5.2 Digitale Realisierung

Jede digitale Regelung setzt einen Abtastvorgang voraus, d.h. die Entnahme diskreter Funktionswerte aus einem kontinuierlichen Zeitverlauf ➔ grundsätzlicher Aufbau einer digitalen Regelungen gemäß Beiblatt 15.

Regelalgorithmen:

a.) <u>quasikontinuierlicher Fall,</u> d.h. der digitale Regler arbeitet wie ein zeitkontinuierlicher Regler ➔ Entwurf im Kontinuierlichen (wie bisher) und anschließend Zeitdiskretisierung des Reglers.

<u>Vorraussetzung:</u> Abtastzeit genügend klein gegenüber Zeitkonstante der Regelung (Faktor 5-10 kleiner).

b.) <u>zeitdiskreter Fall,</u> d.h. digitaler Regler arbeitet als zeitdiskreter Regler (Abtastregler) ➔ Entwurf im zeitdiskreten auf Basis einer zeitdiskreten Regelsystembeschreibung (z.B. z-Übertragungsfunktion). Siehe dazu Vorlesung „Digitale Regelung".

<u>Hier:</u> *quasikontinuierliche Betrachtung*

1.Schritt: Reglerentwurf im Kontinuierlichen (wie bisher).

2. Schritt: Diskretisierung des Reglers.

Beispiel: Idealer PID-Regler mit $U(s) = F_R(s) \cdot E(s)$, wobei

$$F_R(s) = k_R \frac{(1+T_{R1}s)(1+T_{R2}s)}{s} = k_R \left[\frac{1}{s} + (T_{R1} + T_{R2}) + T_{R1}T_{R2}s \right]$$

$$\Rightarrow \quad F_R(s) = \underbrace{k_R(T_{R1} + T_{R2})}_{k_p} \left[1 + \frac{1}{\underbrace{(T_{R1} + T_{R2})}_{T_N} s} + \underbrace{\frac{T_{R1}T_{R2}}{T_{R1} + T_{R2}}}_{T_v} s \right]$$

k_p: Proportionalbeiwert
T_N: Nachstellzeit
T_v: Vorhaltzeit

Hinweis:

- Beim PID-Regler gilt für $T_{R1} >> T_{R2}$: $k_p = k_R T_{R1}$; $T_N = T_{R1}$; $T_v = T_{R2}$
- Beim PI-Regler gilt mit $F_R(s) = k_R \frac{1+T_R s}{s} = k_R T_R (1 + \frac{1}{T_R s})$:

  $k_p = k_R T_R$; $T_N = T_R$ ➔ $F_R(s) = k_p(1 + \frac{1}{T_N s}) = k_p \frac{1+T_N s}{T_N s}$
- Beim PD-Regler gilt $F_R(s) = k_R(1+T_R s)$: $k_p = k_R$; $T_v = T_R$

  ➔ $F_R(s) = k_p(1+T_v s)$

Damit gilt: $U(s)=k_p(1+\frac{1}{T_N s}+T_v s)\cdot E(s)=k_p\left[E(s)+\frac{1}{T_N}\frac{E(s)}{s}+T_v s\cdot E(s)\right]$

$$\rightarrow \qquad u(t)=k_p\left[e(t)+\frac{1}{T_N}\cdot\int_0^t e(\tau)d\tau+T_v\dot{e}(t)\right] \qquad (1)$$

Übergang zum zeitdiskreten Algorithmus , d.h. Betrachtung von Gleichung (1) nur zu den Abtastzeitpunkten t=kT, k=0,1,2....

- u(t) und e(t) gehen über in $u_k=u(kT)$ und $e_k=e(kT)$; k=0,1,2......
- Integral $\int_0^{t=kT} e(\tau)d\tau$ Rechteckregel: $T\sum_{\upsilon=0}^{k-1}e_\upsilon$

  Trapezregel: $\frac{T}{2}\sum_{\upsilon=0}^{k-1}(e_\upsilon+e_{\upsilon+1})$
- Differentialquotient $\dot{e}(t=kT)$ wird ersetzt durch den Differentialquotienten $\frac{e_k-e_{k-1}}{T}$

  Damit wird aus (1), falls die Rechteckregel verwendet wird:

  $u_k=k_p\left[e_k+\frac{T}{T_N}\sum_{\upsilon=0}^{k-1}e_\upsilon+\frac{T_v}{T}(e_k-e_{k-1})\right]$: PID-Stellungsalgorithmus

  rekursive Form durch Differenzbildung: $u_k-u_{k-1}=k_p[e_k+...]-k_p[e_{k-1}+....]$

$$u_k=u_{k-1}+\underbrace{k_p\left[(1+\frac{T_v}{T})e_k-(1-\frac{T}{T_N}+2\frac{T_v}{T})e_{k-1}+\frac{T_v}{T}e_{k-2}\right]}_{\Delta u_k}:$$

PID-Geschwindigkeitsalgorithmus

Hinweise:

- Manche Stelleinrichtungen, z.B. Schrittmotoren, dürfen nicht durch Absolutwerte u_k, sondern nur durch Inkremente $\Delta u_k=u_k-u_{k-1}$ angesteuert werden.
- Algorithmen mit Trapezregel bzw. für reale ÜF-en in entsprechender Weise ableitbar.
- Aus PID-Algorithmus PI-Algorithmen für T_v=0 bzw. PD-Algorithmen für $T_N\rightarrow\infty$ erhältlich.

3. Schritt: Überprüfung des Entwurfs (insbesondere hinsichtlich der gewählten Abtastzeit T) durch Simulation.