequenz, die man in das Zeitraster $\{0, T_s, \dots\}$ legen kann, lautet:

$$f_{\max} = \frac{1}{T} = \frac{1}{2T_s}, \quad [f_{\max}] = \text{Hz} \quad (\text{Nyquist Frequenz})$$

Um einen diskreten Regler $C(z)$ zu erstellen verwendet man das diskrete Analogon der Laplace Transformation, die Z-Transformation:

$$X(z) = \mathcal{Z}\{x(k)\} = \sum_{k=0}^{\infty} z^{-k} \cdot x(k)$$

Einige wichtige Eigenschaften der Z-Transformation:

$$x(k+1) \leftrightarrow z \cdot X(z) - z \cdot x(0) \quad (2)$$

$$x(k-1) \leftrightarrow z^{-1} \cdot X(z) \quad (3)$$

$$z = e^{sT_s} \quad (4)$$

In der diskreten Welt ist die unabhängige Variable nicht mehr s , sondern z . Gln. (2) und (3) sind die diskreten Analogon der Laplace-Transformation einer Ableitung und einer Integration. Gl. (4) liefert die Beziehung zwischen $s \in \mathbb{C}$ und $z \in \mathbb{C}$.

Beispiel: Mit Gl. (4) kann man das diskrete Equivalent eines asymptotisch stabilen Pols nach Lyapunov herleiten:

$$z = e^{\sigma T_s + \omega T_s j} = e^{\sigma T_s} \cdot e^{\omega T_s j} = a \cdot e^{\omega T_s j}$$

$$\text{asympt. stabil} \rightarrow \sigma < 0 \rightarrow 0 < a < 1$$

$$|z| = a \rightarrow |z| < 1$$

D.h. die ganze linke Halbebene ($\sigma < 0$, asymptotisch stabile Pole) im s -Raum wird zu einer Scheibe mit Radius 1 ($|z| < 1$) im z -Raum. Der Kreis mit Radius 1 ist nicht inbegriffen.

Emulationen

Die Umsetzung von Gl. (4) ist nichtlinear und deshalb praktisch umständlich, deshalb approximiert man den Zusammenhang häufig, ein Schritt der *Emulation* genannt wird. Emulationen sind dann sinnvoll, wenn die diskreten Zeitschritte T_s sehr klein sind im Vergleich zur Dynamik des beschriebenen Systems. Es gibt mehrere Approximationen für Gl. (4), die die Emulation mit der Approximation $e^x \approx 1 + x$ erleichtern. Dabei wird Gl. (4) jeweils etwas anders geschrieben:

$z = e^{sT_s}$	$\approx 1 + sT_s$	$\Rightarrow s \approx \frac{z-1}{T_s}$	Euler forward
$z = \frac{1}{e^{-sT_s}}$	$\approx \frac{1}{1 - sT_s}$	$\Rightarrow s \approx \frac{z-1}{zT_s}$	Euler backward
$z = \frac{e^{sT_s/2}}{e^{-sT_s/2}}$	$\approx \frac{1 + sT_s/2}{1 - sT_s/2}$	$\Rightarrow s \approx \frac{2(z-1)}{T_s(z+1)}$	Tustin Emulation

Die Tustin-Emulation wird am häufigsten verwendet, insbesondere weil damit die ganze linke s -Halbebene in einen Kreis mit Radius kleiner eins ($|z| < 1$) gemappt wird in der z -Ebene, d.h. ein stabiles kontinuierliches System resultiert in einem stabilen diskreten System (siehe Beispiel zur Stabilität oben). Die Euler backward Emulation erfüllt dieses Kriterium auch, die Euler forward Emulation hingegen nicht.

2 Algebraische Stabilitätskriterien

Es werden zwei Verfahren gezeigt, die zeigen, ob ein Polynom

$$a_n \cdot s^n + a_{n-1} \cdot s^{n-1} + \dots + a_1 \cdot s + a_0, \quad a_n > 0$$

alle Nullstellen in der linken komplexen Halbebene hat. Die Bedingungen $a_i > 0 \forall i$ ist notwendig aber nicht hinreichend.

Hurwitz Kriterium

Die Koeffizienten a_i seien bekannt. Falls alle Matrizen

$$H_1 = a_{n-1}, H_2 = \begin{bmatrix} a_{n-1} & a_n \\ a_{n-3} & a_{n-2} \end{bmatrix}, H_3 = \begin{bmatrix} a_{n-1} & a_n & 0 \\ a_{n-3} & a_{n-2} & a_{n-1} \\ a_{n-5} & a_{n-4} & a_{n-3} \end{bmatrix}, \dots$$

positiv definit sind, dann sind alle Nullstellen in der linken komplexen Ebene.

Kharitonov Kriterium

Die Koeffizienten a_i seien nicht exakt bekannt, man weiss jedoch, dass sie in einer gewissen Region liegen:

$$p(s, a) = [\underline{a}_n, \bar{a}_n] \cdot s^n + [\underline{a}_{n-1}, \bar{a}_{n-1}] \cdot s^{n-1} + \dots + [\underline{a}_0, \bar{a}_0]$$

Das Problem reduziert sich auf die Ecken eines Rechtecks in der komplexen Ebene und es müssen nur folgende Polynome mit dem Hurwitz-Kriterium getestet werden:

$$p_1(s) = \bar{a}_0 + \underline{a}_1 \cdot s + \underline{a}_2 \cdot s^2 + \bar{a}_3 \cdot s^3 + \dots$$

$$p_2(s) = \bar{a}_0 + \bar{a}_1 \cdot s + \underline{a}_2 \cdot s^2 + \underline{a}_3 \cdot s^3 + \dots$$

$$p_3(s) = \underline{a}_0 + \bar{a}_1 \cdot s + \bar{a}_2 \cdot s^2 + \underline{a}_3 \cdot s^3 + \dots$$

$$p_4(s) = \underline{a}_0 + \underline{a}_1 \cdot s + \bar{a}_2 \cdot s^2 + \bar{a}_3 \cdot s^3 + \dots$$