

Lösningar till tentamen i Industriell reglerteknik TSRT07

Tentamensdatum: 2023-08-25

Martin Enqvist

1. (a) Z-transformering av sambanden som beskriver den tidsdiskreta regulatorm ger

$$I(z) = \frac{KT_S z}{T_i(z-1)} E(z) = \frac{Kz}{2z-2} E(z)$$

och

$$U(z) = KE(z) + I(z) = K \frac{3z-2}{2z-2} E(z).$$

Om man först beräknar den tidsdiskreta motsvarigheten $G_d(z)$ till den tidskontinuerliga överföringsfunktionen $G(s)$ kan man bestämma gränsen för K med hjälp av nyquistkurvan för systemet $F(z)G_d(z)$, där

$$F(z) = \frac{3z-2}{2z-2}.$$

Det visar sig att detta system har amplitudmarginalen $1/0.212 \approx 4.72$, och eftersom det öppna systemet är stabilt innebär det att man måste välja K mindre än 4.72 för att det slutna systemet ska vara stabilt. Matlab-kod:

```
G=tf(1,[1 2 2]);
Gd=c2d(G,1);
F=tf([3 -2],[2 -2],1);
nyquist(F*Gd)
```

- (b) En smithprediktor innehåller en modell som simuleras när regulatorm är inkopplad och därför fungerar denna reglerstrategi bara för stabila system. Eftersom det aktuella systemet är instabilt är det inte möjligt att använda en smithprediktor.
- (c) Eftersom det slutna systemet har en inre loop som reglerar gasflödet och en yttre som reglerar temperaturen är detta ett typiskt exempel på kaskadreglering. Variationer i gstrycket kan kaskadregulatorm kompensera för innan de har hunnit påverka temperaturen särskilt mycket och i detta fall tjänar man alltså på att mäta och återkoppla f . Variationer i vattenflödet har ingen effekt på f och därför är det inte någon fördel att använda kaskadreglering för att kompensera för dessa.
- (d) Eftersom regulatorförstärkningen K_n är tidsvariabel och bestäms av en signal z_n är detta en PI-regulator med parameterstyrning. Om z_n mäts med en extra sensor kommer reglersystemet att kunna påverkas av brus och fel i denna sensor. Jämfört med en vanlig PI-regulator finns det en sensor till som kan gå sönder eller påverkas av störningar.
2. (a) Genom att rita upp stegsvaret för systemet kan man bestämma modellparametrarna L , K_p och T enligt följande principer.

- * Dödtiden L är den tid under vilken stegsvaret (ungefär) är lika med noll.
- * Den statiska förstärkningen K_p är det värde som stegsvaret närmar sig när tiden går mot oändligheten.
- * Tidskonstanten T väljs så att systemets och modellens stegsvar båda (ungefär) har värdet $0.63K_p$ vid tidpunkten $L + T$.

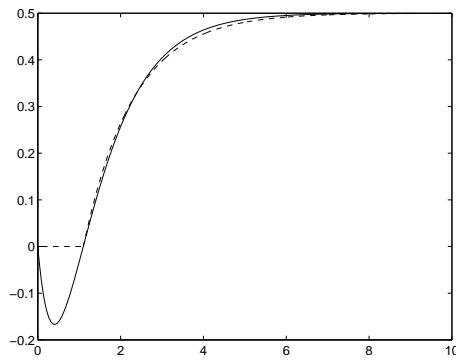
Systemets och modellens stegsvar visas i figur 1.

Treparametermodellen blir

$$\hat{G}(s) = \frac{0.5}{1.2s + 1} e^{-1.1s},$$

vilket ger regulatorparametrarerna $K = 1.4$ och $T_i = 1.2$. Matlab-kod:

```
G=tf([-1 1],[1 3 2]);
Kp=0.5;
L=1.1;
T=1.2;
Ghat=tf(Kp,[T 1], 'InputDelay',L);
step(G,Ghat)
lambda=0.5;
K=T/(Kp*(lambda*T+L))
Ti=T
```



Figur 1: Stegsvaret för systemet (heldraget) och treparametermodellen (streckat) i uppgift 2(a).

- (b) När referenssignalen skalas med $\alpha \neq 1$ i P-delen får man en regulator med två frihetsgrader. Regulatorns överföringsfunktioner är

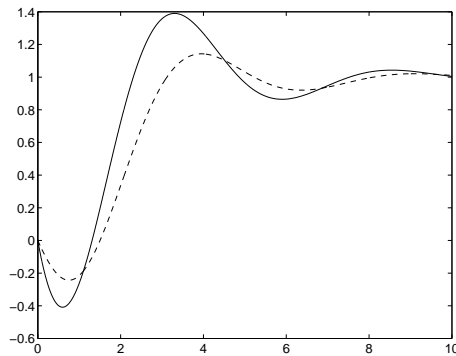
$$F_r(s) = K \left(\alpha + \frac{1}{T_i s} \right)$$

och

$$F_y(s) = K \left(1 + \frac{1}{T_i s} \right).$$

I simuleringarna kan man se att ett minskat α ger en minskad översläng i stegsvaret men samtidigt en större tidskonstant. I figur 2 visas det slutna systemets stegsvar för två val av α . Matlab-kod:

```
Fy=K*tf([Ti 1],[Ti 0]);
Gc=feedback(Fy*G,1);
a=0.4;
Fr=K*tf([a*Ti 1],[Ti 0]);
Gcmod=minreal(Fr*G/(1+Fy*G));
step(Gc,Gcmod)
```



Figur 2: Stegsvaret för det slutna systemet med $\alpha = 1$ (heldraget) och $\alpha = 0.4$ (streckat) i uppgift 2(b).

3. (a) Referensföljningsegenskaperna kan sammanfattas med en referensmodell

$$G_m(s) = \frac{1}{sT_x + 1}$$

och framkopplingslänken ska väljas som

$$F_f(s) = \frac{G_m(s)}{G(s)} = \frac{sT_x + 1}{K_p(sT_x + 1)}.$$

Regulatorn definieras av överföringsfunktionerna $G_m(s)$, $F_f(s)$ och $F(s)$.

- (b) Vid framkoppling från referenssignalen är det viktigare att man har en noggrann modell än vid återkoppling eftersom framkopplingstermen beräknas enbart utifrån referenssignalen och modellen. Vid återkoppling justeras däremot styrsignalen baserat på mätningarna av utsignalen och därmed blir en återkoppling inte lika känslig för modellfel.

(c) Styrsignalen kan skrivas

$$U(s) = (F_f(s) + F(s)G_m(s))R(s) - F(s)Y(s)$$

i fallet med framkoppling från referenssignalen och

$$U(s) = F(s)\tilde{F}_r(s)R(s) - F(s)Y(s)$$

i fallet med en allmän regulator med två frihetsgrader. Genom att identifiera överföringsfunktionen som står framför $R(s)$ får man

$$F(s)\tilde{F}_r(s) = F_f(s) + F(s)G_m(s),$$

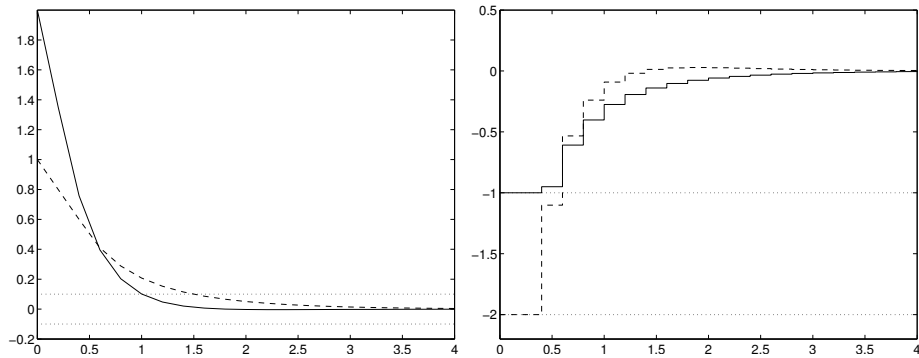
vilket tillsammans med uttrycket $F_f(s) = G_m(s)/G(s)$ (som gäller vid ideal framkoppling) ger

$$\tilde{F}_r(s) = \frac{F_f(s)}{F(s)} + G_m(s) = \frac{G_m(s)}{G(s)F(s)} + G_m(s) = \frac{(1 + G(s)F(s))G_m(s)}{G(s)F(s)} = \frac{G_m(s)}{G_c(s)}.$$

4. (a) Man kan till exempel välja regulatorparametrarna $N = 20$,

$$Q_1 = \begin{pmatrix} 1.8 & 0 \\ 0 & 8 \end{pmatrix}$$

och $Q_2 = I$. Plottar som visar att det slutna systemet uppfyller de ställda kraven visas i figur 3.



(a) Tillståndet x_1 (heldraget) och x_2 (streckat). (b) Insignalen u_1 (heldraget) och u_2 (streckat).

Figur 3: Plottar som visar att designkraven i uppgift 4(a) är uppfyllda.

Matlab-kod:

```
A=[0 0 ; -.2 -.4];
B=[1 0; 0.4 1];
C=eye(2);
D=zeros(2,2);
Gsys=ss(A,B,C,D);
M=eye(2);
x0=[1;2];
Ts=.2;
Gsysd=c2d(Gsys,Ts);
F=Gsysd.A;
G=Gsysd.B;
N=20;
Q1=diag([1.8 8]);
Q2=diag([1 1]);
ubounds=[-1 1;-2 2];
```

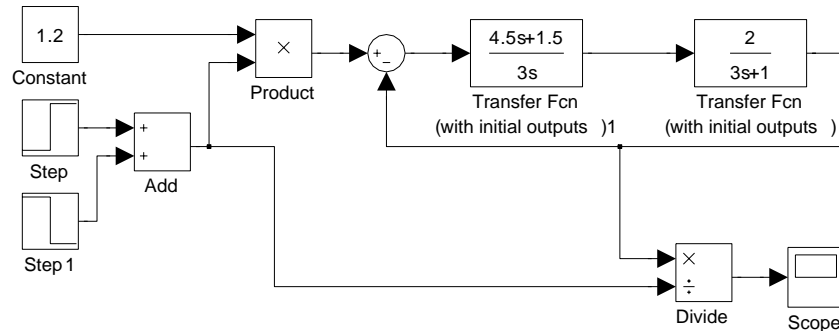
(b) Vid tidpunkten k finns $r(k)$ tillgängligt men inte $r(k+j)$ för $j = 1, \dots, N-1$. Detta är dock inte ett så stort problem som det först kan verka eftersom en MPC-regulator bara ställer ut det första elementet $u(k)$ i den beräknade styrsignalsekvensen $u(k+j)$, $j = 0, \dots, N-1$, vid tiden k . Vid nästa sampeltidpunkt, $k+1$, finns ett nytt värde på referenssignalen tillgängligt och om det värdet skiljer sig åt från det förra kommer den nya beräknade styrsignalsekvensen

att se helt annorlunda ut än den som beräknades vid tiden k . Resultatet är att varje $u(k)$ kommer att bero på $r(k)$ och att följningen av en varierande referenssignal inte kommer att bli så dålig som man kanske först befarar när man ser antagandet om en konstant referenssignal i MPC-problemet.

5. (a) Eftersom man har krav på att kvoten mellan luft- och bränsletillförseln ska vara konstant är det lämpligt att använda en kvotregulator.
- (b) Eftersom robusthetsmarginalerna inte får påverkas av hur stort bränsleflödet y_r är måste man först bilda produkten ry_r , där $r = 1.2$, och sedan reglera på reglerfelet $\tilde{e}(t) = ry_r(t) - y(t)$, där y är luftinflödet. Eftersom $G(s)$ är ett första ordningens system utan tidsfördröjning kan vi använda IMC-trimning med $T_c = 1$ för att få ett slutet system med de önskade egenskaperna. Detta ger

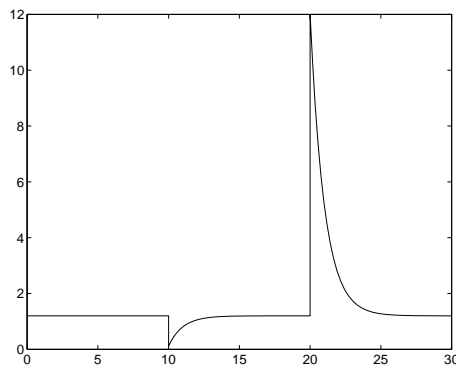
$$K = \frac{T}{K_p T_c} = \frac{3}{2 \cdot 1} = 1.5 \quad \text{och} \quad T_i = T = 3.$$

Ett simulinkschema som kan användas för att simulera det slutna systemet visas i figur 4.



Figur 4: Simulinkschema till uppgift 5(b).

Resultatet av simuleringen visas i figur 5.



Figur 5: Kvoten mellan luft- och bränsleflödet som funktion av tiden i uppgift 5(b).

- (c) Att bilda kvoten mellan två signaler är en olinjär operation och det är därför inte förvånande att signalernas storlek påverkar det transienta felet vid kvotregleringen. Om man antar att $y(10) = 0.12$ och $y_r(10) = 1$ får man de båda initiala reglerfelen

$$\tilde{e}(10) = ry_r(10) - y(10) = 1.2 - 0.12 = 1.08,$$

$$e(10) = r - \frac{y(10)}{y_r(10)} = 1.2 - \frac{0.12}{1} = 1.08.$$

Med $y(20) = 1.2$ och $y_r(20) = 0.1$ får man istället

$$\tilde{e}(20) = ry_r(20) - y(20) = 0.12 - 1.2 = -1.08,$$

$$e(20) = r - \frac{y(20)}{y_r(20)} = 1.2 - \frac{1.2}{0.1} = -10.8.$$

Eftersom regulatorn bara har kunskap om $\tilde{e}(t)$ kommer det att ta lika lång tid för $y(t)$ att ställa in sig på rätt nivå i båda fallen. Felet i den storhet som man egentligen vill reglera, y/y_r , är dock tio gånger större vid tiden $t = 20$ än vid $t = 10$.