

# 1 Tidsdiskret PID-regulering

## 1.1 Innledning

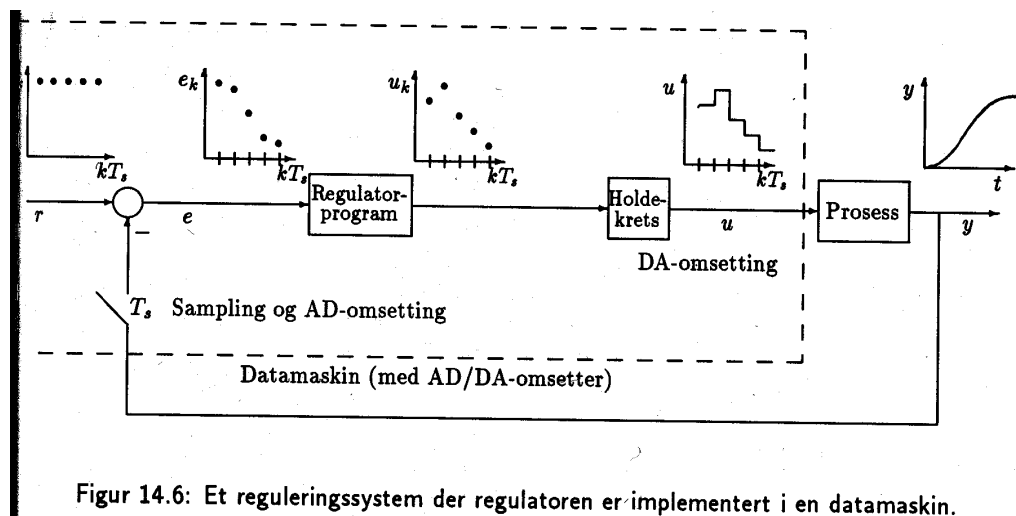
Dette notatet gir en kortfattet beskrivelse av analyse av tidsdiskrete PID-reguleringssystemer. Notatet supplerer kap. 7 og 14 i [1].

En MATLAB-øving m/løsning som dreier seg om innholdet i dette notatet, fins på

<http://www.techteach.no/fag/sig50/0102/leksjon17/matlaboving.htm>.

## 1.2 Tidsdiskret PID-regulator

Figur 1 viser delsystemer og signaler i et generelt tidsdiskret reguleringssystem (figuren er en kopi<sup>1</sup> av figur 14.6 i [1]). Regulatoren er



Figur 14.6: Et reguleringssystem der regulatoren er implementert i en dataskin.

Figur 1:

typisk en tidsdiskret PID-regulator.

Tidsdiskrete PID-regulatorer baseres på diskretisering av den tidskontinuerlige PID-regulatorfunksjonen. Flere diskretiseringsmetoder er aktuelle — spesielt Eulers forovermetode, Eulers bakovermetode, Tustins

<sup>1</sup>gjengitt med forfatterens tillatelse

metode. Det viser seg at de fleste kommersielle tidsdiskrete PID-funksjonene er basert på Eulers bakovermetode.

I det følgende gjengis en diskret PID-regulator på såkalt absolutt form (ikke inkrementell) basert på diskretisering (med Eulers bakovermetode) av følgende tidskontinuerlige regulatorfunksjon:

$$u(t) = u_0 + u_p + u_i + u_{df} \quad (1)$$

der

$$u_p(t) = K_p e(t) \quad (\text{proporsjonalledd}) \quad (2)$$

$$u_i(t) = \frac{K_p}{T_i} \int_0^t e \, d\tau \quad (\text{integralledd}) \quad (3)$$

$$u_{df}(t) = K_p T_d \frac{de_f(t)}{dt} \quad (\text{derivatledd m/lavpassfilter}) \quad (4)$$

der det lavpassfiltrerte avviket  $e_f$  er gitt ved filter-differensiallikningen

$$T_f \frac{de_f(t)}{dt} + e_f(t) = e(t) \quad (5)$$

der  $T_f$  er filtertidskonstanten. (I denne PID-regulatoren er det antatt at vekt faktorene for referansen er 1 både i proporsjonalleddet og i derivatleddet.)

Vi finner en tidsdiskret PID-regulatorfunksjon ved å diskretisere (1) – (5). Hvis vi bruker Eulers bakovermetode, som generelt lyder ( $T_s$  er samplingsintervallet)

$$\begin{aligned} \frac{dx(t)}{dt} &\approx \frac{x(t_k) - x(t_{k-1})}{T_s} & (6) \\ &\equiv \frac{x(kT_s) - x[(k-1)T_s]}{T_s} \\ &\equiv \frac{x(k) - x(k-1)}{T_s} \end{aligned}$$

fås følgende diskrete PID-regulator:

$$u(k) = u_0(k) + u_p(k) + u_i(k) + u_{df}(k) \quad (7)$$

der

$$u_p(k) = K_p e(k) \quad (8)$$

$$u_i(k) = u_i(k-1) + \frac{K_p T_s}{T_i} e(k) \quad (9)$$

$$u_{df}(k) = \frac{K_p T_d}{T_s} [e_f(k) - e_f(k-1)] \quad (10)$$

der  $e_f$  er gitt ved filter-differenslikningen

$$e_f(k) = \frac{1}{1 + T_s/T_f} e_f(k-1) + \frac{T_s/T_f}{1 + T_s/T_f} e(k) \quad (11)$$

|                       | $K_p$        | $T_i$        | $T_d$      |
|-----------------------|--------------|--------------|------------|
| <b>P-regulator:</b>   | $0,5K_{pk}$  | —            | —          |
| <b>PI-regulator:</b>  | $0,45K_{pk}$ | $T_{pk}/1,2$ | —          |
| <b>PID-regulator:</b> | $0,6K_{pk}$  | $T_{pk}/2$   | $T_{pk}/8$ |

Tabell 1: Ziegler-Nichols' lukket-sløyfe-metode: Formler for regulatorparametrene i en PID-regulator

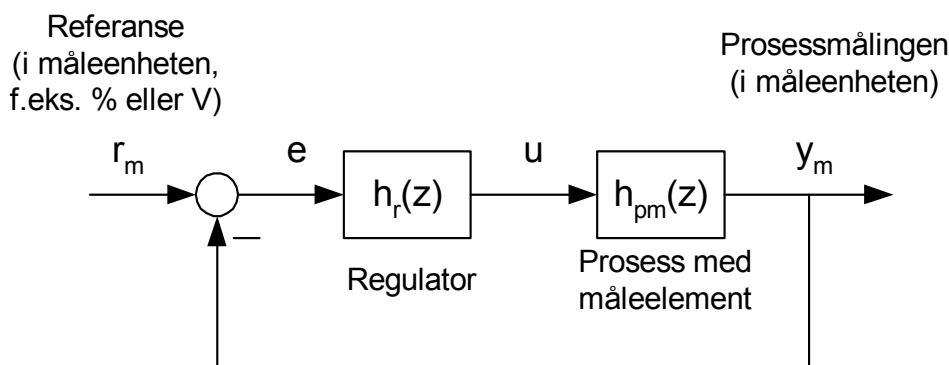
### 1.3 Innstilling av regulatorparametre

Regulatorparametrene i en tidsdiskret PID-regulator kan stilles inn med samme metoder som for en tidskontinuerlig PID-regulator. Eksempelvis kan vi bruke Ziegler-Nichols' lukket-sløyfe-metode, jf. kap. 7 i [1]. Siden regulatorfunksjonen er tidsdiskret, skal Ziegler-Nichols' parameterformler egentlig modifiseres noe, men dersom vi antar at samplingsintervallet  $T_s$  er lite sammenliknet med prosessens responstid  $T_r$  (anslagvis mindre enn  $1/5$  av responstiden), vil modifikasjonene bety lite, og vi kan ganske enkelt bruke de opprinnelige formlene, som er vist i tabell 1. I tabellen er  $K_{pk}$  kritisk regulatorforsterkning og  $T_{pk}$  kritisk periodetid.

### 1.4 Analyse av diskret reguleringsystem

#### 1.4.1 Transferfunksjoner i reguleringsystemet

**Innledning.** Figur 2 viser et matematisk blokkdiagram av et PID-reguleringsystem. Blokkene inneholder  $z$ -transferfunksjoner. Disse



Figur 2: Blokkdiagram for tidsdiskret reguleringsystem

kan finnes som forklart nedenfor. Merk at prosessens og måleelementets transferfunksjon er slått sammen (multiplisert) for å få en kombinert

transferfunksjon (de framstår kombinert sett fra regulatorens side).

**Regulatorens transferfunksjon.** Regulatorens transferfunksjon  $h_r(z)$  fra reguleringsavviket  $e$  til pådraget  $u$  kan finnes ved å  $z$ -transformere den tidsdiskrete regulatorfunksjonen. La oss som eksempel finne  $z$ -transferfunksjon for den tidsdiskrete PI-regulatoren gitt ved (7) – (9).  $z$ -transferfunksjon av (7) gir (vi kan sette  $u_0 = 0$ )

$$u(z) = u_p(z) + u_i(z) \quad (12)$$

$z$ -transferfunksjon av (8) gir

$$u_p(z) = K_p e(z) \quad (13)$$

$z$ -transferfunksjon av (9) gir

$$u_i(z) = z^{-1}u_i(z) + \frac{K_p T_s}{T_i} e(z)$$

som gir

$$u_i(z) = \frac{K_p T_s}{T_i(1 - z^{-1})} e(z) \quad (14)$$

Ved å sette inn (13) og (14) i (12) får vi

$$\frac{u(z)}{e(z)} = h_{r,\text{diskret}}(z) = \frac{K_p(T_s + T_i)z - K_p}{T_i z - T_i}$$

som er PI-regulatorens  $z$ -transferfunksjon.

I forbindelse med analyse og design av reguleringsystemet i MATLAB trengs en LTI-modell<sup>2</sup> av regulatoren. Dessverre har ikke MATLAB noen funksjon for å diskretisere med Eulers bakovermetode eller Eulers forovermetode, så du må lage LTI-modellen vha. funksjonen `tf` slik (det antas at parametrene `Kp`, `Ti` og `Ts` eksisterer i MATLABs arbeidsrom):

```
hrteller = [Kp*(Ts+Ti), -Kp];  
hrnevner = [Ti, -Ti];  
hr = tf(hrteller,hrnevner,Ts)
```

### Prosessens og måleelementets kombinerte transferfunksjon.

Prosessens og måleelementets kombinerte transferfunksjon,  $h_{pm}(z)$ , må diskretiseres under forutsetning av at det er nullte ordens holdeelement på prosessens inngang. Dette skyldes at prosessen faktisk blir styrt via et nullte ordens holdeelement (elementet holder pådraget fast mellom hvert

---

<sup>2</sup>LTI står for Linear Time-Invariant. LTI-modeller lages med funksjoner i MATLABs Control System Toolbox.

tidsskritt), se figur 1. Det er nokså komplisert å utføre manuell diskretisering (med håndregning) av en gitt  $s$ -transferfunksjon med nullte-ordens holdeelement på prosessinngangen (metoden er beskrevet bl.a. i [2]). I praksis bruker vi nok heller MATLAB.

MATLAB kan brukes til å finne  $h_{pm}(z)$  svarende til en gitt tidskontinuerlig transferfunksjon  $h_{pm,kont}(s)$ . Her er et eksempel: Gitt transferfunksjonen

$$h_{pm,kont}(s) = \frac{K}{Ts + 1} e^{-\tau s}$$

Denne transferfunksjonen kan defineres som en LTI-modell i MATLAB slik (det antas at parametrene fins i MATLABs arbeidsrom):

```
hpmkontteller=[K];
hpmkontnevner=[T,1];
hpmkont = tf(hpmkontteller,hpmkontnevner,'InputDelay',tau)
```

Fra den tidskontinuerlige LTI-modellen `hpmkont` kan vi nå finne den tilsvarende tidsdiskrete LTI-modellen `hpmdiskret` slik:

```
hpmdiskret = c2d(hpmkont,Ts,'zoh')
```

der 'zoh' står for zero-order hold, dvs. nullte ordens holdeelement, dvs. at prosessen antas å være påvirket av et trappetrinnsformet (stykkevis konstant) inngangssignal. Funksjonsnavnet `c2d` står for "continuous-to-discrete".

### 1.4.2 Reguleringsystemets følgeforhold

Analyse av reguleringsystemet gjøres vanligvis på basis av en (tidsdiskret)  $z$ -transferfunksjon for reguleringsystemet. Den mest aktuelle transferfunksjonen er *følgeforholdet*, som er transferfunksjonen  $M(z)$  fra referansen  $r_m$  (som er referansen i måleenheter) til prosessmålingen  $y_m$ . Vi kan finne  $M(z)$  fra blokkdiagrammet i figur 2 slik:  $y_m$  er gitt ved

$$y_m(z) = h_{pm}(z)h_r(z)[r_m(z) - y_m(z)]$$

som gir

$$\frac{y_m(z)}{r_m(z)} = M(z) = \frac{h_r(z)h_{pm}(z)}{1 + h_r(z)h_{pm}(z)} = \frac{h_0(z)}{1 + h_0(z)} \quad (15)$$

der

$$h_0(z) = h_r(z)h_{pm}(z) \quad (\text{sløyfetransferfunksjonen}) \quad (16)$$

Et eksempel: Anta at

$$h_r(z) = K_p$$

og

$$h_{pm}(z) = \frac{b}{z - a}$$

Følgeforholdet blir da

$$M(z) = \frac{h_r(z)h_{pm}(z)}{1 + h_r(z)h_{pm}(z)} \quad (17)$$

$$= \frac{K_p \frac{b}{z-a}}{1 + K_p \frac{b}{z-a}} \quad (18)$$

$$= \frac{K_p b}{z - (a - K_p b)} \quad (19)$$

Du kan lage en LTI-modell for følgeforholdet  $M(z)$  vha. funksjonene **series** og **feedback** i MATLAB. **series** kombinerer to transferfunksjoner som står i serie, dvs. multipliserer transferfunksjonene, mens **feedback** lager en transferfunksjon av en blokkdiagramstruktur som inneholder en tilbakekopling. Figur 3 viser et utdrag av **help** for **feedback**.

```

>> help feedback

FEEDBACK Feedback connection of two LTI models.

SYS = FEEDBACK(SYS1, SYS2) computes an LTI model SYS for
the closed-loop feedback system

    u ---->0---->[ SYS1 ]----+----> y
                       |           |
                       +-----[ SYS2 ]<----+
                                     y = SYS * u

Negative feedback is assumed and the resulting system SYS
maps u to y. To apply positive feedback, use the syntax
SYS = FEEDBACK(SYS1, SYS2, +1).
  
```

Figur 3: Utdrag av **help feedback**

Følgende uttrykk genererer  $M(z)$  i form av en LTI-modell med navn **M** (det antas at LTI-modellene **hr** og **hpm** fins i MATLABs arbeidsrom, og at **Ts** er samplingsintervallet i reguleringsystemet):

```

tilbakekopling=tf(1,1,Ts);%Transferfunksjon 1 i
tilbakekoplingen.
M=feedback(series(hr,hpm),tilbakekopling);
  
```

LTI-modellen (**M**) kan så benyttes i bl.a. simulering, stabilitetsanalyse og

frekvensresponsanalyse. Dette beskrives i de følgende kapitler.

### 1.4.3 Simulering

Du kan simulere reguleringsystemet vha. simuleringsfunksjonene `step`, `impulse` eller `lsim` i Control System Toolbox eller ved å bruke SIMULINK.

Anta som eksempel at du vha. `step` skal simulere responsen i prosessmålingen  $y_m$  etter et enhetsprang i referansen  $r_m$ . Dette kan gjøres med følgende uttrykk:

```
step(M)
```

der  $M$  er reguleringsystemets følgeforhold i form av en LTI-modell.

`step(M)` viser sprangresponsen i et plott. Dersom spranget skal ha høyde  $U$  i stedet for 1, kan du bruke

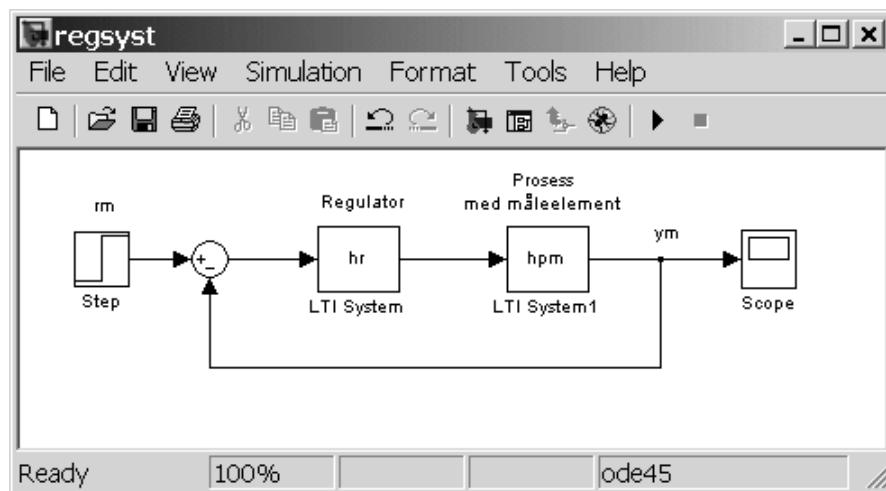
```
step(M*U)
```

Dersom du ønsker å få den simulerte responsen returnert som et array (eller vektor) i MATLAB, kan du skrive

```
[ym,t]=step(M)
```

der  $y_m$  er arrayet for responsen og  $t$  er et array som inneholder de tilhørende tidspunktene.

Dersom du skal simulere reguleringsystemet i SIMULINK, kan du selvsagt bruke én LTI-blokk for  $M$ , men det er kanskje mer illustrativt å lage et blokkdiagram av reguleringsystemet bestående av en LTI-blokk for regulatoren og en LTI-blokk for prosessen, som illustrert i figur 4.



Figur 4: SIMULINK-blokkdiagram av et reguleringsystem

#### 1.4.4 Reguleringsystemets statiske egenskaper

Det meste av tiden et reguleringsystem er i virksomhet, er referansen konstant og alle variable har tilnærmet konstante verdier. Dersom du skal analysere reguleringsystemet under slike statiske forhold, kan du i beregningene benytte statiske versjoner av de aktuelle transferfunksjonene. Det kan vises at den statiske versjonen av transferfunksjonen  $h(z)$  er (se f.eks. [2])

$$h_{\text{statisk}} = h(z)|_{z=1} \quad (20)$$

La oss som eksempel beregne den statiske responsen i  $y_m$  svarende til referansen  $r_m$  er konstant og lik  $R_m$  for følgeforsholdet gitt ved (19). Den statiske responsen blir

$$\begin{aligned} y_{m,\text{statisk}} &= M_{\text{statisk}} \cdot R_m \\ &= M(z)|_{z=1} \cdot R_m \\ &= \frac{K_p b}{1 - (a - K_p b)} R_m \end{aligned} \quad (21)$$

#### 1.4.5 Stabilitetsanalyse

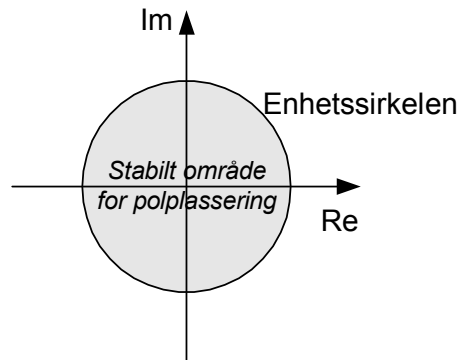
Reguleringsystemets stabilitet kan fastslås fra plasseringen av polene av følgeforsholdet  $M(z)$  i det komplekse plan. Sammenhengen mellom polplassering og stabilitetsegenskap er som følger:

- **Asymptotisk stabilt system:** Alle poler innenfor enhetssirkelen (ingen på), dvs. at *alle polene har absoluttverdi mindre enn 1*.
- **Marginalt stabilt system** (dvs. på stabilitetsgrensen): Enkeltpoler på enhetssirkelen, men ingen utenfor.
- **Ustabilt system:** Minst én pol utenfor enhetssirkelen, eller multiple poler på enhetssirkelen (og ingen utenfor).

Figur 5 illustrerer det stabile området (dvs. det indre av enhetssirkelen) i det komplekse plan. La oss som eksempelet finne betingelsen for at reguleringsystemet som har følgeforshold gitt ved (19) skal være asymptotisk stabilt. Systemet har én pol, nemlig  $(a - K_p b)$ , og kravet for asymptotisk stabilitet er at denne polen har absoluttverdi mindre enn 1, dvs. at

$$|a - K_p b| < 1$$

MATLAB-funksjonen `pzmap` kan brukes til å beregne evt. plote polene (og nullpunktene) for en LTI-modell. Dersom LTI-modellen er  $M$  (reguleringsystemets følgeforshold), finnes polene (og nullpunktene) med



Figur 5: Det stabile område for polplassering i det komplekse plan

`[polene,nullpunktene] = pzmap(M)`

Absoluttverdien av polene kan beregnes (og listes opp) med `abs(polene)`

Plott (ingen eksplisitt beregning) av polene og nullpunktene gjøres med `pzmap(M)`

#### 1.4.6 Frekvensresponsanalyse

Frekvensresponsen for tidsdiskrete systemer har samme definisjon eller betydning som for tidskontinuerlige modeller. Ut fra frekvensresponsen kan vi finne bl.a. båndbredden<sup>3</sup> for et reguleringsystem.

Formelen for beregning av frekvensresponsen for en tidsdiskret transferfunksjon  $h(z)$  er

$$h(e^{j\omega T}) = h(z)|_{z=e^{j\omega T}} \quad (22)$$

der  $T$  er samplingsintervallet.  $z$  i transferfunksjonen skal altså erstattes av  $z = e^{j\omega T}$ . Dette kan vi anvende på f.eks. et reguleringsystems følgeforhold  $M(z)$ . Eksempelvis er frekvensresponsen for (19)

$$M(e^{j\omega T}) = \frac{K_p b}{e^{j\omega T} - (a - K_p b)}$$

Frekvensresponsens *amplitdefunksjon* og *fasefunksjon* kan så finnes etter følgende operasjoner (som ender opp med at den komplekse størrelsen

<sup>3</sup>som angir grensen mellom stoppbåndet og passbåndet for frekvensresponsens amplitdefunksjon

$M(e^{j\omega T})$  er skrevet på polar form):

$$\begin{aligned}
 M(e^{j\omega T}) &= \frac{K_p b}{e^{j\omega T} - (a - K_p b)} \\
 &= \frac{K_p b}{\cos(\omega T) + j \sin(\omega T) - (a - K_p b)} \\
 &= \frac{K_p b}{[\cos(\omega T) - (a - K_p b)] + j [\sin(\omega T)]} \\
 &= \frac{K_p b}{\sqrt{[\cos(\omega T) - (a - K_p b)]^2 + [\sin(\omega T)]^2}} e^{j \arctan\left\{\frac{\sin(\omega T)}{\cos(\omega T) - (a - K_p b)}\right\}} \\
 &= \frac{K_p b}{\sqrt{[\cos(\omega T) - (a - K_p b)]^2 + [\sin(\omega T)]^2}} e^{-j \arctan\left\{\frac{\sin(\omega T)}{\cos(\omega T) - (a - K_p b)}\right\}}
 \end{aligned}$$

Amplitudedefunksjonen er derfor

$$\left| M(e^{j\omega T}) \right| = \frac{K_p b}{\sqrt{[\cos(\omega T) - (a - K_p b)]^2 + [\sin(\omega T)]^2}}$$

og fasefunksjonen er

$$\arg M(e^{j\omega T}) = - \arctan\left\{ \frac{\sin(\omega T)}{\cos(\omega T) - (a - K_p b)} \right\} \text{ [rad]}$$

Disse funksjonene kan plottes i f.eks. et Bode-diagram.

I MATLAB kan vi bruke funksjonen `bode` til å plote amplitude- og fasefunksjonen for LTI-modells frekvensrespons. Et eksempel: Følgende uttrykk genererer Bode-plottet for følgeforholdet  $M$  for et reguleringsystem:

```

frekvensomraade = {frek_min, frek_maks}; % [rad/s]
bode(h, frekvensomraade)

```

## Referanser

- [1] F. Haugen: **Regulering av dynamiske systemer, bind 1.** Tapir forlag, 2000.
- [2] F. Haugen: **Regulering av dynamiske systemer, bind 2.** Tapir forlag, 2001.