



**Høgskolen i Telemark**

Avdeling for teknologiske fag

Masterutdanningen

---

## **HOVEDOPPGAVE 2005**

Student: Ole Christian Skilbred

Oppgavens tittel: Design av desentraliserte og suboptimale regulatorer for diskrete systemer



**Avdeling for teknologiske fag**

Adresse: Kjølnes, 3914 Porsgrunn, telefon 35 57 50 00, telefax 35 55 75 47

---

Ingeniørutdanning - Sivilingeniørutdanning - Doktorgradsutdanning



# Høgskolen i Telemark

Avdeling for teknologiske fag

Masterutdanningen

## RAPPORT FRA HOVEDOPPGAVE, FAG F4203

**Student** : Ole Christian Skilbred

**Oppgavens tittel** : Design av desentraliserte og suboptimale regulatorer for diskrete systemer

**Signatur** : . . . . .

**Antall sider** : 113

**Emneord** : **Avansert regulering**  
**Optimalisering**  
**Programmering**  
**Implementering**

**Faglærer** : David Di Ruscio sign.: . . . . .

**Biveileder** : sign.: . . . . .

**Sensor** : Olav Aaker sign.: . . . . .

**Ekstern partner** :

**Tilgjengelighet** : Åpen

**Godkjent for arkivering** (sign. faglærer): . . . . . **Dato:** . . . . .

### Sammendrag:

Denne rapporten beskriver hvordan optimaliseringsproblemet knyttet til optimalregulering, kan løses ved hjelp av optimaliseringsrutiner. En algoritme er laget og implementert i MatLab versjon 6.5 (R13) for optimal innstilling av tilbakekoblede systemer. Tilbakekoplingen kan komme fra tilstandsvektoren eller utgangsvektoren, og tilbakekoplingsmatrisene kan ha en spesifisert struktur. Løsningen er optimal på den måten at et kriterie minimaliseres. En diskret tilstandsrommodell sammen med vektmatriser er utgangspunktet for den implementerte algoritmen.

To av MatLabs optimaliseringsrutiner, "fminsearch" og "fmincon", er brukt for utarbeiding av algoritmen. Det trengs kun å løse en Lyapunovligning for hver iterasjon, der Lyapunov-ligningen er en omskrivning av Ricatti-ligningen. Dette er gjort for systemer både med og uten direktevirkning i tilstandsrommodellen.

Hvis en stabiliserende regulator er tilgjengelig er det vist at optimaliseringsrutinene finner en løsning. Denne løsningen kan være en multivariabel regulator eller den kan ha en spesifisert struktur. Det er også laget to algoritmer for hvordan ustabile prosesser kan stabiliseres. Og der løsningen er stabil for hver iterasjon i optimaliseringsrutinen.

Dersom løsningene har en spesifisert struktur er det vist hvordan dette kan utnyttes til å stille inn optimale PI-regulatorer. Dette gjelder både for PI-regulator på standardform og for PI-regulator på endringsform.

Det er diskutert muligheten til å bruke systemidentifikasjonsmetoder for å estimere den diskrete tilstandsrommodellen som er utgangspunkt for regulatordesignmetodene. Disse metodene gir en "raskere" vei til en tilstandsrommodell, fordi de genereres av inngangs- og utgangsdata, og ikke av vanlige balanseligninger brukt i modellering. Modeller som er generert av en systemidentifikasjonsmetode er egnet til bruk når prosessen ligger rundt arbeidspunktet.

# Innhold

<b>1</b>	<b>Innledning</b>	<b>5</b>
1.1	Litt om regulatorer . . . . .	5
1.2	Oppgaven . . . . .	5
<b>2</b>	<b>Optimal løsning som et optimaliseringsproblem</b>	<b>7</b>
2.1	Optimal løsning . . . . .	7
2.2	Optimaliseringsrutiner . . . . .	14
<b>3</b>	<b>Lyapunovs diskrete matrise ligning</b>	<b>17</b>
3.1	Utleddning . . . . .	17
3.1.1	Uten kryssledd . . . . .	17
3.1.2	Med kryssledd . . . . .	19
<b>4</b>	<b>Tilbakekopling fra <math>y_k</math></b>	<b>22</b>
4.1	Utleddning . . . . .	22
4.1.1	Uten integralvirkning . . . . .	22
4.1.2	Med integralvirkning . . . . .	24
4.1.3	Ulike typer augmentering . . . . .	25
4.1.4	Litt om vekting . . . . .	27

<b>5</b>	<b>Syntese av PI-regulatorer</b>	<b>29</b>
5.1	Standard PI-regulatorer . . . . .	29
5.2	PI-regulator på endringsform . . . . .	32
<b>6</b>	<b>Metoder for stabilisering av prosesser</b>	<b>35</b>
6.1	To løsninger . . . . .	35
6.1.1	Innføring av $\beta$ . . . . .	35
6.1.2	Innføring av $\delta$ . . . . .	37
<b>7</b>	<b>Eksempler</b>	<b>39</b>
7.1	Simuleringer . . . . .	40
7.1.1	SISO-system med direktevirkning . . . . .	40
7.1.2	AIRC . . . . .	44
7.1.3	Example 3 . . . . .	59
7.1.4	Destillasjonskolonne . . . . .	66
7.1.5	Vanntank . . . . .	85
7.1.6	Unstable plant . . . . .	89
<b>8</b>	<b>Diskusjon</b>	<b>94</b>
8.1	Optimal løsning . . . . .	94
8.2	Syntese av PI-regulatorer . . . . .	94
8.3	Eksemplene . . . . .	95
8.4	Stabiliseringsproblemet . . . . .	95
8.5	Bruk av systemidentifikasjonsmetoder . . . . .	97
<b>9</b>	<b>Konklusjon</b>	<b>99</b>

<b>A Modeller</b>	<b>102</b>
A.1 SISO . . . . .	102
A.2 AIRC . . . . .	102
A.3 Destillasjonskolonne . . . . .	103
A.4 Example 3 . . . . .	106
A.5 Vanntank . . . . .	106
A.6 Unstable plant . . . . .	107
<b>B Oversikt over innhold på CD</b>	<b>108</b>
B.1 Innhold . . . . .	108
<b>C Oppgavebeskrivelse</b>	<b>109</b>
C.1 Oppgaven . . . . .	109

# Forord

Denne rapporten er et resultat av en hovedoppgave gitt ved Høgskolen i Telemark avdeling for teknologiske fag, studieretning kybernetikk og industriell IT. Formålet med denne oppgaven var å finne en algoritme til å stille inn optimale parametre for PI-regulatorer. Det kreves kunnskap om avansert regulering og programmering for å få fullt utbytte av denne rapporten.

En spesiell takk rettes til David Di Ruscio for å ha vært behjelpelig med å svare på spørsmål underveis i oppgaven. Jeg vil også rette en takk til alle som var behjelpelige med små og store oppgaver under prosjektets gang.

Det er vedlagt en CD som inneholder noen utvalgte eksempler fra rapporten. Denne ligger i den bakerste permen.

---

*Olé Christian Skilbred*

Porsgrunn 24. mai 2005

# Kapittel 1

## Innledning

### 1.1 Litt om regulatorer

I dagens reguleringsystemer benyttes det mange forskjellige typer regulatorer. Det kan være alt fra en P-regulator til en optimal- eller prediktiv regulator. Regulatoren man velger må stå i stil til det man ønsker å regulere. Hvis en PI-regulator gjør jobben på en bra måte, er det ikke nødvendig å implementere en mer avansert regulator. Men en mer avansert regulator kan ha bedre egenskaper, og mer kontroll over prosessen enn en standard regulator. Dette skyldes i stor grad at pådraget genereres av en tilbakekopling av tilstandene. En optimalregulator har mange gode egenskaper. En regulator betegnes ”optimal” ved at man finner det pådraget som minimaliserer et kriterie. Et slikt diskret kriterie kan se slik ut på standardform

$$J = \sum_{k=0}^{\infty} (x_k^T Q x_k + u_k^T P u_k) \quad (1.1.1)$$

Der  $x_k$  og  $u_k$  er henholdsvis tilstands- og pådragsvektor, og  $Q$  og  $P$  er vektmatriser. Disse er det opp til brukeren å velge på en hensiktsmessig måte. Det kan vises at det optimale pådraget vil være på formen  $u_k = Gx_k$ .

### 1.2 Oppgaven

I denne oppgaven skal det sees nærmere på design av desentraliserte og suboptimale regulatorer for diskrete systemer. Den tradisjonelle løsningen for å finne

$G$ -matrisen skal omskrives til et optimaliseringsproblem. Dette innebærer også en omskrivning av den berømte Ricatti-ligningen, se Di Ruscio (2004), som stadig dukker opp i avansert reguleringsteori. Regulatormatrisene skal også kunne ha en spesifisert struktur. Det skal videre sees på hvordan man kan la pådraget være en funksjon av  $y_k$ , det vil si  $u_k = G_y y_k$ . Det som oppnås ved dette er et ”optimalt” pådrag generert av  $y_k$ . Denne metoden skal utnyttes til å lage en syntese av PI-regulatorer. Det er ønskelig å se på både standard PI-regulatorer og PI-regulatorer på endringsform. De forskjellige regulatorene skal illustreres med eksempler. På denne måten kan optimalregulering knyttes sammen med de kommersielle regulatorene.

Det skal også diskuteres muligheten å bruke en systemidentifikasjonsmetode som for eksempel ”dsr” til å estimere den diskrete tilstandsrommodellen som er utgangspunkt for regulatordesignmetodene.

Dersom tiden tillater det skal de viktigste resultatene presenteres i en kort artikkel på engelsk. Dette gjøres i samarbeid med faglærer.

En lignende algoritme er beskrevet for kontinuerlige systemer. Dette er blitt gjort av Di Ruscio og Balchen (1990) i en artikkel.



# Kapittel 2

## Optimal løsning som et optimaliseringsproblem

### 2.1 Optimal løsning

Gitt kriteriet

$$J = \sum_{k=0}^{\infty} (x_k^T Q x_k + u_k^T P u_k) \quad (2.1.1)$$

Der  $x_k \in \mathbb{R}^n$  er tilstandsvektor,  $u_k \in \mathbb{R}^r$  er pådragsvektor,  $Q$  og  $P$  er vektmatriser, henholdsvis  $Q \in \mathbb{R}^{n \times n}$  og  $P \in \mathbb{R}^{r \times r}$ . Gitt er også modellen

$$x_{k+1} = Ax_k + Bu_k \quad (2.1.2)$$

$$y_k = Dx_k + Eu_k \quad (2.1.3)$$

Der  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times r}$ ,  $D \in \mathbb{R}^{m \times n}$ ,  $E \in \mathbb{R}^{m \times r}$  og  $y_k \in \mathbb{R}^m$  er utgangsvektor. Det optimale pådraget som kan brukes til å styre modellen gitt i ligningene (2.1.2) og (2.1.3) kan utledes fra dette, se Di Ruscio (2004). Det kan vises at pådraget blir av formen  $u_k = Gx_k$ . Der  $G$  blir som følger hvis  $E = 0$

$$G = -(P + B^T R B)^{-1} (B^T R A) \quad (2.1.4)$$

Er  $E \neq 0$  kan kriteriet se slik ut

$$J = \sum_{k=0}^{\infty} (y_k^T Q y_k + u_k^T P u_k) \quad (2.1.5)$$

og pådraget blir av formen

$$G = -(P + B^T R B + E^T Q E)^{-1} (E^T Q D + B^T R A) \quad (2.1.6)$$

med dimensjonen  $G \in \mathbb{R}^{r \times n}$ . Merk når  $y_k$  vektet i kriterie er  $Q \in \mathbb{R}^{m \times m}$

$R$  er løsningen av den diskrete Ricatti-ligningen.  $R$  ser slik ut med  $E = 0$

$$R = Q + A^T R A - A^T R B H B^T R A \quad (2.1.7)$$

$$H = (P + B^T R B)^{-1} \quad (2.1.8)$$

Og  $R$  ser slik ut med  $E \neq 0$

$$\begin{aligned} R &= D^T Q D - D^T Q E H E^T Q D - D^T Q E H B^T R A + A^T R A \\ &\quad - A^T R B H E^T Q D - A^T R B H B^T R A \end{aligned} \quad (2.1.9)$$

$$H = (P + B^T R B + E^T Q E)^{-1} \quad (2.1.10)$$

Denne ligningen kan skrives på forskjellige måter, avhengig av hva som er mest hensiktsmessig for problemet som skal løses. I denne sammenheng lønner det seg å skrive Ricatti-ligningen på Lyapunov-form. Dette er vist i kapittel 3 som omhandler Lyapunovs diskrete matrise ligning. I MatLab brukes funksjonen "dlqr" for å finne  $G$ . Det vil si i MatLab er  $G = -K$ . Det eneste forskjellen er at minustegnet som er foran ligningene (2.1.4) og (2.1.6) er satt på utsiden. Dette er en viktig ting å være klar over. All teori i denne rapporten omhandler  $G$ .

Istedenfor å løse dette problemet som er vist over, er det mulig å løse dette som et optimaliseringsproblem. Man benytter istedenfor optimaliseringsrutiner for å finne minimum av funksjonsverdien  $J$  gitt i ligningene (2.1.1) og (2.1.5). For at dette skal være mulig må man gjøre noen grep på problemet.

Minimumsverdien av kriteriet er

$$J = x_0^T R x_0 \quad (2.1.11)$$

Dette kan omskrives til

$$J = x_0^T R x_0 = tr(x_0 x_0^T R) \quad (2.1.12)$$

Der  $tr$  står for trace og er definert som følger

$$tr(A) = \sum_{k=1}^{\min(n,m)} a_{kk} \quad (2.1.13)$$

Man kan videre gjøre antakelser om  $x_0$ , og si at denne er stokastisk. Dette medfører følgende

$$E(x_0) = 0 \quad (2.1.14)$$

$$E(x_0 x_0^T) = I \quad (2.1.15)$$

Ligning (2.1.12) kan skrives om til

$$tr(\overbrace{x_0 x_0^T}^I R) = tr(R) \quad (2.1.16)$$

En ting å merke seg om dette er at Ricatti-ligningen løses uten hensyn til  $x_0$ . Man kan derfor tenke seg at  $x_0$  heller ikke bør ha noe å si for minimumsverdien til kriteriet, og derfor settes  $x_0 x_0^T = I$ .

Innfører nå en kolonnevektor  $\theta$  og en strukturmatrise  $F$  som har samme dimensjon som  $G$ . Strukturmatrisen  $F$  er definert som følger

$$f_{ij} = \begin{cases} 1 & \text{hvis } g_{ij} \neq 0 \\ 0 & \text{hvis } g_{ij} = 0 \end{cases} \quad (2.1.17)$$

Det er viktig å merke seg at  $F \times G = G$ . Der  $\times$  betegner element for element multiplikasjon. Sammenhengen mellom  $G$ ,  $\theta$  og  $F$  er gitt av funksjonen *css*. Der *css* står "column structure string"-operator. Den er definert som følger

$$\theta = css(G, F) \quad (2.1.18)$$

Det som *css* gjør er å sette hvert element i  $G$  over til  $\theta$ . Dette gjøres ved hjelp av  $F$ -matrisen. Det er vist et lite eksempel som illustrerer dette.

**Eksempel 2.1.1**

Det er gitt følgende matriser

$$G = \begin{bmatrix} 3 & 5 \\ 0 & 1 \end{bmatrix} \quad (2.1.19)$$

$$F = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \quad (2.1.20)$$

Ved bruk av  $\theta = \text{css}(G, F)$  vil dette gi følgende  $\theta$

$$\theta = \begin{bmatrix} 3 \\ 5 \\ 1 \end{bmatrix} \quad (2.1.21)$$

I MatLab ser funksjonen slik ut:

```
function theta = css(G,F)
[nu, nx] = size(F);
n = 0;
for i = 1:nx
    for j = 1:nu
        if F(j,i)~=0
            n = n + 1;
        end
    end
end

theta = zeros(n,1);
k = 1;
for i = 1:nx
    for j = 1:nu
        if F(j,i)~=0
            theta(k,1) = G(j,i);
            k = k + 1;
        end
    end
end
end
```

Innfører også en ny operator

$$G = css^{-1}(\theta, F) \quad (2.1.22)$$

Denne fungerer på samme måte, men setter elementene tilbake til  $G$  fra  $\theta$  ved hjelp av  $F$ -matrisen. Et enkelt eksempel er tatt med for å vise dette.

### Eksempel 2.1.2

Det er gitt følgende matriser

$$\theta = \begin{bmatrix} 9 \\ 8 \end{bmatrix} \quad (2.1.23)$$

$$F = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (2.1.24)$$

Ved bruk av  $G = css^{-1}(\theta, F)$  vil dette gi følgende  $G$

$$G = \begin{bmatrix} 9 & 0 \\ 0 & 8 \end{bmatrix} \quad (2.1.25)$$

I MatLab ser funksjonen slik ut:

```
function G = invcss(theta,F)
[nu, nx] = size(F);
G = zeros(nu,nx);
n = 1;
for i = 1:nx
    for j = 1:nu
        if F(j,i)~=0
            G(j,i)=theta(n,1);
            n = n + 1;
        end
    end
end
end
```

Ved å gjøre det på denne måten kan brukeren forme  $G$  på en slik måte som er ønskelig.

Det som da blir optimaliseringsproblemet er å finne minimum av  $J = tr(R)$  ved hjelp av  $\theta$ , som er spesifisert i henhold til  $G$  og  $F$ . Det er da løsningen av Ricatti-ligningen kommer inn i bildet. Det er to Ricatti-ligninger å ta hensyn til. Disse er skrevet på Lyapunov-form Den ene der  $E = 0$  og den andre der  $E \neq 0$ . Disse ligningene ser slik ut

$$R = (A + BG)^T R (A + BG) + (Q + G^T P G) \quad (2.1.26)$$

og

$$R = (A + BG)^T R (A + BG) + \tilde{Q} \quad (2.1.27)$$

$$\tilde{Q} = D^T Q D + D^T Q E G + G^T E^T Q D + G^T E^T Q E G + G^T P G \quad (2.1.28)$$

For å kunne løse Ricatti-ligningen på Lyapunov-form er det i MatLab en funksjon "dlyap" som gjør nettopp dette.

$$R = dlyap(A, Q) \quad (2.1.29)$$

$$A R A^T - R + Q = 0 \quad (2.1.30)$$

Ligning (2.1.30) løses ved hjelp funksjonskallet gitt i ligning (2.1.29). Det er viktig å få Lyapunovligningen på denne formen. Som vi ser av ligningene (2.1.26) og (2.1.27) krever dette at  $A + BG$  må transponeres.

Det er nå mulig å sette opp en algoritme for hvordan man kan beregne  $J(\theta_k)$  for et tidsskritt.

### Algoritme 2.1.1

1. Gitt en parameter vektor  $\theta_k$ . Finn  $G_k$  fra funksjonen

$$G_k = css^{-1}(\theta_k, F) \quad (2.1.31)$$

der  $F$  er definert etter den struktur som er ønsket.

2. Løs  $R_k$  fra gitt  $G_k$ , der  $R_k$  er løsningen av Lyapunov-ligningen. Ligning (2.1.26) løses hvis  $E = 0$  og ligning (2.1.27) dersom  $E \neq 0$ .

## 3. Bestem funksjonsverdien fra ligning

$$J_k = \text{tr}(R_k) \quad (2.1.32)$$

Dette er implementert i MatLab på følgende måte:

```
function J = theta_fun_old(theta,F,A,B,D,E,Q,P,flag)

Gy = invcss(theta,F);

if flag == 0
    G = Gy*D;
elseif flag == 1
    G = Gy;
end

A_tilde = (A+B*G);

if E == 0
    Q_tilde = G'*P*G + Q;
else
    Q_tilde = G'*P*G + D'*Q*D + D'*Q*E*G + G'*E'*Q*D + G'*E'*Q*E*G;
end

R = dlyap(A_tilde',Q_tilde);
J = trace(R);
```

Ønsker man integralvirkning i regulatoren, er det gjort noen små forandringer som tar seg av dette. Ved å lage en funksjon  $[G_1, G_2] = \text{css\_int}^{-1}(\theta, F_1, F_2)$ , der  $G_1$  er den delen som tar seg av P-delen i regulatoren, mens  $G_2$  er den delen som tar seg av integralvirkningen. For eksempel kan pådraget se slik ut:  $u_k = G_1 x_k + G_2 z_k$ , der  $z_{k+1} = z_k + r - y_k$ .  $F_1$  og  $F_2$  bestemmer strukturen til henholdsvis  $G_1$  og  $G_2$ . Denne funksjonen er implementert i MatLab på følgende måte:

```
function [G1, G2] = invcss_int(theta,F1,F2)
[nu_F1, nx_F1] = size(F1);
```

```
[nu_F2, ny_F2] = size(F2);
G1 = zeros(nu_F1,nx_F1);
G2 = zeros(nu_F2,ny_F2);
n = 1;
for i = 1:nx_F1
    for j = 1:nu_F1
        if F1(j,i)~=0
            G1(j,i)=theta(n,1);
            n = n + 1;
        end
    end
end

for i = 1:ny_F2
    for j = 1:nu_F2
        if F2(j,i)~=0
            G2(j,i)=theta(n,1);
            n = n + 1;
        end
    end
end
```

Algoritme (2.1.1) bestemmer funksjonsverdien kun for ett tidsskritt. Det som er ønskelig er å finne den  $\theta$  som faktisk minimaliserer  $J$ , nemlig  $\theta^*$ . For å gjøre dette trengs det en optimaliseringsrutine.

## 2.2 Optimaliseringsrutiner

I MatLab er det mange forskjellige optimaliseringsrutiner. De har ulike egenskaper og har mange forskjellige bruksområder. Når det gjelder dette optimaliseringsproblemet viser det seg at det er ulineært og multidimensjonalt. Det har blitt sett på to ulike optimaliseringsrutiner for å løse dette problemet. Disse er



- `fminsearch` - Multidimensjonal ulineær minimalisering uten begrensninger, med Nelder-Mead direkte søkemetode
- `fmincon` - Multidimensjonal ulineær minimalisering med begrensninger

I bruk ser disse rutinene slik ut

```
[X,FVAL,EXITFLAG,OUTPUT] = FMINSEARCH(FUN,X0,OPTIONS,P1,P2,...)
[X,FVAL,EXITFLAG,OUTPUT] = FMINCON(FUN,X0,A,B,Aeq,Beq,LB,UB,
NONLCON,OPTIONS,P1,P2,...)
```

Begge funksjonene har "FUN" som argument. Det er her man definerer funksjonen som skal minimaliseres, i dette tilfelle algoritme (2.1.1). Argumentet "X0" bestemmer hvor rutinen skal begynne å søke etter løsningen. Dette er et veldig sentralt moment for hvordan man skal få rutinen til å konvergere til en stabil løsning. Det er viktig å sette en "X0" som gjør at  $A + BG_0$  blir en stabil matrise. Det vil si at absoluttverdiene til egenverdiene ligger innenfor enhetssirkelen. Det er to ting til som optimaliseringsrutinene har felles. Det er "OPTIONS" og "P"-parameterne. I "OPTIONS" kan man definere blant annet hvor mange ganger rutinen skal iterere før den skal stoppe. Man kan tilpasse rutinen mer til problemet. Når det gjelder "P"-parameterne, er det de ekstra argumentene i funksjonen "FUN". Disse blir definert til slutt, og overført inn til algoritmen. "FMINCON"-rutinen har ytterligere et par argumenter som skal hjelpe til å innsnevre søkeområdet, og sette begrensninger på løsningen. Det er argumentene "A", "B", "Aeq", "Beq", "LB", "UB" og "NONLCON". Sammenhengen mellom disse er som følger

$$A\theta \leq B \quad (2.2.1)$$

$$A_{eq}\theta = B_{eq} \quad (2.2.2)$$

$$C(\theta) \leq 0 \quad (2.2.3)$$

$$C_{eq}(\theta) = 0 \quad (2.2.4)$$

$$L_B \leq \theta \leq U_B \quad (2.2.5)$$

Der ligningene (2.2.3) og (2.2.4) er definert i funksjonen "NONLCON". En ting som er viktig å huske på i forbindelse med "NONLCON" er at den skal ha like mange argumenter som "FUN".

Når det gjelder begrensningen gitt i ligning (2.2.3), er det som nevnt viktig å få en stabil løsning. Det er ulike ting som kan testes for å sikre at det lukkede systemet  $A + BG$  gir stabilitet. Disse er:

1.  $|eig(A + BG)| < 1$
2.  $eig(R) > 0$
3.  $det(R_{ii}) > 0, i = 1, \dots, n$

Punkt 1 gjør at absoluttverdiene til egenverdiene kommer innenfor enhetssirkelen. Punkt 2 tester på egenverdiene til  $R$ -matrisen. Punkt 3 er Sylvesters betingelser, som skal sikre at alle underdeterminantene til  $R$  er større enn 0.

Det holder å teste på en av disse for å sikre at  $R$  blir positiv definit. Ricattiligningen kan ha flere løsninger, men den som skal brukes må være positiv definit. Da  $\theta^*$  er funnet via  $R$ , kan  $G$  nå lages ved hjelp av  $G = css^{-1}(\theta^*, F)$ . Eller med integralvirkning  $\begin{bmatrix} G_1, & G_2 \end{bmatrix} = css\_int^{-1}(\theta^*, F_1, F_2)$ .

# Kapittel 3

## Lyapunovs diskrete matrise ligning

### 3.1 Utledning

Her følger en utledning av hvordan Ricatti-ligningen kan skrives som Lyapunovs diskrete matrise ligning. Først uten kryssledd i kriteriet, og deretter med kryssledd i kriteriet.

#### 3.1.1 Uten kryssledd

Det antas at en prosess kan beskrives av følgende lineære tidsinvariante diskrete tilstandsrommodell

$$x_{k+1} = Ax_k + Bu_k \quad (3.1.1)$$

$$y_k = Dx_k \quad (3.1.2)$$

Anta følgende kriterie

$$J = \sum_{k=0}^{\infty} (x_k^T Q x_k + u_k^T P u_k) \quad (3.1.3)$$

Det kan videre vises at ved å minimalisere kriteriet er det optimale pådrag gitt ved

$$u_k = Gx_k \quad (3.1.4)$$

der

$$G = -(P + B^T R B)^{-1} (B^T R A) \quad (3.1.5)$$

Ved å sette inn (3.1.4) inn i (3.1.3) gir dette følgende sammenheng

$$J = \sum_{k=0}^{\infty} (x_k^T Q x_k + x_k^T G^T P G x_k) \quad (3.1.6)$$

Dette kan skrives som

$$J = \sum_{k=0}^{\infty} (x_k^T (Q + G^T P G) x_k) \quad (3.1.7)$$

Ved å innsette optimalt pådrag (3.1.4) inn i (3.1.1) sees følgende sammenheng

$$x_{k+1} = (A + B G) x_k \quad (3.1.8)$$

Det er så ønskelig å finne  $x_k$ . Indeksen settes så et skritt tilbake

$$x_k = (A + B G) x_{k-1} \quad (3.1.9)$$

Nå er det mulig å kombinere (3.1.7) og (3.1.9), samtidig som  $k$  går fra 1 istedenfor 0.

$$J = \sum_{k=1}^{\infty} (((A + B G) x_{k-1})^T (Q + G^T P G) ((A + B G) x_{k-1})) \quad (3.1.10)$$

Som igjen kan skrives som

$$J = \sum_{k=1}^{\infty} (x_{k-1}^T (A + B G)^T (Q + G^T P G) ((A + B G) x_{k-1})) \quad (3.1.11)$$

Den diskrete observerbarhetsgramianen er gitt ved

$$W_o = \sum_{k=1}^{\infty} \tilde{A}^{(k-1)T} \tilde{D}^T \tilde{D} \tilde{A}^{(k-1)} \in \mathbb{R}^{n \times n} \quad (3.1.12)$$

Dersom  $\tilde{A}$  er en stabil matrise og  $N \rightarrow \infty$  kan (3.1.12) skrives som Lyapunovs diskrete matriseligning

$$\tilde{A}^T W_o \tilde{A} - W_o = -\tilde{D}^T \tilde{D} \quad (3.1.13)$$

$R$  defineres nå fra kriteriet i ligning (3.1.11)

$$J = \sum_{k=1}^{\infty} (x_{k-1}^T \overbrace{(A + B G)^T (Q + G^T P G) (A + B G)}^R x_{k-1}) \quad (3.1.14)$$

Ved å sammenligne (3.1.12) og (3.1.14) finnes følgende sammenhenger

$$\tilde{A} = A + BG \quad (3.1.15)$$

$$\tilde{D}^T \tilde{D} = Q + G^T P G \quad (3.1.16)$$

$$W_o = R \quad (3.1.17)$$

Dette settes så inn i ligning (3.1.13)

$$(A + BG)^T R (A + BG) - R = -(Q + G^T P G) \quad (3.1.18)$$

Ved å flytte over ledd gir dette Lyapunovs diskrete matrise ligning

$$R = (A + BG)^T R (A + BG) + (Q + G^T P G) \quad (3.1.19)$$

Man kan også velge å vekte  $y_k$ , da blir  $Q \rightarrow D^T Q D$  og  $Q \in \mathbb{R}^{m \times m}$

### 3.1.2 Med kryssledd

Det antas at en prosess kan beskrives av følgende lineære tidsinvariante diskrete tilstandsrommodell

$$x_{k+1} = Ax_k + Bu_k \quad (3.1.20)$$

$$y_k = Dx_k + Eu_k \quad (3.1.21)$$

Anta følgende kriterie

$$J = \sum_{k=0}^{\infty} (y_k^T Q y_k + u_k^T P u_k) \quad (3.1.22)$$

Det kan videre vises at ved å minimalisere kriteriet er det optimale pådrag gitt ved

$$u_k = G x_k \quad (3.1.23)$$

der

$$G = -(P + B^T R B + E^T Q E)^{-1} (E^T Q D + B^T R A) \quad (3.1.24)$$

Ved å sette inn (3.1.21) inn i (3.1.22) gir dette følgende sammenheng

$$J = \sum_{k=0}^{\infty} ((Dx_k + Eu_k)^T Q (Dx_k + Eu_k) + u_k^T P u_k) \quad (3.1.25)$$

Dette kan skrives som

$$J = \sum_{k=0}^{\infty} (x_k^T D^T Q D x_k + x_k^T D^T Q E u_k + u_k^T E^T Q D x_k + u_k^T E^T Q E u_k + u_k^T P u_k) \quad (3.1.26)$$

Ved å kombinere (3.1.23) og (3.1.26) fåes følgende ligning

$$J = \sum_{k=0}^{\infty} ((x_k^T D^T Q D x_k + x_k^T D^T Q E G x_k + x_k^T G^T E^T Q D x_k + x_k^T G^T E^T Q E G x_k + x_k^T G^T P G x_k)) \quad (3.1.27)$$

Dette kan skrives som

$$J = \sum_{k=0}^{\infty} (x_k^T (D^T Q D + D^T Q E G + G^T E^T Q D + G^T E^T Q E G + G^T P G) x_k) \quad (3.1.28)$$

Ved å innsette optimalt pådrag (3.1.23) inn i (3.1.20) sees følgende sammenheng

$$x_{k+1} = (A + BG)x_k \quad (3.1.29)$$

Det er ønskelig å finne  $x_k$ . Indeksen settes så et skritt tilbake

$$x_k = (A + BG)x_{k-1} \quad (3.1.30)$$

Nå er det mulig å kombinere (3.1.28) og (3.1.30), samtidig som  $k$  går fra 1 istedenfor 0.

$$J = \sum_{k=1}^{\infty} ((A + BG)x_{k-1})^T (D^T Q D + D^T Q E G + G^T E^T Q D + G^T E^T Q E G + G^T P G) ((A + BG)x_{k-1}) \quad (3.1.31)$$

Som igjen kan skrives som

$$J = \sum_{k=1}^{\infty} (x_{k-1}^T (A + BG)^T (D^T Q D + D^T Q E G + G^T E^T Q D + G^T E^T Q E G + G^T P G) (A + BG) x_{k-1}) \quad (3.1.32)$$

Den diskrete observerbarhetsgramianen er gitt ved

$$W_o = \sum_{k=1}^{\infty} \tilde{A}^{(k-1)T} \tilde{D}^T \tilde{D} \tilde{A}^{(k-1)} \in \mathbb{R}^{n \times n} \quad (3.1.33)$$

Dersom  $\tilde{A}$  er en stabil matrise og  $N \rightarrow \infty$  kan (3.1.33) skrives som Lyapunovs diskrete matriseligning

$$\tilde{A}^T W_o \tilde{A} - W_o = -\tilde{D}^T \tilde{D} \quad (3.1.34)$$

$R$  defineres nå fra kriteriet i ligning (3.1.32)

$$\begin{aligned} R &= (A + BG)^T (D^T QD + D^T QEG + G^T E^T QD \\ &+ G^T E^T QEG + G^T PG)(A + BG) \end{aligned} \quad (3.1.35)$$

Ved å sammenligne (3.1.32) og (3.1.33) finnes følgende sammenhenger

$$\tilde{A} = A + BG \quad (3.1.36)$$

$$\tilde{D}^T \tilde{D} = D^T QD + D^T QEG + G^T E^T QD + G^T E^T QEG + G^T PG \quad (3.1.37)$$

$$W_o = R \quad (3.1.38)$$

Dette settes inn i ligning (3.1.34)

$$(A + BG)^T R(A + BG) - R = -\tilde{Q} \quad (3.1.39)$$

$$\tilde{Q} = D^T QD + D^T QEG + G^T E^T QD + G^T E^T QEG + G^T PG \quad (3.1.40)$$

Ved å flytte over ledd gir dette Lyapunovs diskrete matrise ligning

$$R = (A + BG)^T R(A + BG) + \tilde{Q} \quad (3.1.41)$$

Ved å sette  $E = 0$  og  $D = I$  kommer man frem til samme svar som i ligning (3.1.19).

# Kapittel 4

## Tilbakekopling fra $y_k$

### 4.1 Utledning

Det er mulig å la  $G \rightarrow G_y$  hvis man ønsker tilbakekopling fra utgangen  $y_k$ . Det er også tenkelig at måleligningen inneholder  $Eu_k$ -ledd. Dette fører til litt mer komplisert utregning av  $u_k$ . Dette er vist under, både med og uten integralvirkning.

#### 4.1.1 Uten integralvirkning

$y_k$  er gitt av den lineære ligningen

$$y_k = Dx_k + Eu_k \quad (4.1.1)$$

slik at

$$u_k = G_y y_k = G_y(Dx_k + Eu_k) \quad (4.1.2)$$

Denne må løses med hensyn på  $u_k$ . Dette vil gi følgende ligning

$$u_k = (I - G_y E)^{-1} G_y D x_k \quad (4.1.3)$$

Er  $E = 0$  vil dette gi ligningen

$$u_k = G_y D x_k \quad (4.1.4)$$

Merk at  $G_y \in \mathbb{R}^{r \times m}$ . Ved å sette ligning (4.1.3) på standard form

$$u_k = \overbrace{(I - G_y E)^{-1} G_y D}^{\tilde{G}} x_k = \tilde{G} x_k \quad (4.1.5)$$



kan dette utnyttes i utledningen av Riccati-ligningen på Lyapunov-form. For at dette pådraget skal eksistere må  $G_y E \neq I$ . Ellers vil dette medføre at  $u_k = 0$ . Riccati-ligningen er gitt av følgende ligning

$$(A + BG)^T R(A + BG) - R = -\tilde{Q} \quad (4.1.6)$$

$$\tilde{Q} = D^T QD + D^T QEG + G^T E^T QD + G^T E^T QEG + G^T PG \quad (4.1.7)$$

Ved å sette  $G = \tilde{G}$  gir dette ligningen

$$(A + B\tilde{G})^T R(A + B\tilde{G}) - R = -\tilde{Q} \quad (4.1.8)$$

$$\tilde{Q} = D^T QD + D^T QE\tilde{G} + \tilde{G}^T E^T QD + \tilde{G}^T E^T QE\tilde{G} + \tilde{G}^T P\tilde{G} \quad (4.1.9)$$

Denne ligningen vil være den som må løses, og der  $\tilde{G}$  er spesifisert i ligning (4.1.5).

I ligning (4.1.2) kan det tenkes at man ønsker referanse. En ting å merke seg er at referansen  $r$  må settes til 0 under syntesen. Dette gjøres for å unngå et foroverkoplingsignal  $h$ , som fremkommer under utledningen av det optimale pådraget, se Di Ruscio (2004).  $r$  brukes videre i regulatoren etter syntesen. Det vil si følgende

$$u_k = G_y(y_k - r) = G_y((Dx_k + Eu_k) - r) \quad (4.1.10)$$

eller

$$u_k = G_y(r - y_k) = G_y(r - (Dx_k + Eu_k)) \quad (4.1.11)$$

Dette vil gi følgende ligninger med  $r = 0$

$$u_k = (I - G_y E)^{-1} G_y D x_k \quad (4.1.12)$$

eller

$$u_k = (I + G_y E)^{-1} (-G_y D) x_k \quad (4.1.13)$$

er  $E = 0$  gir dette da følgende ligninger

$$u_k = G_y D x_k \quad (4.1.14)$$

eller

$$u_k = -G_y D x_k \quad (4.1.15)$$

Når det gjelder stabilitet er det viktig at  $A + B\tilde{G}$  har egenverdier som ligger inne i enhetssirkelen. Det vil si  $|eig(A + B\tilde{G})| < 1$ .

### 4.1.2 Med integralvirkning

Hvis man ønsker integralvirkning, det vil si at pådraget er av formen

$$u_k = \hat{G} \begin{bmatrix} y_k \\ z_k \end{bmatrix} \quad (4.1.16)$$

der

$$\hat{G} = \begin{bmatrix} G_1 & G_2 \end{bmatrix} \quad (4.1.17)$$

Her må man på forhånd ha augmentert modellen slik at den passer med pådraget. Merk at  $G_1$  og  $G_2 \in \mathbb{R}^{r \times m}$ . Ved å bruke ligning (4.1.16) og (4.1.17) blir pådraget av følgende form

$$u_k = G_1 y_k + G_2 z_k \quad (4.1.18)$$

som ved bruk av ligning (4.1.1) blir til

$$u_k = G_1 \overbrace{(Dx_k + Eu_k)}^{y_k} + G_2 z_k \quad (4.1.19)$$

Ligning (4.1.19) løses med hensyn på  $u_k$

$$u_k = (I - G_1 E)^{-1} \begin{bmatrix} G_1 D & G_2 \end{bmatrix} \tilde{x}_k \quad (4.1.20)$$

der

$$\tilde{x}_k = \begin{bmatrix} x_k \\ z_k \end{bmatrix} \quad (4.1.21)$$

For at dette pådraget skal eksistere må  $G_1 E \neq I$ . Ellers vil dette medføre at  $u_k = 0$ . Viser det seg at  $E = 0$  vil ligningen bli

$$u_k = \begin{bmatrix} G_1 D & G_2 \end{bmatrix} \tilde{x}_k \quad (4.1.22)$$

Ved å sette ligning (4.1.20) på standard form, det vil si

$$u_k = \overbrace{(I - G_1 E)^{-1} \begin{bmatrix} G_1 D & G_2 \end{bmatrix}}^{\tilde{G}} \tilde{x}_k = \tilde{G} \tilde{x}_k \quad (4.1.23)$$

kan dette utnyttes i utledningen av Riccati-ligningen på Lyapunov-form. Denne er gitt av følgende ligning

$$(A + BG)^T R (A + BG) - R = -\tilde{Q} \quad (4.1.24)$$

$$\tilde{Q} = D^T Q D + D^T Q E G + G^T E^T Q D + G^T E^T Q E G + G^T P G \quad (4.1.25)$$

Ved å sette  $G = \tilde{G}$  gir dette ligningen

$$(\tilde{A} + \tilde{B}\tilde{G})^T R(\tilde{A} + \tilde{B}\tilde{G}) - R = -\tilde{Q} \quad (4.1.26)$$

$$\tilde{Q} = \tilde{D}^T Q \tilde{D} + \tilde{D}^T Q \tilde{E} \tilde{G} + \tilde{G}^T \tilde{E}^T Q \tilde{D} + \tilde{G}^T \tilde{E}^T Q \tilde{E} \tilde{G} + \tilde{G}^T P \tilde{G} \quad (4.1.27)$$

Denne ligningen vil være den som må løses, og der  $\tilde{G}$  er spesifisert i ligning (4.1.23). Matrisene i modellen  $\tilde{A}, \tilde{B}, \tilde{D}, \tilde{E}$  og vektmatrisene  $Q, P$  er her tilpasset med augmentering.

### 4.1.3 Ulike typer augmentering

Det finnes ulike typer augmentering. En forskjell som er viktig å være klar over om man utvider tilstandsrommodellen med

$$z_{k+1} = z_k + y_k - r \quad (4.1.28)$$

eller

$$z_{k+1} = z_k + r - y_k \quad (4.1.29)$$

Ved bruk av ligning (4.1.28) vil regulatoren man kjører mot prosessen bli av formen

$$u_k = G_1(y_k - r) + G_2 z_k \quad (4.1.30)$$

$$z_{k+1} = z_k + y_k - r \quad (4.1.31)$$

og ved bruk av (4.1.29) vil regulatoren bli

$$u_k = G_1(r - y_k) + G_2 z_k \quad (4.1.32)$$

$$z_{k+1} = z_k + r - y_k \quad (4.1.33)$$

Har man modellen

$$x_{k+1} = Ax_k + Bu_k \quad (4.1.34)$$

$$y_k = Dx_k + Eu_k \quad (4.1.35)$$

og utvider denne med ligning (4.1.28) og setter  $r = 0$  vil man få denne augmenterte modellen

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \overbrace{\begin{bmatrix} A & 0 \\ D & I \end{bmatrix}}^{\tilde{A}_+} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \overbrace{\begin{bmatrix} B \\ E \end{bmatrix}}^{\tilde{B}_+} u_k \quad (4.1.36)$$

vil pådraget som er med i syntesen av regulatoren, bli som vist over i ligning (4.1.20), nemlig

$$u_k = (I - G_1 E)^{-1} \begin{bmatrix} G_1 D & G_2 \end{bmatrix} \tilde{x}_k \quad (4.1.37)$$

Utvider man (4.1.34) og (4.1.35) med (4.1.29) får man derimot en augmentert modell av formen

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \overbrace{\begin{bmatrix} A & 0 \\ -D & I \end{bmatrix}}^{\tilde{A}_-} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \overbrace{\begin{bmatrix} B \\ -E \end{bmatrix}}^{\tilde{B}_-} u_k \quad (4.1.38)$$

Her må man huske å inkludere minustegnet i regulatoren. Pådraget i syntesen vil bli

$$u_k = (I + G_1 E)^{-1} \begin{bmatrix} -G_1 D & G_2 \end{bmatrix} \tilde{x}_k \quad (4.1.39)$$

Dette er viktig å huske på når man skal analysere stabiliteten til det lukkede systemet. Det er viktig at det lukkede systemet har egenverdier som ligger innenfor enhetssirkelen. Det vil si  $|\text{eig}(\tilde{A} + \tilde{B}\tilde{G})| < 1$ . Det vil si fullt utskrevet

$$|\text{eig}(\tilde{A}_+ + \tilde{B}_+(I - G_1 E)^{-1} \begin{bmatrix} G_1 D & G_2 \end{bmatrix})| < 1 \quad (4.1.40)$$

eller

$$|\text{eig}(\tilde{A}_- + \tilde{B}_-(I + G_1 E)^{-1} \begin{bmatrix} -G_1 D & G_2 \end{bmatrix})| < 1 \quad (4.1.41)$$

Er  $E = 0$  vil ligningene bli

$$|\text{eig}(\tilde{A}_+ + \tilde{B} \begin{bmatrix} G_1 D & G_2 \end{bmatrix})| < 1 \quad (4.1.42)$$

eller

$$|\text{eig}(\tilde{A}_- + \tilde{B} \begin{bmatrix} -G_1 D & G_2 \end{bmatrix})| < 1 \quad (4.1.43)$$

der

$$\tilde{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (4.1.44)$$

En ting å være klar over er om man velger positiv eller negativ tilbakekopling fra  $y_k$  ikke spiller noen rolle for svaret. Om man velger positiv tilbakekopling og får svaret  $G_1 = \alpha$  og  $G_2 = \rho$ , vil man med negativ tilbakekopling få  $G_1 = -\alpha$  og  $G_2 = -\rho$ .

#### 4.1.4 Litt om vekting

Gitt kriterie

$$J = \sum_{k=0}^{\infty} (x_k^T Q x_k + u_k^T P u_k) \quad (4.1.45)$$

Sett nå at man ønsker integralvirkning i den resulterende regulatoren. Det kan dermed være ønskelig å kunne vekte også dette i kriterie, som blir seende slik ut

$$J = \sum_{k=0}^{\infty} (x_k^T Q x_k + z_k^T Q_z z_k + u_k^T P u_k) \quad (4.1.46)$$

For å få til dette må kriterie (4.1.46) settes på standard form. Ved å utvide tilstandsvektoren  $x_k$  til  $\tilde{x}_k$ , der  $\tilde{x}_k = \begin{bmatrix} x_k \\ z_k \end{bmatrix}$ . Kriterie (4.1.46) kan omskrives til

$$J = \sum_{k=0}^{\infty} (\tilde{x}_k^T \overbrace{\begin{bmatrix} Q & 0 \\ 0 & Q_z \end{bmatrix}}^{\tilde{Q}} \tilde{x}_k + u_k^T P u_k) = \sum_{k=0}^{\infty} (\tilde{x}_k^T \tilde{Q} \tilde{x}_k + u_k^T P u_k) \quad (4.1.47)$$

Man kan nå vekte alle tilstandene i prosessen. Det er også mulig å sette opp dette slik at utgangene blir vektet. Kriterie som betraktes er

$$J = \sum_{k=0}^{\infty} (y_k^T Q y_k + z_k^T Q_z z_k + u_k^T P u_k) \quad (4.1.48)$$

For å skrive om dette brukes det ligningen

$$y_k = D x_k + E u_k \quad (4.1.49)$$

Det er viktig å merke seg at det er  $\tilde{x}_k$  som er vektoren, derfor blir  $D \rightarrow \tilde{D}$ , der  $\tilde{D} = \begin{bmatrix} D & 0 \end{bmatrix}$ . Innsatt i kriterie (4.1.48) gir det

$$J = \sum_{k=0}^{\infty} (x_k^T D^T Q D x_k + z_k^T Q_z z_k + 2\tilde{x}_k^T \tilde{D}^T Q E u_k + u_k^T E^T Q E u_k + u_k^T P u_k) \quad (4.1.50)$$

Dette skrives videre på standardform

$$J = \sum_{k=0}^{\infty} (\tilde{x}_k^T \overbrace{\begin{bmatrix} D^T Q D & 0 \\ 0 & Q_z \end{bmatrix}}^{\tilde{Q}_y} \tilde{x}_k + 2\tilde{x}_k^T \tilde{D}^T Q E u_k + u_k^T E^T Q E u_k + u_k^T P u_k) \quad (4.1.51)$$

som gir

$$J = \sum_{k=0}^{\infty} (\tilde{x}_k^T \tilde{Q}_y \tilde{x}_k + 2\tilde{x}_k^T \tilde{D}^T Q E u_k + u_k^T E^T Q E u_k + u_k^T P u_k) \quad (4.1.52)$$

eller

$$J = \sum_{k=0}^{\infty} (\tilde{x}_k^T \tilde{Q}_y \tilde{x}_k + 2\tilde{x}_k^T \tilde{N} u_k + u_k^T \tilde{P} u_k) \quad (4.1.53)$$

der  $\tilde{N} = \tilde{D}^T Q E$  og  $\tilde{P} = E^T Q E + P$ . Er  $E = 0$  resulterer dette i følgende kriterie

$$J = \sum_{k=0}^{\infty} (\tilde{x}_k^T \tilde{Q}_y \tilde{x}_k + u_k^T P u_k) \quad (4.1.54)$$

Noen prosesser krever at man må vekte alle tilstandene, for at det skal være mulig å stabilisere den. Mens noen prosesser holder det å vekte utgangene. Dette må undersøkes på hver enkelt prosess.

# Kapittel 5

## Syntese av PI-regulatorer

### 5.1 Standard PI-regulatorer

Gitt systemet

$$x_{k+1} = Ax_k + Bu_k \quad (5.1.1)$$

$$y_k = Dx_k + Eu_k \quad (5.1.2)$$

Denne modellen augmenteres følgende

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ \pm D & I \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ \pm E \end{bmatrix} u_k \quad (5.1.3)$$

Ved bruk av optimalregulator med positiv tilbakekopling fra  $y_k$  og uten direkteledd  $Eu_k$  vil man få et pådrag av typen

$$u_k = \begin{bmatrix} G_1 D & G_2 \end{bmatrix} \tilde{x}_k \quad (5.1.4)$$

Merk at  $E$  er med i syntesen for regulatoren, men ikke med i selve implementeringen av den, det vil si  $E = 0$ . Selv om man bruker  $G = (I \pm G_1 E)^{-1} \begin{bmatrix} \mp G_1 D & G_2 \end{bmatrix}$  er det  $G_1$  og  $G_2$  som utgjør matrisene i regulatoren. Ved å skrive ut dette, her med positiv tilbakekopling som eksempel, gir dette

$$u_k = u_s + G_1(Dx_k - Dx_s - r) + G_2 z_k \quad (5.1.5)$$

$$u_k = \underbrace{u_s - G_1 Dx_s}_{\text{konstant}} + G_1 \underbrace{(y_k - r)}_{e_k} + G_2 z_k \quad (5.1.6)$$

Det er mulig å sløyfe konstantleddet først i ligningen fordi  $z_k$  vil kunne stille seg uavhengig av denne. Man kan derfor skrive pådraget på formen

$$u_k = G_1 e_k + G_2 z_k \quad (5.1.7)$$

der

$$z_{k+1} = z_k + e_k \quad (5.1.8)$$

$$(5.1.9)$$

og  $e_k$  er enten

$$e_k = r - y_k \quad (5.1.10)$$

eller

$$e_k = y_k - r \quad (5.1.11)$$

avhengig av hva slags augmentering man har. Nå kan man utnytte dette til å finne  $K_p$  og  $T_i$  i en PI-regulator.

En standard PI-regulator kan skrives på denne måten

$$u_k = K_p e_k + \frac{K_p}{T_i} z_k \quad (5.1.12)$$

$$z_{k+1} = z_k + e_k \quad (5.1.13)$$

Det er viktig at man er nøye med hva slags  $e_k$  man bruker. Slik at man bruker samme i PI-regulatoren, gitt i ligning (5.1.12), og i syntesen. Ved å sammenligne ligning (5.1.7) med ligning (5.1.12) gir dette

$$G_1 = K_p \quad (5.1.14)$$

$$G_2 = \frac{K_p}{T_i} \quad (5.1.15)$$

slik at

$$K_p = G_1 \quad (5.1.16)$$

$$T_i = \frac{G_1}{G_2} \quad (5.1.17)$$

Denne metoden kan brukes multivariabelt. Da kommer dette med parring inn i bilde. Det vil si hvilken inngang skal brukes til å styre hvilken utgang. Det kan



vises hvilket pådrag som er best til å styre en gitt utgang. Dette kan blant annet vises med en RGA-analyse (Relative Gain Array), se Di Ruscio (2002).

For å illustrere dette kan  $2 \times 2$ -system brukes, det vil si et system med 2 innganger og 2 utganger. Det er ønskelig å finne et pådrag

$$u_k = G_1 e_k + G_2 z_k \quad (5.1.18)$$

der man for eksempel velger følgende struktur

$$G_1 = \begin{bmatrix} k_{11} & 0 \\ 0 & k_{22} \end{bmatrix} \quad (5.1.19)$$

$$G_2 = \begin{bmatrix} z_{11} & 0 \\ 0 & z_{22} \end{bmatrix} \quad (5.1.20)$$

Her betyr det at  $u_1$  brukes til å styre  $y_1$ , og  $u_2$  brukes til å styre  $y_2$ . Ved å sette inn disse ligningen i ligning (5.1.18) gir det

$$u_k = \begin{bmatrix} k_{11} & 0 \\ 0 & k_{22} \end{bmatrix} e_k + \begin{bmatrix} z_{11} & 0 \\ 0 & z_{22} \end{bmatrix} z_k \quad (5.1.21)$$

der  $u_k$ ,  $e_k$  og  $z_k$  er  $\in \mathbb{R}^{2 \times 1}$ . Ved å skrive dette fullt ut gir dette

$$u_k^1 = k_{11} e_k^1 + z_{11} z_k^1 \quad (5.1.22)$$

$$u_k^2 = k_{22} e_k^2 + z_{22} z_k^2 \quad (5.1.23)$$

Ved å velge en slik struktur kan  $K_p$  og  $T_i$  regnes ut for to forskjellige PI-regulatorer. Det vil si

$$K_p^1 = k_{11} \quad (5.1.24)$$

$$T_i^1 = \frac{k_{11}}{z_{11}} \quad (5.1.25)$$

$$K_p^2 = k_{22} \quad (5.1.26)$$

$$T_i^2 = \frac{k_{22}}{z_{22}} \quad (5.1.27)$$

Dette vil bli tilsvarende dersom det velges en annen struktur slik at  $u_1$  brukes til å styre  $y_2$ , og  $u_2$  brukes til å styre  $y_1$ . Matrisene  $G_1$  og  $G_2$  vil se slik ut

$$G_1 = \begin{bmatrix} 0 & k_{12} \\ k_{21} & 0 \end{bmatrix} \quad (5.1.28)$$

$$G_2 = \begin{bmatrix} 0 & z_{12} \\ z_{21} & 0 \end{bmatrix} \quad (5.1.29)$$

Dette kan gjøres også for flere variable. Men man må tenke gjennom hvilke paringer som er best for systemet.

## 5.2 PI-regulator på endringsform

For å lese mer om regulatorer på endringsform se Di Ruscio (2003b) for et forelesningsnotat. Gitt systemet

$$x_{k+1} = Ax_k + Bu_k + v \quad (5.2.1)$$

$$y_k = Dx_k + w \quad (5.2.2)$$

Ved å trekke fra ligningen over med ett tidskritt bakover gir dette

$$x_{k+1} - x_k = A(x_k - x_{k-1}) + B(u_k - u_{k-1}) + v - v \quad (5.2.3)$$

$$y_k - y_{k-1} = D(x_k - x_{k-1}) + w - w \quad (5.2.4)$$

Dette vil gi følgende modell

$$\Delta x_{k+1} = A\Delta x_k + B\Delta u_k \quad (5.2.5)$$

$$\Delta y_k = D\Delta x_k \quad (5.2.6)$$

der  $\Delta x_k = x_k - x_{k-1}$ ,  $\Delta u_k = u_k - u_{k-1}$  og  $\Delta y_k = y_k - y_{k-1}$ . Ved å flytte over  $y_{k-1}$  og trekke fra referansen  $r$  på begge sider i ligning (5.2.6), gir følgende ligninger

$$\Delta x_{k+1} = A\Delta x_k + B\Delta u_k \quad (5.2.7)$$

$$y_k - r = D\Delta x_k + y_{k-1} - r \quad (5.2.8)$$

Dette utnyttes videre ved å sette modellen på augmentert form

$$\begin{bmatrix} \Delta x_{k+1} \\ y_k - r \end{bmatrix} = \begin{bmatrix} A & 0 \\ D & I \end{bmatrix} \begin{bmatrix} \Delta x_k \\ y_{k-1} - r \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \Delta u_k \quad (5.2.9)$$

$$\begin{bmatrix} y_k - r \end{bmatrix} = \begin{bmatrix} D & I \end{bmatrix} \begin{bmatrix} \Delta x_k \\ y_{k-1} - r \end{bmatrix} \quad (5.2.10)$$

Ved bruk av en optimalregulator vil dette generere pådraget

$$\Delta u_k = \begin{bmatrix} G_1 & G_2 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ y_{k-1} - r \end{bmatrix} \quad (5.2.11)$$

som gir fullt utskrevet

$$u_k = u_{k-1} + G_1(x_k - x_{k-1}) + G_2(y_{k-1} - r) \quad (5.2.12)$$

Ved å bruke teknikken med tilbakekopling fra  $y_k$ , det vil si velge  $G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$  i syntesen, kan man skrive denne regulatoren på følgende form

$$u_k = u_{k-1} + G_1 D(x_k - x_{k-1}) + G_2(y_{k-1} - r) \quad (5.2.13)$$

$$u_k = u_{k-1} + G_1(y_k - y_{k-1}) + G_2(y_{k-1} - r) \quad (5.2.14)$$

En vanlig PI-regulator på endringsform kan skrives

$$u_k - u_{k-1} = K_p(r - y_k) + \frac{K_p}{T_i} z_k - (K_p(r - y_{k-1}) + \frac{K_p}{T_i} z_{k-1}) \quad (5.2.15)$$

$z_k$  kan formuleres på ulike måter, det vil si med og uten samplingstid. Her brukes det uten samplingstid slik at

$$z_k - z_{k-1} = r - y_{k-1} \quad (5.2.16)$$

Ordner på ligning (5.2.15) og innsetter ligning (5.2.16), det gir

$$u_k = u_{k-1} - K_p(y_k - y_{k-1}) - \frac{K_p}{T_i}(y_{k-1} - r) \quad (5.2.17)$$

Ved å sammenligne ligning (5.2.14) med ligning (5.2.17) gir dette følgende sammenhenger

$$G_1 = -K_p \quad (5.2.18)$$

$$G_2 = -\frac{K_p}{T_i} \quad (5.2.19)$$

slik at

$$K_p = -G_1 \quad (5.2.20)$$

$$T_i = \frac{G_1}{G_2} \quad (5.2.21)$$

Også for denne regulatoren kan dette gjøres multivariabelt. Her vises et eksempel

$$G_1 = \begin{bmatrix} k_{11} & 0 \\ 0 & k_{22} \end{bmatrix} \quad (5.2.22)$$

$$G_2 = \begin{bmatrix} z_{11} & 0 \\ 0 & z_{22} \end{bmatrix} \quad (5.2.23)$$

Her betyr det at  $u_1$  brukes til å styre  $y_1$ , og  $u_2$  brukes til å styre  $y_2$ . Ved å sette inn disse ligningene inn i ligning (5.2.14) gir det

$$u_k = u_{k-1} + \begin{bmatrix} k_{11} & 0 \\ 0 & k_{22} \end{bmatrix} \begin{bmatrix} y_k^1 - y_{k-1}^1 \\ y_k^2 - y_{k-1}^2 \end{bmatrix} + \begin{bmatrix} z_{11} & 0 \\ 0 & z_{22} \end{bmatrix} \begin{bmatrix} y_{k-1}^1 - r^1 \\ y_{k-1}^2 - r^2 \end{bmatrix} \quad (5.2.24)$$

der  $u_k \in \mathbb{R}^{2 \times 1}$ . Ved å skrive dette fullt ut gir dette

$$u_k^1 = u_{k-1}^1 + k_{11}(y_k^1 - y_{k-1}^1) + z_{11}(y_{k-1}^1 - r^1) \quad (5.2.25)$$

$$u_k^2 = u_{k-1}^2 + k_{22}(y_k^2 - y_{k-1}^2) + z_{22}(y_{k-1}^2 - r^2) \quad (5.2.26)$$

Ved å velge en slik struktur kan  $K_p$  og  $T_i$  regnes ut for to forskjellige PI-regulatorer. Det vil si

$$K_p^1 = -k_{11} \quad (5.2.27)$$

$$T_i^1 = \frac{k_{11}}{z_{11}} \quad (5.2.28)$$

$$K_p^2 = -k_{22} \quad (5.2.29)$$

$$T_i^2 = \frac{k_{22}}{z_{22}} \quad (5.2.30)$$

En fordel med en regulator på endringsform er at man slipper å ta hensyn til eventuelle stasjonærverdier. Disse blir utelatt nettopp på grunn av endringsformen. Selv om det er lett å simulere en modell med stasjonærverdier, er det ikke så lett å gjøre dette med en virkelig prosess. Da må nemlig stasjonærverdiene stemme meget godt overens med prosessen. Og det er ikke alltid tilfelle, da mange prosesser er meget ulineære.

# Kapittel 6

## Metoder for stabilisering av prosesser

### 6.1 To løsninger

Noen prosesser kan i utgangspunktet være ustabile. Eller ved å augmentere en prosess med integralvirkning, gjør sitt til at prosessen får egenverdier som ligger på enhetssirkelen. Dette kan gjøre det vanskelig å velge en  $G_0$  som stabiliserer prosessen. En diskret prosess er stabil dersom

$$|eig(A + BG)| < 1 \quad (6.1.1)$$

Det er sett på to metoder som kan være med å stabilisere prosesser.

#### 6.1.1 Innføring av $\beta$

Betrakter følgende kriterie

$$J = \sum_{k=1}^{\infty} (x_{k-1}^T (A + BG)^T \tilde{Q} (A + BG) x_{k-1}) \quad (6.1.2)$$

Fra dette settes det så inn en hjelpestørrelse  $\beta$ , der  $\beta > 0$  og et skalart tall. Dette gir

$$J = \sum_{k=1}^{\infty} (x_{k-1}^T (A + BG - \beta I)^T \tilde{Q} (A + BG - \beta I) x_{k-1}) \quad (6.1.3)$$

Dette gir følgende Lyapunov-ligning

$$R = (A + BG - \beta I)^T R (A + BG - \beta I) + \tilde{Q} \quad (6.1.4)$$

der  $\tilde{Q}$  ser slik ut med  $E = 0$

$$\tilde{Q} = D^T Q D + G^T P G \quad (6.1.5)$$

eller med  $E \neq 0$

$$\tilde{Q} = D^T Q D + D^T Q E G + G^T E^T Q D + G^T E^T Q E G + G^T P G \quad (6.1.6)$$

Systemet er nå stabilt dersom

$$|\text{eig}(A + BG - \beta I)| < 1 \quad (6.1.7)$$

Der  $G$  kan formes som vist i kapittel 4. Det som er meningen med dette er å anta en stor nok  $\beta$  som gjør at systemet blir stabilt. Deretter la  $\beta \rightarrow 0$ . Dette gjør det mulig å velge  $G_0 = 0$  som startverdi, samtidig som det lukkede systemet er stabilt. En lignende algoritme er laget for kontinuerlige systemer, se Di Ruscio (1993). Stoffet er hentet derfra, men det er gjort noen modifikasjoner siden dette omhandler diskrete systemer.

### Algoritme 6.1.1

1. *Bestem*

$$\bar{x}_k = \max(|\text{eig}(A + BG_k)|) \quad (6.1.8)$$

og initialiser  $\beta_k$

$$\beta_k > \bar{x}_k - 1 \text{ hvis } \bar{x}_k \geq 1 \quad (6.1.9)$$

$$\beta_k = 0 \text{ hvis } \bar{x}_k < 1 \quad (6.1.10)$$

2. *Initialiser  $G_k$  og løs optimaliseringsproblemet*

$$R = (A + BG_k - \beta_k I)^T R (A + BG_k - \beta_k I) + \tilde{Q} \quad (6.1.11)$$

*slik at*

$$\min_G J = J(R) \quad (6.1.12)$$

der  $\tilde{Q}$  er som i ligning (6.1.5) hvis  $E = 0$  og ligning (6.1.6) dersom  $E \neq 0$

3. Sjekk for konvergens. Dersom  $\beta_k = 0$  er optimaliseringen ferdig. Hvis ikke, definer  $G_{k+1} = G_k$ ,  $k = k + 1$  og gå til første punkt.

$\beta_k$ -parameteren bør velges på følgende måte

$$\beta_k = \bar{x}_k - 1 + d\beta \quad (6.1.13)$$

der  $d\beta > 0$ . Det modifiserte systemet  $A + BG_k - \beta_k I$  vil ha egenverdier innenfor enhets sirkelen dersom  $d\beta$  velges til en passende størrelse.  $d\beta$  bør videre velges slik at  $\bar{x}_{k+1} - \bar{x}_k < 0$  og  $\beta_{k+1} < \beta_k$ . Dermed vil  $\beta_k$  til slutt bli 0 og løsningen konvergere.

### 6.1.2 Innføring av $\delta$

Betrakter følgende kriterie, se Anderson og Moore (1971),

$$J = \sum_{k=1}^{\infty} \lambda^{-2k} (x_k^T Q x_k + u_{k-1}^T P u_{k-1}) \quad (6.1.14)$$

der  $\lambda$  ligger i rommet  $0 < \lambda \leq 1$ . Med modellen

$$x_{k+1} = Ax_k + Bu_k \quad (6.1.15)$$

vil  $A$ , i følge Anderson og Moore (1971), gå til  $A \rightarrow \lambda^{-1}A$  i Ricatti-ligningen, og alle egenverdiene i det lukkede systemet vil være mindre en  $\lambda$ .

Ved å manipulere på kriterie (6.1.14), gir det følgende

$$J = \sum_{k=1}^{\infty} \delta^{2k} (x_k^T Q x_k + u_k^T P u_k) \quad (6.1.16)$$

det vil si følgende transformasjon

$$\lambda^{-2k} \rightarrow \delta^{2k} \quad (6.1.17)$$

der  $\delta$  ligger i rommet  $0 < \delta \leq 1$ . Dette vil gi at  $A \rightarrow \delta A$ . Ved å utnytte det i utledningen i Ricatti-ligningen på Lyapunov-form vil dette gi følgende ligning

$$R = (\delta A + BG)^T R (\delta A + BG) + \tilde{Q} \quad (6.1.18)$$

der

$$\tilde{Q} = D^T Q D + G^T P G \quad (6.1.19)$$

$$\tilde{Q} = D^T Q D + D^T Q E G + G^T E^T Q D + G^T E^T Q E G + G^T P G \quad (6.1.20)$$

avhengig om  $E = 0$  eller  $E \neq 0$ .

Dette kan utnyttes i en stabiliseringsalgoritme. Ved å velge en  $\delta$  som gjør at det lukkede systemet  $(\delta A + BG)$  er stabilt, kan man finne en stabiliserende regulator med  $G_0 = 0$  som startverdi. Man kan deretter la  $\delta \rightarrow 1$  slik at den endelige regulatoren er uavhengig av  $\delta$ .  $G$  kan formes som vist i kapittel 4.

### Algoritme 6.1.2

1. Bestem

$$\bar{x}_k = \max(|\text{eig}(A + BG_k)|) \quad (6.1.21)$$

2. Initialiser  $\delta_k$  slik at  $|\text{eig}(\delta_k A + BG_k)| < 1$

(a) Ved første iterasjon gjøres dette

$$0 < \delta_k < 1 \text{ hvis } \bar{x}_k \geq 1 \quad (6.1.22)$$

$$\delta_k = 1 \text{ hvis } \bar{x}_k < 1 \quad (6.1.23)$$

(b) Ved alle andre iterasjoner gjøres dette

$$\delta_{k+1} = \delta_k + d\delta \quad (6.1.24)$$

3. Initialiser  $G_k$  og løs optimaliseringsproblemet

$$R = (\delta_k A + BG_k)^T R (\delta_k A + BG_k) + \tilde{Q} \quad (6.1.25)$$

slik at

$$\min_G J = J(R) \quad (6.1.26)$$

der  $\tilde{Q}$  er som i ligning (6.1.19) hvis  $E = 0$  og ligning (6.1.20) dersom  $E \neq 0$

4. Sjekk for konvergens. Dersom  $\bar{x}_k = \max(|\text{eig}(A + BG_k)|) < 1$  er  $G_0$  funnet. Optimaliser så med  $G_0$ . Hvis ikke, definer  $G_{k+1} = G_k$ ,  $k = k + 1$  og gå til punkt 2.

På denne måten vil  $\delta$  bevege seg mot 1 med farten  $d\delta$ . Ved å velge en passende  $d\delta$  vil løsningen holde seg stabil underveis. Til slutt vil man finne en  $G_0$  som kan brukes til å stabilisere prosessen.



# Kapittel 7

## Eksempler

I dette kapitlet er det vist mange forskjellige eksempler på teknikken som er utledet i de tidligere kapitlene. De viser hvordan regulatorene faktisk virker på ulike prosesser. For å kunne sammenligne ulike regulatorer typer på en prosess må man ha noe å sammenligne med. For hvert eksempel er det oppgitt noen faste punkter slik at dette skal være mulig. Disse er:

- $J(G) = \text{tr}(R)$
- $\gamma = \frac{\text{tr}(R)}{\text{tr}(R_{opt})} \geq 1$
- Hvordan  $G$  ser ut i syntesen
- $G$  sine verdier
- $|\text{eig}(A + BG)|$
- Vekting av  $Q$  og  $P$
- Eventuell augmentering

Disse verdiene er ment å si noe om hvor bra regulatoren er, sett ut fra hva som er optimalt, og hvor godt den stabiliserer prosessen. Det er plottet mange simuleringer og disse gir også et godt innblikk for hvor godt regulatorene fungerer.

Det er gitt en oversikt over de forskjellige modellene som er brukt i tabell (7.1.1).

Tabell 7.1.1: Oversikt over modeller

Modell	Tilstander	Pådrag	Utganger	Tillegg
SISO	1	1	1	A.1
AIRC	5	3	3	A.2
Dest. kol	8	2	2	A.3
Example 3	3	2	2	A.4
Vanntank	2	1	1	A.5
Unstable plant	3	2	2	A.6

## 7.1 Simuleringer

### 7.1.1 SISO-system med direktevirkning

For å illustrere et eksempel med direktevirkning, det vil si  $E \neq 0$ , har det blitt valgt ut et enkelt eksempel. Modellen er da

$$x_{k+1} = Ax_k + Bu_k \quad (7.1.1)$$

$$y_k = Dx_k + Eu_k \quad (7.1.2)$$

I syntesen er  $E$  tatt med, men i selve simuleringen av systemet er  $E = 0$ .

#### Eksempel 7.1.1 *Optimal regulator og regulator med tilbakekopling fra $y_k$*

*Det er her bare plottet hvordan regulatorene svinger seg inn mot 0. Se figur (7.1.1) for selve simuleringen. Her ser man at man får samme verdi på egenverdiene og  $J$ . Men man plukker ut  $G_y$  fra syntesen. Det er denne som representerer matrisen i regulatoren med tilbakekopling fra  $y_k$ . Det er ikke brukt noen augmentering.*

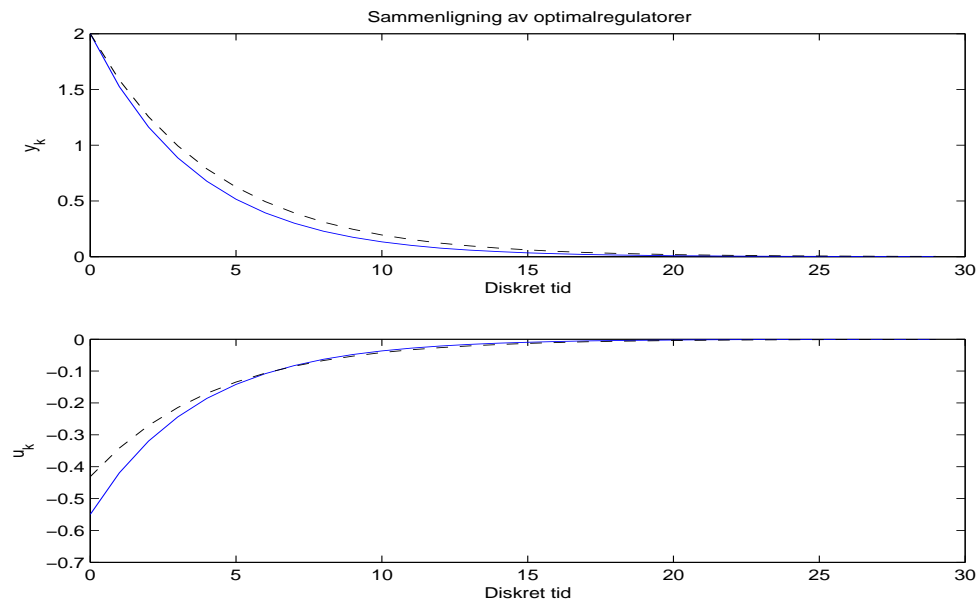
$$J(G) = 4,062$$

$$\gamma = 1$$

$$G = (I - G_y E)^{-1} G_y D$$

$$G_{opt} = -0.2748$$

$$G_y = -0.2156$$



**Figur 7.1.1:** Figuren viser optimalregulator i heltrukken blå, mot tilbakekopling fra  $y_k$  i strekete svart.

$$|eig(A + BG)| = [ 0.7626 ]$$

$Q = 1$  og  $P = 1$ , det vil si fullt utskrevet

$$\tilde{Q} = D^T Q D, \tilde{P} = P + E^T Q E \text{ og } \tilde{N} = D^T Q E$$

### Eksempel 7.1.2 *Optimalregulator og regulator med tilbakekopling fra $y_k$ med integralvirkning*

Her får man også samme verdi på egenverdiene og  $J$ . Men man plukker ut  $G_1$  og  $G_2$  fra syntesen. Det er disse som representerer matrisene i regulatoren med tilbakekopling fra  $y_k$ . Dette genereres ved kjøring av MatLab skript-filen **SISO.m**. Se figur (7.1.2) for selve simuleringen. Optimalregulatoren er simulert uten noen stasjonærverdier  $u_s$  eller  $x_s$  da disse ikke er oppgitt for modellen. Figurene genereres ved kjøring av MatLab skript-filen **simulering2.m**

$$J(G) = 433,9729$$

$$\gamma = 1$$

$$G = (I - G_1 E)^{-1} \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$

$$G_{opt} = \begin{bmatrix} -2,0496 & -0.3502 \end{bmatrix}$$

$$G_1 = -0,2156$$

$$G_2 = -0,1148$$

$$|eig(A + BG)| = \begin{bmatrix} 0.5131 \\ 0.7123 \end{bmatrix}$$

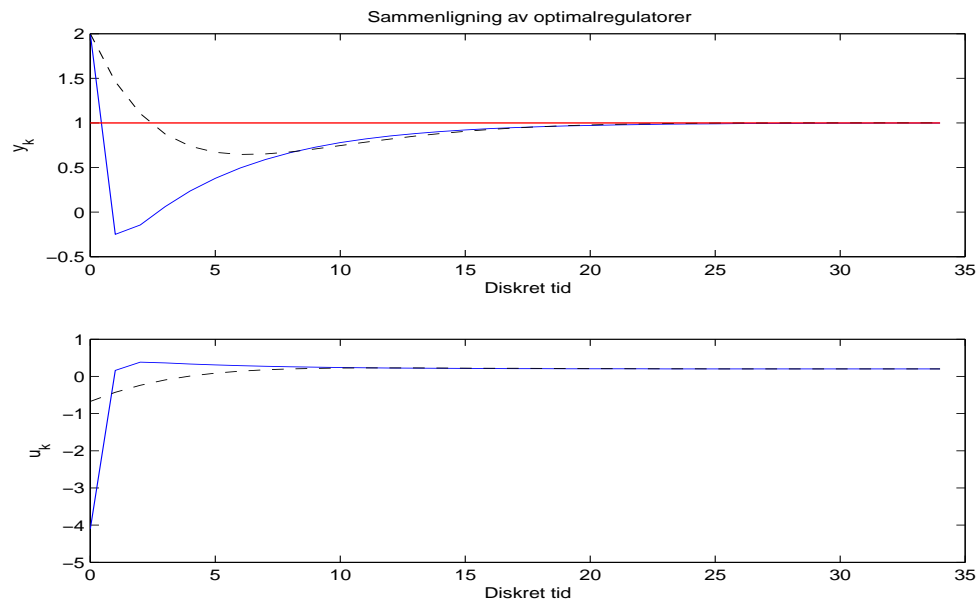
Vektingen er er her:

$Q = 10$ ,  $Q_z = 5$  og  $P = 1$  Det vil si fullt utskrevet

$$\tilde{Q} = \begin{bmatrix} D^T Q D & 0 \\ 0 & Q_z \end{bmatrix}, \tilde{P} = P + E^T Q E \text{ og } \tilde{N} = \tilde{D}^T Q E$$

Det er her brukt følgende augmentering:

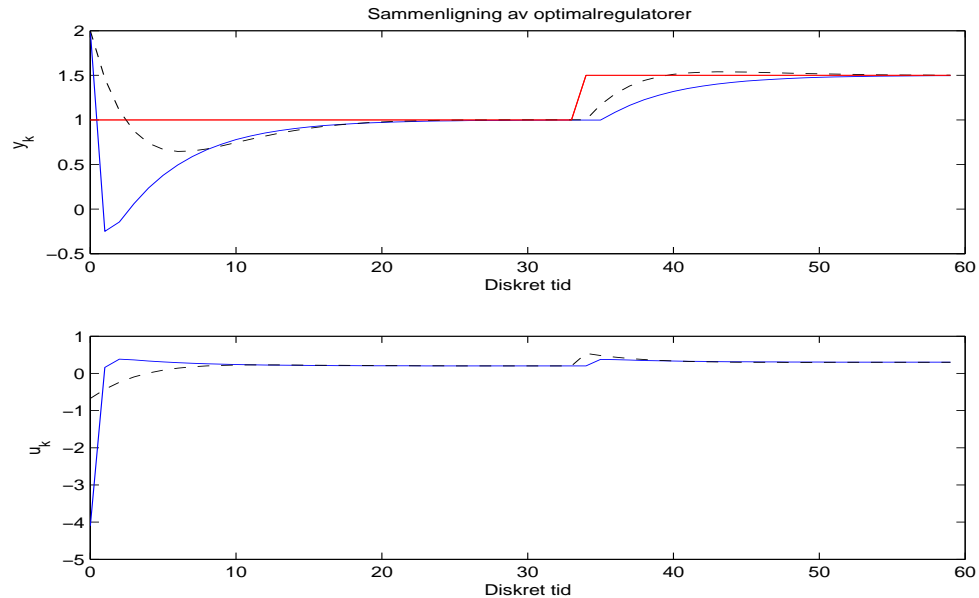
$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ D & I \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ E \end{bmatrix} u_k$$



**Figur 7.1.2:** Figuren viser optimalregulator i heltrukken blå, mot tilbakekopling fra  $y_k$  i strekete svart. Rød er referansen.

### Eksempel 7.1.3 *Optimalregulator og regulator med tilbakekopling fra $y_k$ med integralvirkning og sprang i referansen*

I dette eksemplet er det plottet de samme regulatorene som eksempel (7.1.2). Det er her tatt med et sprang i referanse for å se hvordan forskjellen er mellom de to regulatorene etter de har svingt seg inn til en fast stasjonærverdi. Fra figur (7.1.3) ser man at optimalregulatoren er litt tregere, men den gir ikke oversving. Begge regulatorene opptrer her som PI-regulatorer, men optimalregulatoren har tilbakekopling fra  $x_k$ . Dette genereres ved kjøring av MatLab skript-filen **SISO.m**. Optimalregulatoren er simulert uten noen stasjonærverdier  $u_s$  eller  $x_s$  da disse ikke er oppgitt for modellen. Figurene genereres ved kjøring av MatLab skript-filen **simulering3.m**



**Figur 7.1.3:** Figuren viser optimalregulator i heltrukken blå, og tilbakekopling fra  $y_k$  i strekete svart. Referansen er rød, og gjør et sprang fra 1 til 1,5.

## 7.1.2 AIRC

Dette er en modell som beskriver en lineærisert modell av vertikalplandynamikken til et fly. Modellen ble diskretisert på følgende måte

$$\begin{aligned} A_d &= e^{A_c h} \\ B_d &= A_c^{-1}(e^{A_c h} - I)B_c \end{aligned}$$

der  $h$  er samplingsintervallet. Den kontinuerlige lineære modellen er på formen

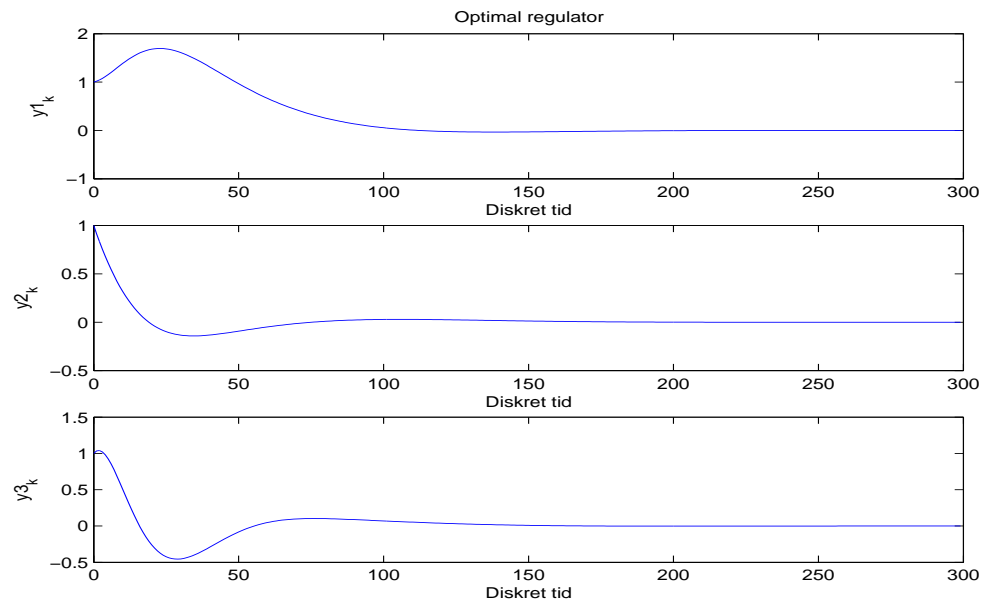
$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Dx \end{aligned}$$

Denne modellen brukes til å se på ulike former for regulatorer, og ulike strukturer.

### Eksempel 7.1.4 *Optimalregulator*

Det er her vist hvordan en optimalregulator fungerer mot modellen. Dette er vist i figurene (7.1.4), (7.1.5) og (7.1.6). Det er ikke brukt augmentering i dette eksemplet.

$$J(G) = 274,8462$$



**Figur 7.1.4:** Figuren viser optimalregulator og hvordan utgangene går mot 0.

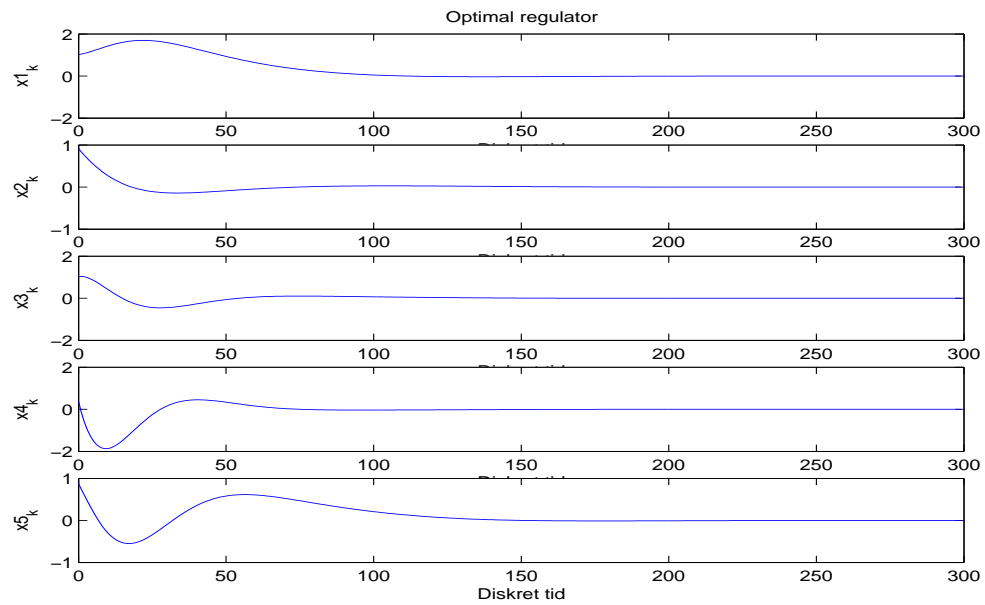
$$\gamma = 1$$

$$G_{opt} = \begin{bmatrix} 0,0287 & -0,0074 & -1,2395 & -0,4959 & 0,3671 \\ -0,2770 & -1,0136 & -0,6678 & -0,1738 & 0,4111 \\ 0,9307 & 0,2668 & 2,5764 & 0,6712 & -1,3673 \end{bmatrix}$$

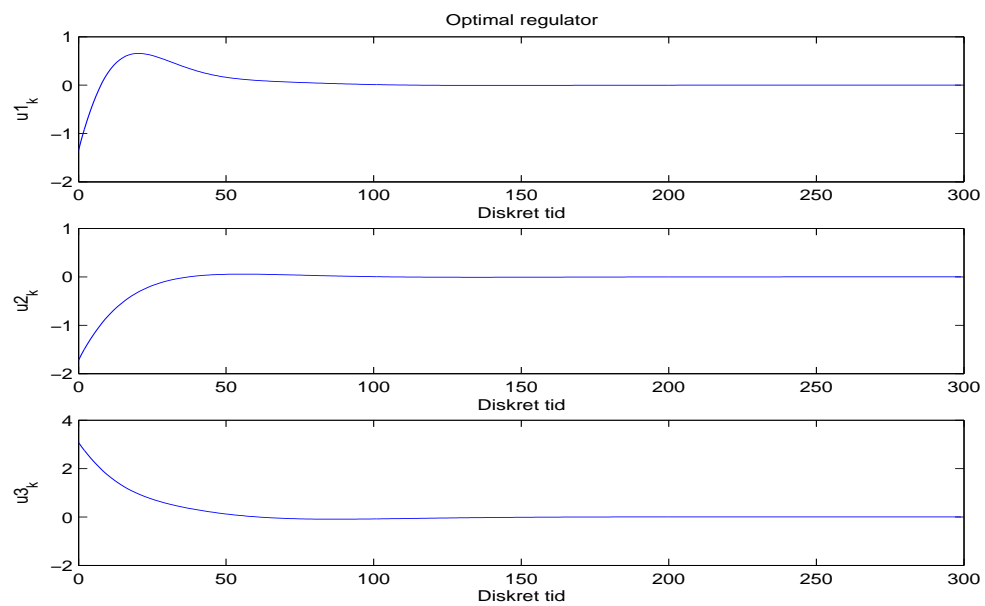
$$|eig(A + BG)| = \begin{bmatrix} 0,9271 \\ 0,9271 \\ 0,9686 \\ 0,9686 \\ 0,9499 \end{bmatrix}$$

Vektingen er som følger:

$$Q = D^T D \text{ og } P = I_{3 \times 3}$$

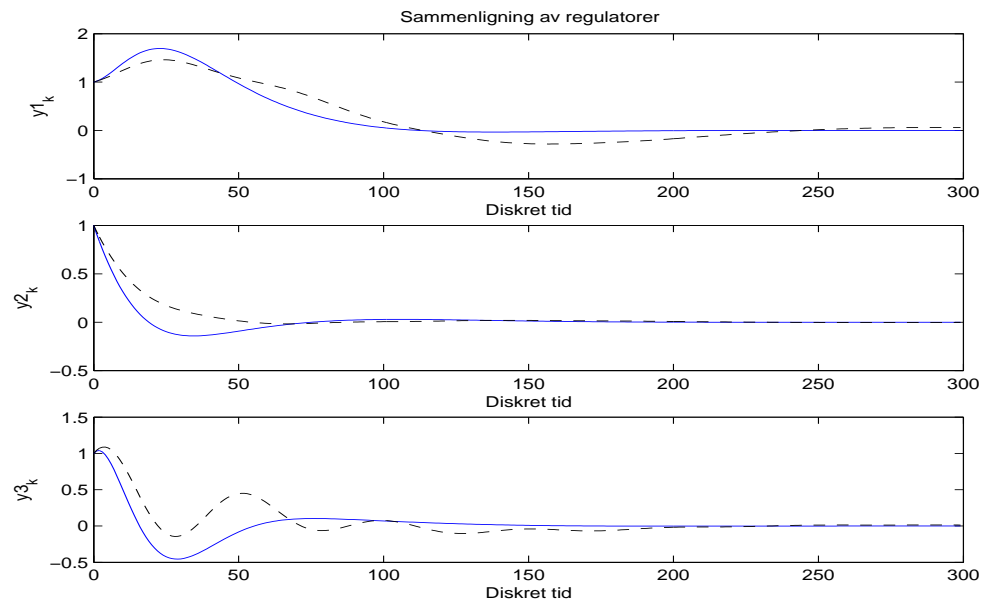


Figur 7.1.5: Figuren viser optimalregulator og hvordan tilstandene går mot 0.



Figur 7.1.6: Figuren viser optimalregulator og hvordan pådragene går mot 0.





**Figur 7.1.7:** Figuren viser hvordan utgangene reagerer mot prosessen. Heltrukken blå er optimal regulator, mens strekete svart er regulator med struktur.

### Eksempel 7.1.5 *Optimalregulator med spesifikk struktur*

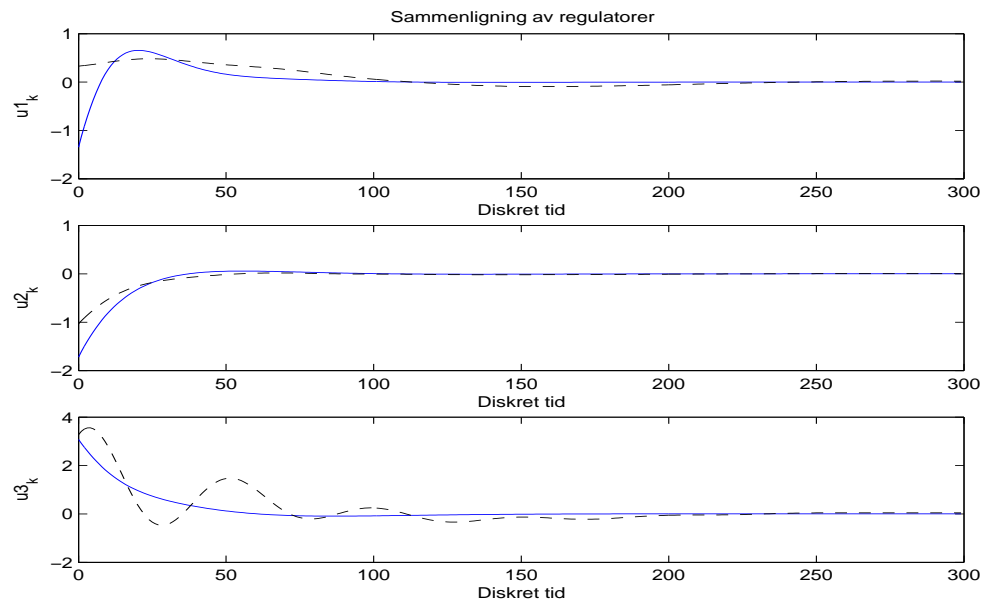
Det er her vist hvordan en optimalregulator med struktur fungerer mot modellen. Dette er vist i figurene (7.1.7) og (7.1.8). Det er tydelig å se at denne regulatoren ikke er av de beste til å styre et fly. Vektingen er som i forrige eksempel.

$$J(G) = 536,7030$$

$$\gamma = 1,9527$$

$$G = \begin{bmatrix} 0,3299 & 0 & 0 & 0 & 0 \\ 0 & -1,0353 & 0 & 0 & 0 \\ 0 & 0 & 3,2719 & 0 & 0 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,9772 \\ 0,9772 \\ 0,9884 \\ 0,9884 \\ 0,9461 \end{bmatrix}$$



**Figur 7.1.8:** Figuren viser hvordan pådragene påvirker prosessen. Heltrukken blå er optimal regulator, mens strekete svart er regulator med struktur.

#### Eksempel 7.1.6 *Optimalregulator med spesifikk struktur*

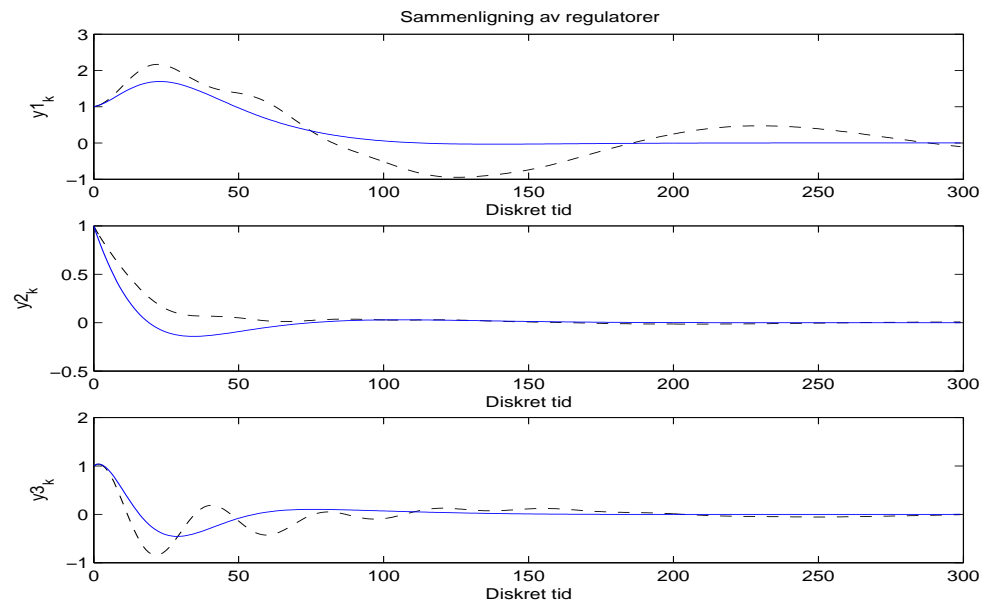
Det er her vist en annen struktur enn forrige eksempel. Dette er vist i figurene (7.1.9) og (7.1.10). Heller ikke denne regulatoren egner seg noe særlig til å styre et fly. Vektingen er som i forrige eksempel.

$$J(G) = 662,7364$$

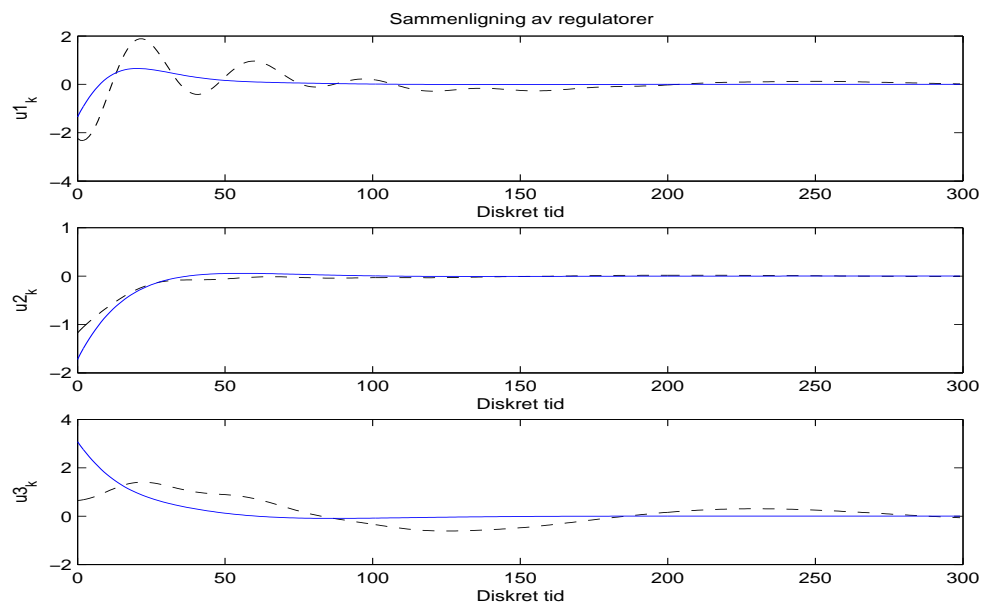
$$\gamma = 2,4113$$

$$G = \begin{bmatrix} 0 & 0 & -2,2384 & 0 & 0 \\ 0 & -1,1749 & 0 & 0 & 0 \\ 0,6476 & 0 & 0 & 0 & 0 \end{bmatrix}$$

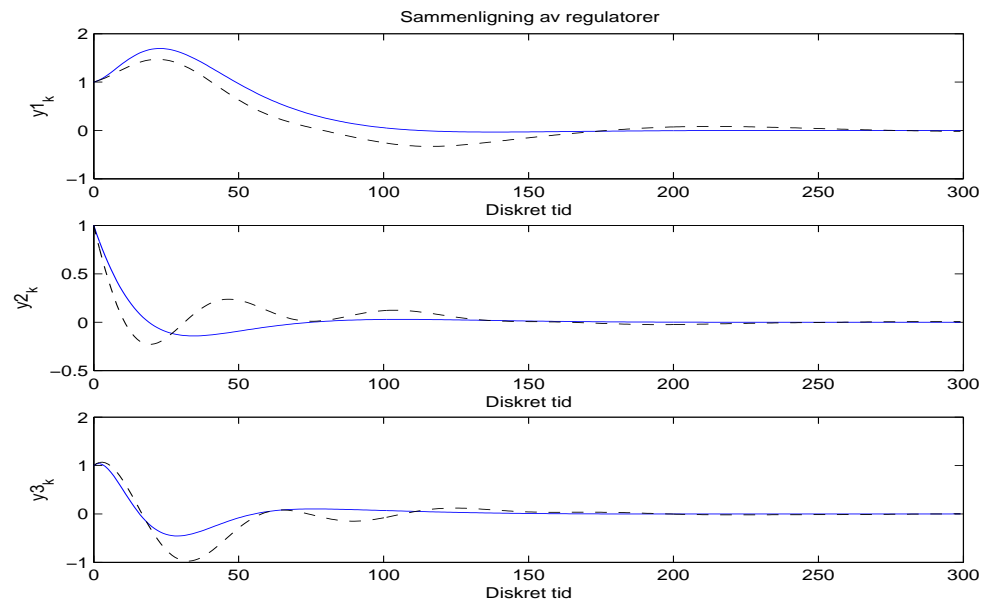
$$|eig(A + BG)| = \begin{bmatrix} 0,9744 \\ 0,9744 \\ 0,9933 \\ 0,9933 \\ 0,9392 \end{bmatrix}$$



**Figur 7.1.9:** Figuren viser hvordan utgangene reagerer mot prosessen. Heltrukken blå er optimal regulator, mens strekete svart er regulator med struktur.



**Figur 7.1.10:** Figuren viser hvordan pådragene påvirker prosessen. Heltrukken blå er optimal regulator, mens strekete svart er regulator med struktur.



**Figur 7.1.11:** Figuren viser hvordan utgangene reagerer mot prosessen. Heltrukken blå er optimal regulator, mens strekete svart er regulator med struktur.

#### Eksempel 7.1.7 *Optimalregulator med spesifikk struktur*

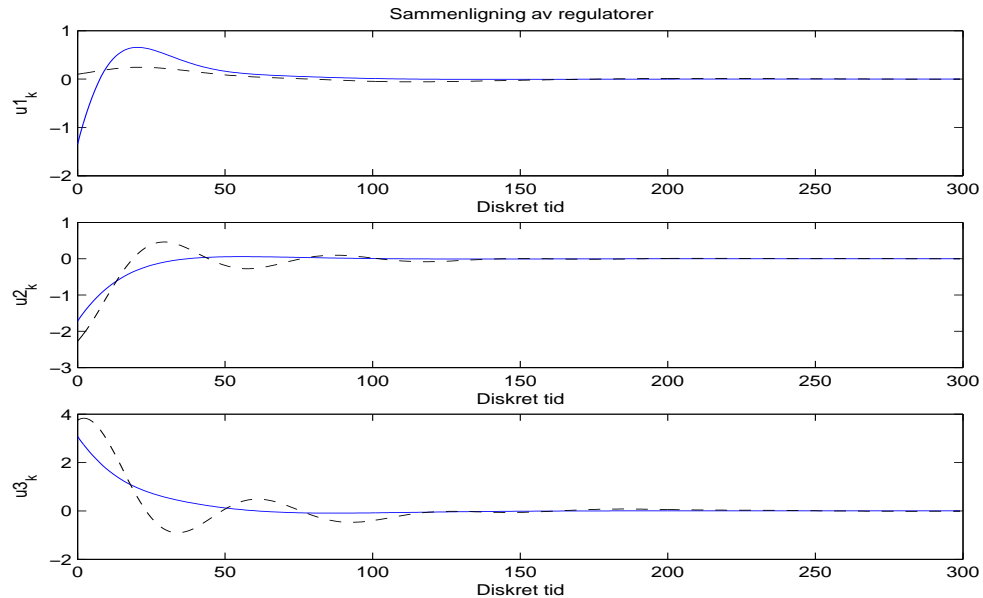
I dette eksemplet er det vist en annen struktur. Dette er vist i figurene (7.1.11) og (7.1.12). Heller ikke denne regulatoren egner seg til å styre et fly. Vektingen er som tidligere.

$$J(G) = 365,0280$$

$$\gamma = 1,3281$$

$$G = \begin{bmatrix} 0,1572 & -0,0571 & -0,0015 & 0 & 0 \\ -0,3046 & -1,0909 & -0,8786 & 0 & 0 \\ 1,0330 & 0,4113 & 2,3029 & 0 & 0 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,9729 \\ 0,9729 \\ 0,9864 \\ 0,9864 \\ 0,9531 \end{bmatrix}$$



**Figur 7.1.12:** Figuren viser hvordan pådragene påvirker prosessen. Heltrukken blå er optimal regulator, mens strekete svart er regulator med struktur.

### Eksempel 7.1.8 *Optimalregulator med integralvirkning*

Det er her vist hvordan en optimalregulator med integralvirkning fungerer mot modellen. Dette er vist i figurene (7.1.13) og (7.1.14). Referansen gjør her et sprang

fra  $\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0,7 \\ 0,6 \\ 0,5 \end{bmatrix}$ . Optimalregulatoren er simulert uten stasjonærverdier  $u_s$  eller  $x_s$  da disse ikke er oppgitt for modellen.

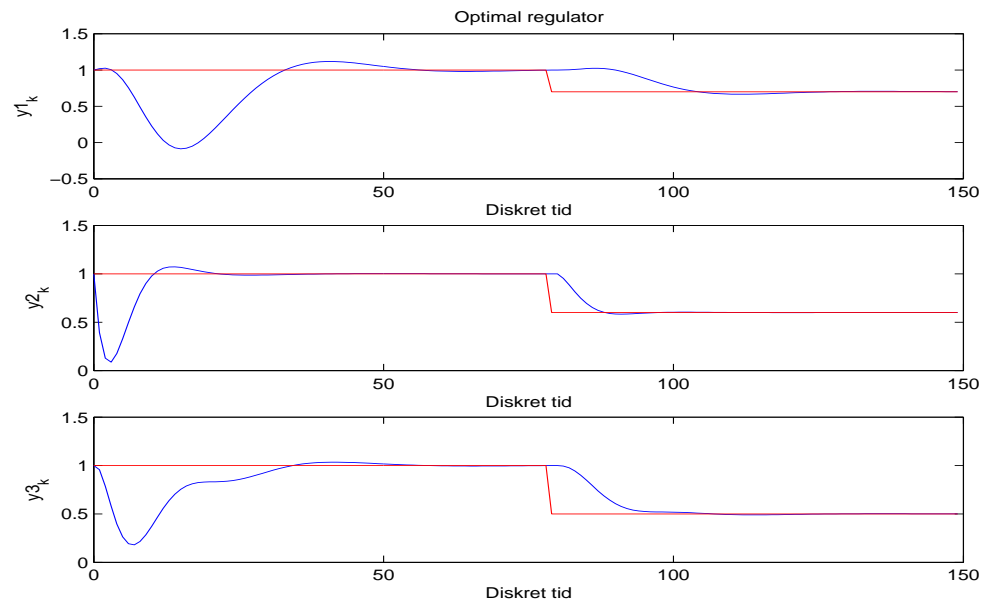
$$J(G) = 1,5024 \cdot 10^4$$

$$\gamma = 1$$

$$G_{opt} = \begin{bmatrix} G_1 & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 16,0654 & 0,3212 & -5,7514 & -0,8971 & -4,1729 \\ 0,6733 & -11,1842 & -0,4212 & -0,1084 & -0,3127 \\ 35,3828 & 0,2034 & 30,5247 & 4,1668 & -11,3534 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -1,3967 & -0,0783 & 1,9205 \\ -0,0439 & 2,3588 & 0,1198 \\ -2,3310 & 0,0611 & -1,3192 \end{bmatrix}$$



**Figur 7.1.13:** Figuren viser optimalregulator og hvordan utgangene følger referansene.

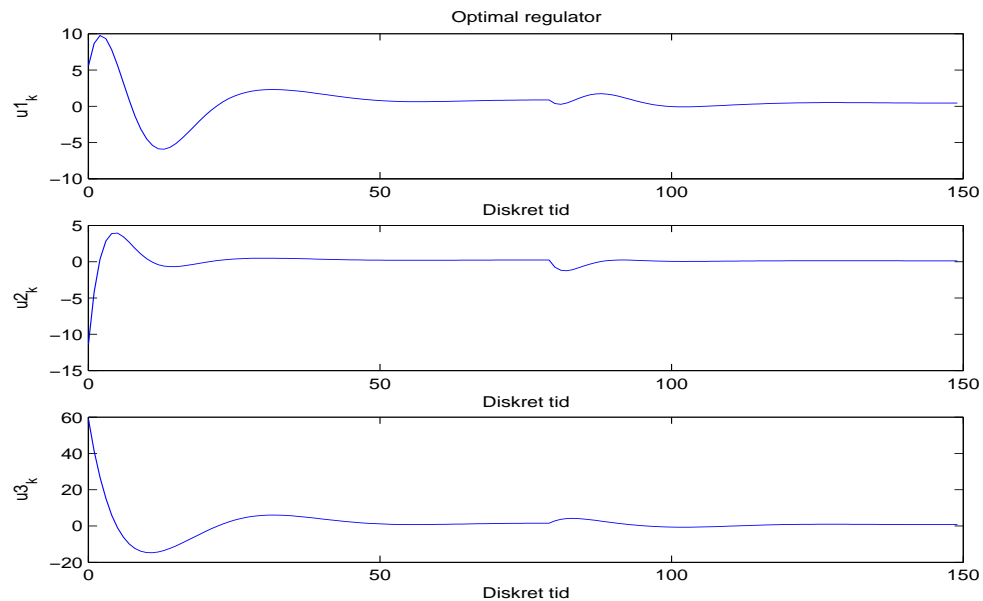
$$|eig(A + BG)| = \begin{bmatrix} 0,8392 \\ 0,8392 \\ 0,7449 \\ 0,7449 \\ 0,9249 \\ 0,9249 \\ 0,8575 \\ 0,7187 \end{bmatrix}$$

Vektingen er som følger:

$$Q = 10D^T D, \quad Q_z = 10I_{3 \times 3} \quad \text{og} \quad P = I_{3 \times 3}$$

Det er brukt følgende augmentering:

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -D & I \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k$$



**Figur 7.1.14:** Figuren viser pådragene til simuleringen.

**Eksempel 7.1.9** *Optimalregulator med integralvirkning og tilbakekopling fra  $y_k$*

Det er her vist hvordan en multivariabel PI-regulator fungerer mot modellen. Dette er vist i figurene (7.1.15) og (7.1.16). Vekting, augmentering og referansen er den samme som i forrige eksempel.

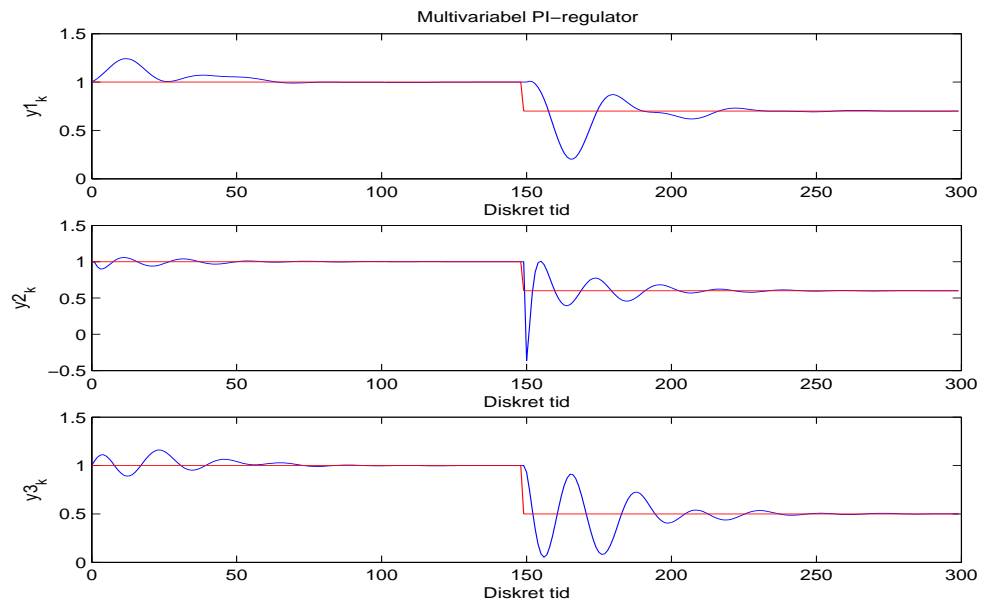
$$J(G) = 2,8199 \cdot 10^4$$

$$\gamma = 1,8769$$

$$G = \begin{bmatrix} -G_1 D & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} -8,3518 & 3,9441 & 5,4079 \\ 10,6321 & 30,0128 & 24,6129 \\ -36,7402 & -4,8236 & -33,1984 \end{bmatrix}$$

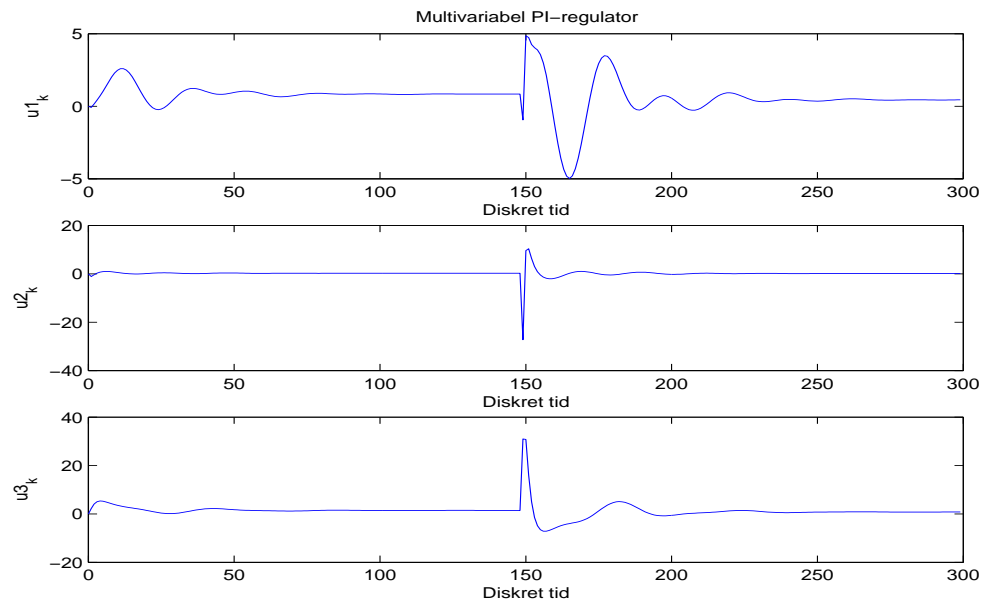
$$G_2 = \begin{bmatrix} 0,0454 & 0,4441 & -0,4711 \\ 0,1403 & 11,4528 & 2,3214 \\ 0,7298 & -14,4814 & -6,2550 \end{bmatrix}$$



**Figur 7.1.15:** Figuren viser multivariabel PI-regulator og hvordan utgangene følger referansene.

$$|eig(A + BG)| = \begin{bmatrix} 0,2460 \\ 0,4664 \\ 0,9558 \\ 0,9558 \\ 0,9560 \\ 0,9560 \\ 0,9585 \\ 0,9585 \end{bmatrix}$$





**Figur 7.1.16:** Figuren viser pådragene til simuleringen.

**Eksempel 7.1.10** *Optimalregulator med integralvirkning, struktur og tilbakekopling fra  $y_k$*

Det er her vist hvordan PI-regulatorer fungerer mot modellen. Dette er vist i figurene (7.1.17) og (7.1.18). Vekting, øgumentering og referansen er den samme som i forrige eksempel. Det er valgt en spesiell parring av variablene. Denne parringen er nødvendigvis ikke den mest optimale.

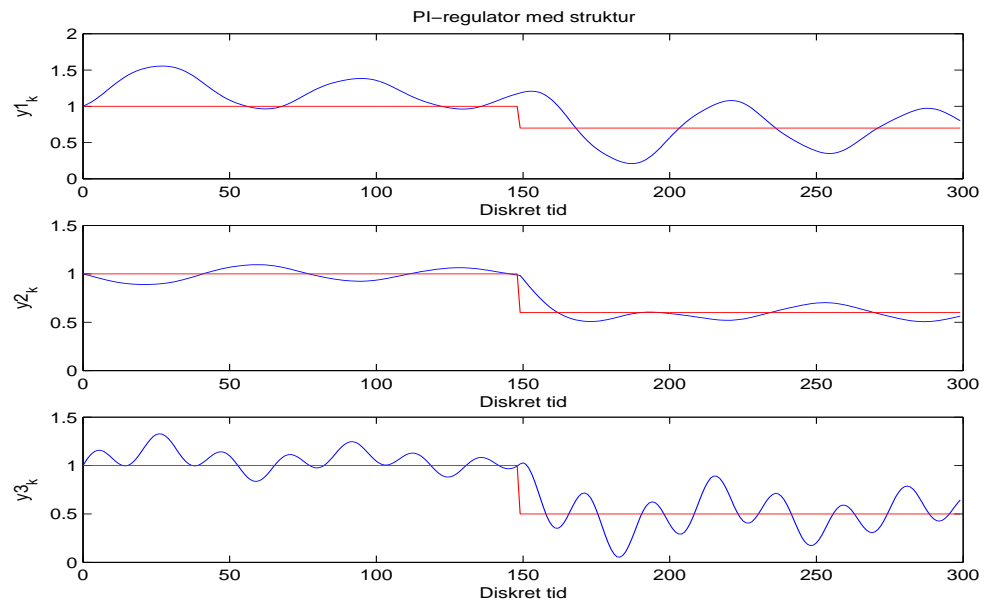
$$J(G) = 2,587 \cdot 10^5$$

$$\gamma = 17,2196$$

$$G = \begin{bmatrix} -G_1 D & G_2 \end{bmatrix}$$

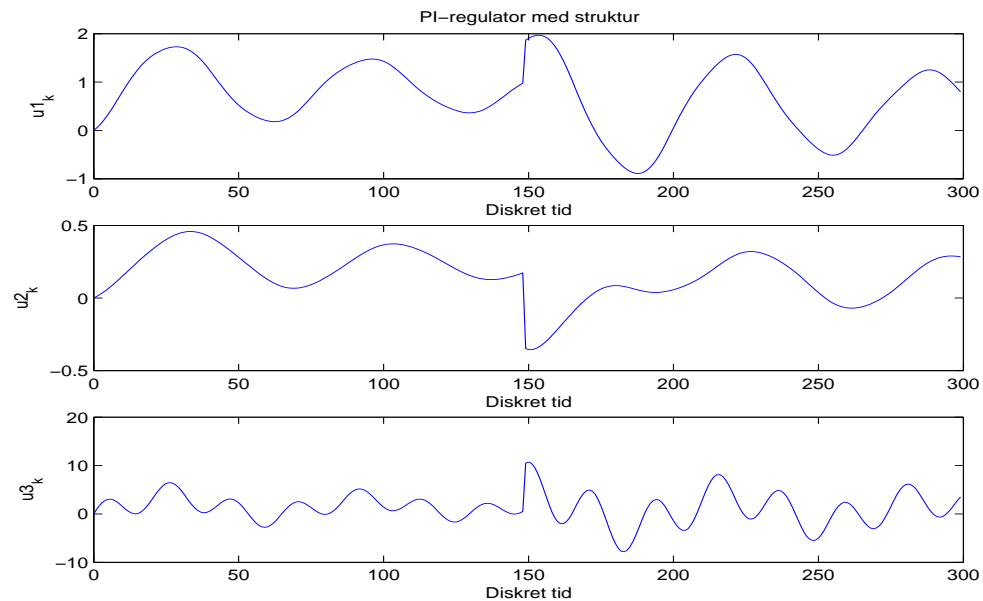
$$G_1 = \begin{bmatrix} -2,8493 & 0 & 0 \\ 0 & 1,3241 & 0 \\ 0 & 0 & -19,1121 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -0,0162 & 0 & 0 \\ 0 & 0,1393 & 0 \\ 0 & 0 & -0,0663 \end{bmatrix}$$



**Figur 7.1.17:** Figuren viser PI-regulator og hvordan utgangene følger referansene.

$$|eig(A + BG)| = \begin{bmatrix} 0,9964 \\ 0,9964 \\ 0,9550 \\ 0,9550 \\ 0,9686 \\ 0,9686 \\ 0,9942 \\ 0,9964 \end{bmatrix}$$



Figur 7.1.18: Figuren viser pådragene til simuleringen.

**Eksempel 7.1.11** *Optimalregulator med integralvirkning, struktur og tilbakekopling fra  $y_k$*

Det er her vist hvordan PI-regulatorer fungerer mot modellen. Dette er vist i figurene (7.1.19) og (7.1.20). Vekting, øgementering og referansen er den samme som i forrige eksempel. Det er valgt en annen parring av variablene enn forrige eksempel. Denne parringen er nødvendigvis ikke den mest optimale.

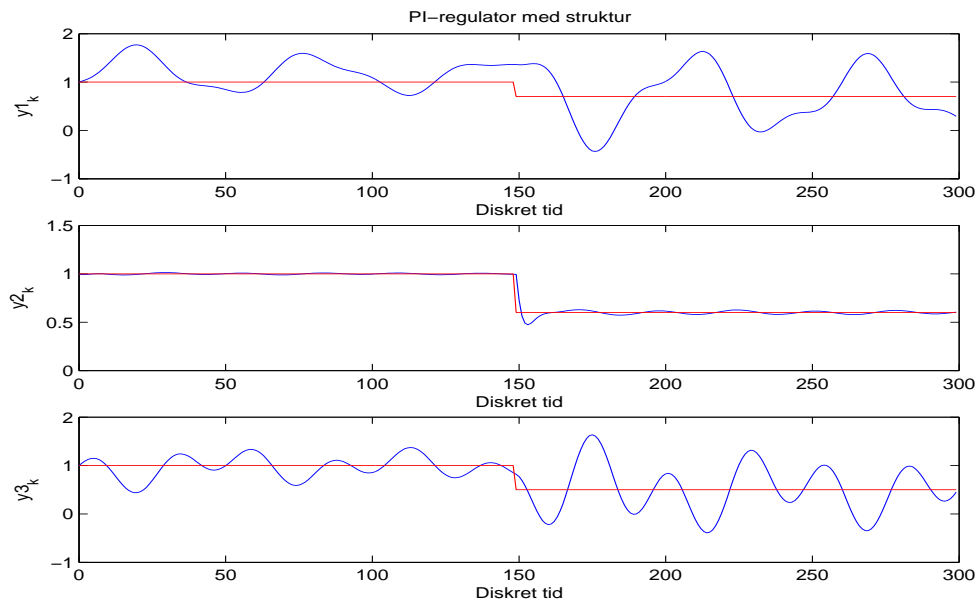
$$J(G) = 1,5024 \cdot 10^4$$

$$\gamma = 32,8517$$

$$G = \begin{bmatrix} -G_1 D & G_2 \end{bmatrix}$$

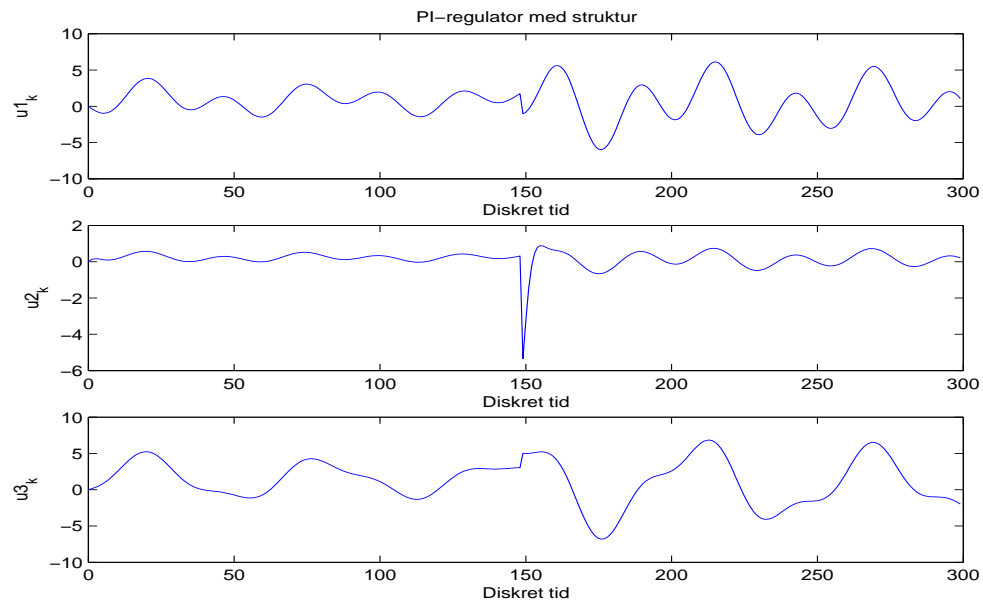
$$G_1 = \begin{bmatrix} 0 & 0 & 5,9794 \\ 0 & 14,1969 & 0 \\ -6,5962 & 0 & 0 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 0 & 0 & 0,1947 \\ 0 & 4,2163 & 0 \\ -0,0216 & 0 & 0 \end{bmatrix}$$



**Figur 7.1.19:** Figuren viser PI-regulator og hvordan utgangene følger referansene.

$$|eig(A + BG)| = \begin{bmatrix} 0,7064 \\ 0,7064 \\ 0,9974 \\ 0,9974 \\ 0,9966 \\ 0,9966 \\ 0,9967 \\ 0,9699 \end{bmatrix}$$



Figur 7.1.20: Figuren viser pådragene til simuleringen.

### 7.1.3 Example 3

Denne modellen har formen

$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k \\ y_k &= Dx_k\end{aligned}$$

#### Eksempel 7.1.12 *Optimalregulator med integralvirkning*

Det er her vist hvordan en optimalregulator fungerer mot modellen. Dette er vist i figurene (7.1.21) og (7.1.22). Referansen gjør et sprang fra

$\begin{bmatrix} 2 \\ 2 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ . Optimalregulatoren er simulert uten noen stasjonærverdier  $u_s$  eller  $x_s$  da disse ikke er oppgitt for modellen.

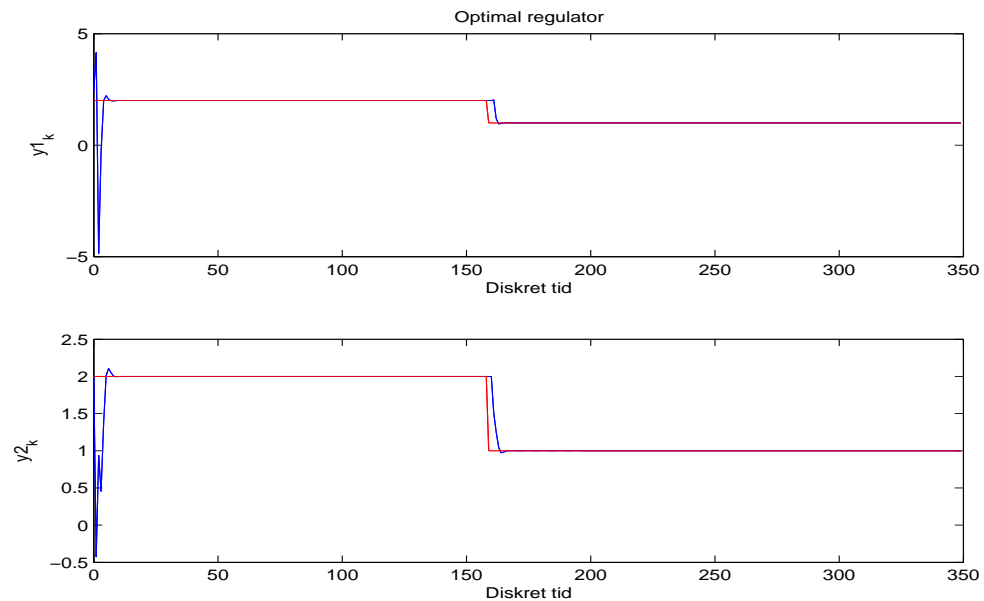
$$J(G) = 347,7048$$

$$\gamma = 1$$

$$G_{opt} = \begin{bmatrix} G_1 & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 4,7410 & 1,7955 & -1,3010 \\ -3,3720 & -2,2567 & -0,2840 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -0,4344 & 0,4909 \\ 0,2948 & 0,1415 \end{bmatrix}$$



**Figur 7.1.21:** Figuren viser optimalregulator og hvordan utgangene følger referansene.

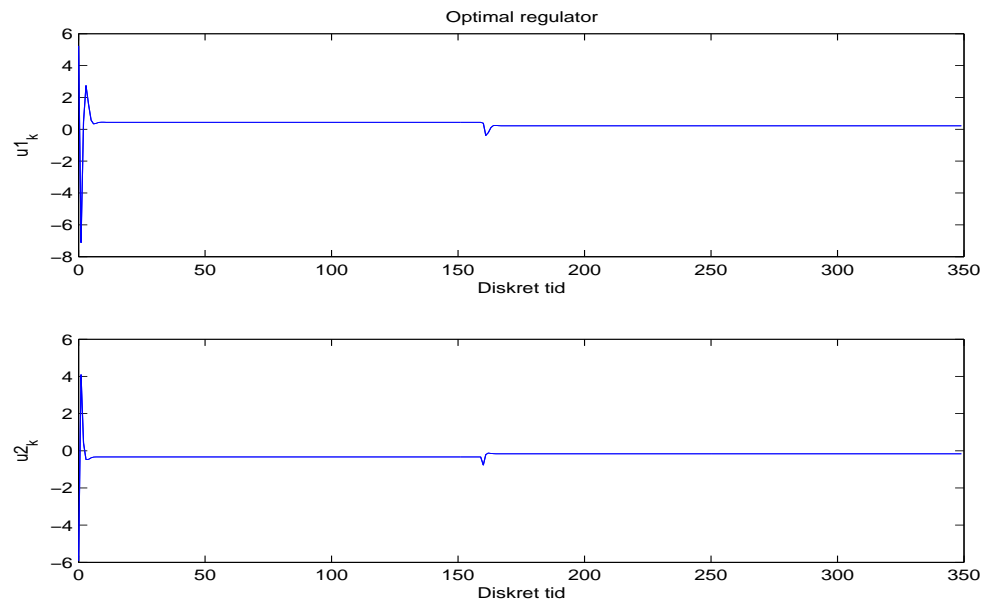
$$|eig(A + BG)| = \begin{bmatrix} 0,3784 \\ 0,3784 \\ 0,1787 \\ 0,1787 \\ 0,2214 \end{bmatrix}$$

Vektingen som er brukt er:

$$Q = D^T I_{2 \times 2} D, \quad Q_z = 5 I_{2 \times 2} \quad \text{og} \quad P = I_{2 \times 2}$$

Det er brukt denne augmenteringen for å oppnå integralvirkning:

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -D & I \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k$$



**Figur 7.1.22:** Figuren viser pådragene til simuleringen.

**Eksempel 7.1.13** *Multivariabel PI-regulator med tilbakekopling fra  $y_k$*

Det er her vist hvordan en multivariabel PI-regulator fungerer mot modellen. Dette er vist i figurene (7.1.23) og (7.1.24). Vektingen, augmenteringen og referansen er det samme som i forrige eksempel.

$$J(G) = 999,1200$$

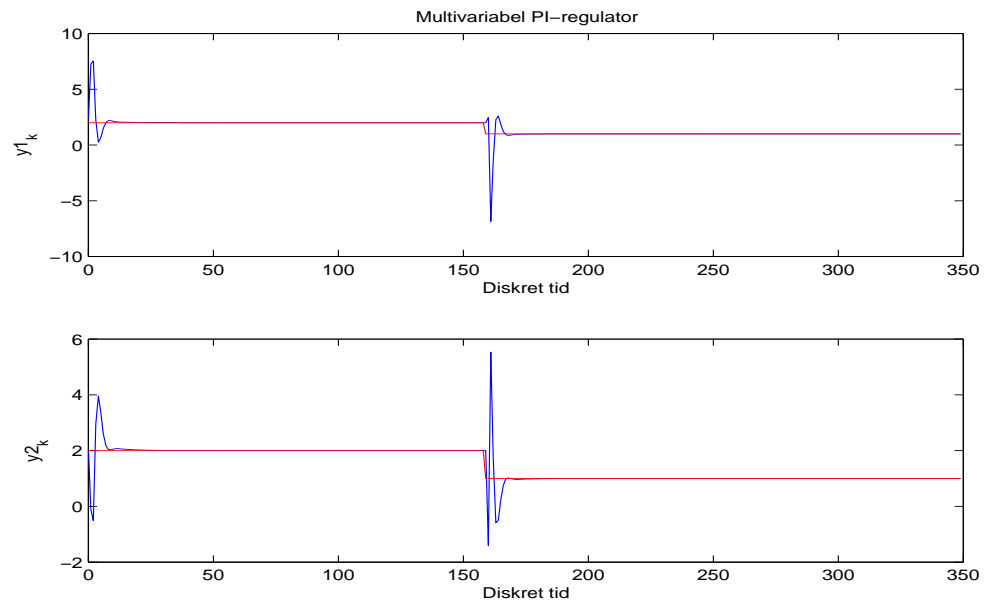
$$\gamma = 2,8735$$

$$G = \begin{bmatrix} -G_1 D & G_2 \end{bmatrix}$$

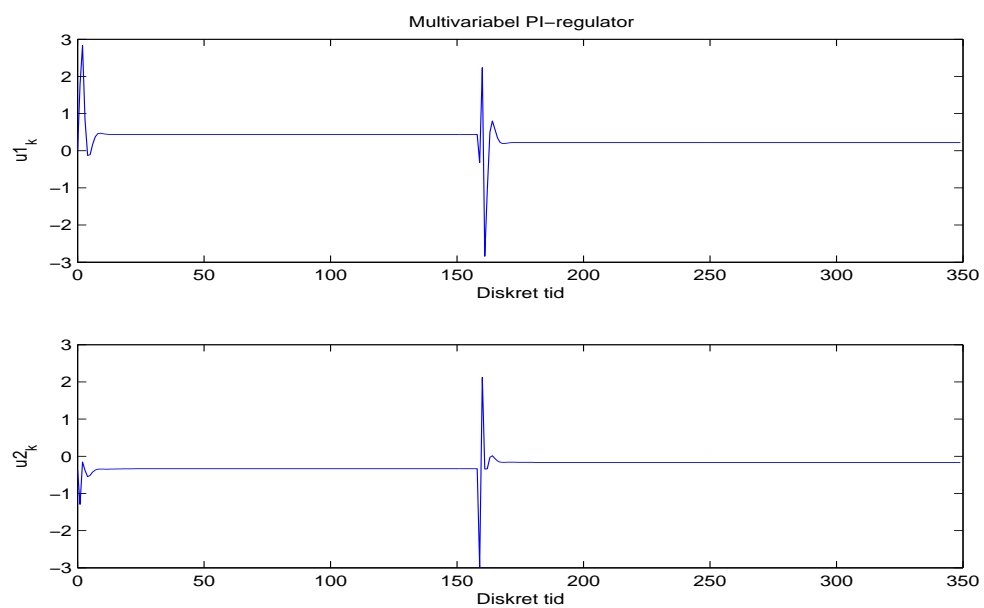
$$G_1 = \begin{bmatrix} -0,0209 & 0,7771 \\ 0,9310 & 1,7149 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -0,0702 & 0,1698 \\ -0,0028 & 0,3148 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,2020 \\ 0,2020 \\ 0,5036 \\ 0,5036 \\ 0,8712 \end{bmatrix}$$

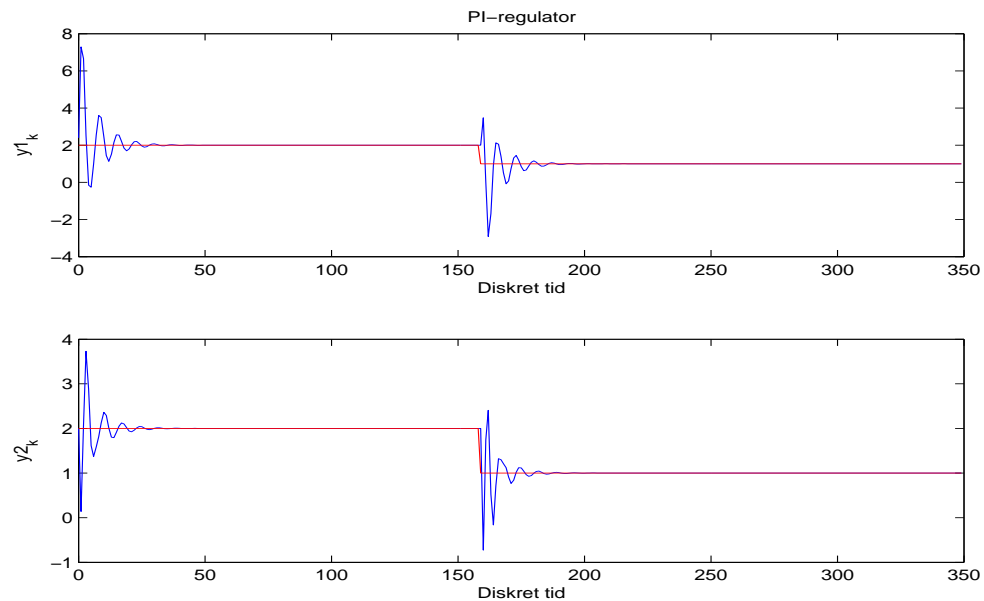


**Figur 7.1.23:** Figuren viser multivariabel PI-regulator og hvordan utgangene følger referansene.



**Figur 7.1.24:** Figuren viser pådragene til simuleringen.





**Figur 7.1.25:** Figuren viser PI-regulator og hvordan utgangene følger referansene.

**Eksempel 7.1.14** *PI-regulator med struktur og tilbakekopling fra  $y_k$*

Det er her vist hvordan PI-regulatorer fungerer mot modellen. Strukturen som her er valgt er nødvendigvis ikke den beste. Simuleringene er vist i figurene (7.1.25) og (7.1.26). Vektingen, augmenteringen og referansen er det samme som i forrige eksempel.

$$J(G) = 1,7320 \cdot 10^4$$

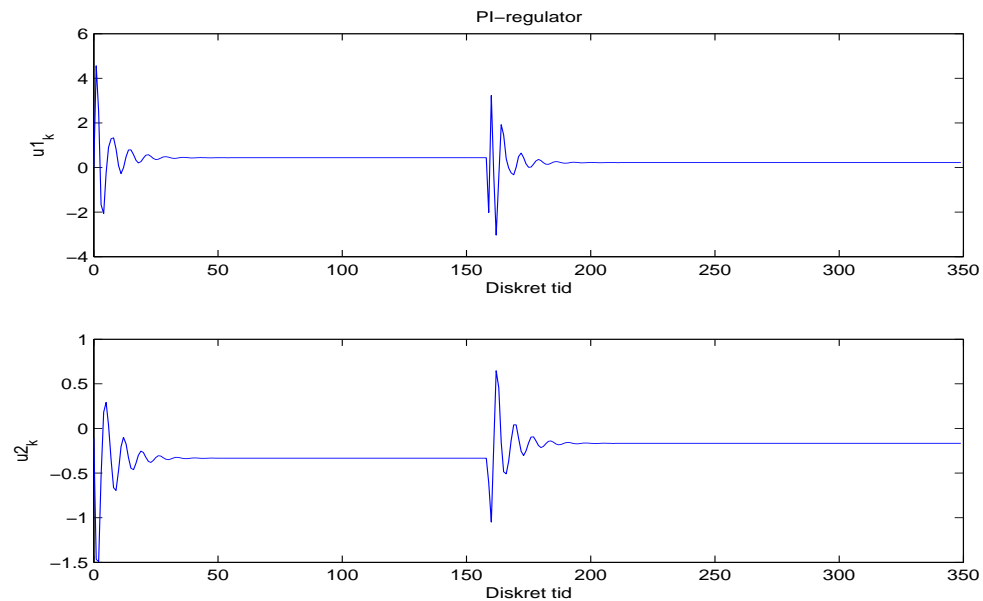
$$\gamma = 4,9553$$

$$G = \begin{bmatrix} -G_1 D & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 0 & 2,4531 \\ 0,2744 & 0 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 0 & 1,4509 \\ 0,0377 & 0 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,6704 \\ 0,6704 \\ 0,8638 \\ 0,8638 \\ 0,8663 \end{bmatrix}$$



**Figur 7.1.26:** Figuren viser pådragene til simuleringen.

**Eksempel 7.1.15** *PI-regulator med struktur tilbakekopling fra  $y_k$*

Det er her vist hvordan PI-regulatorer fungerer mot modellen. Her er det valgt en annen struktur, enn i forrige eksempel. Denne strukturen er ikke nødvendigvis den beste. Simuleringene er vist i figurene (7.1.27) og (7.1.28). Vektingen, augmenteringen og referansen er det samme som i forrige eksempel.

$$J(G) = 3,6306 \cdot 10^4$$

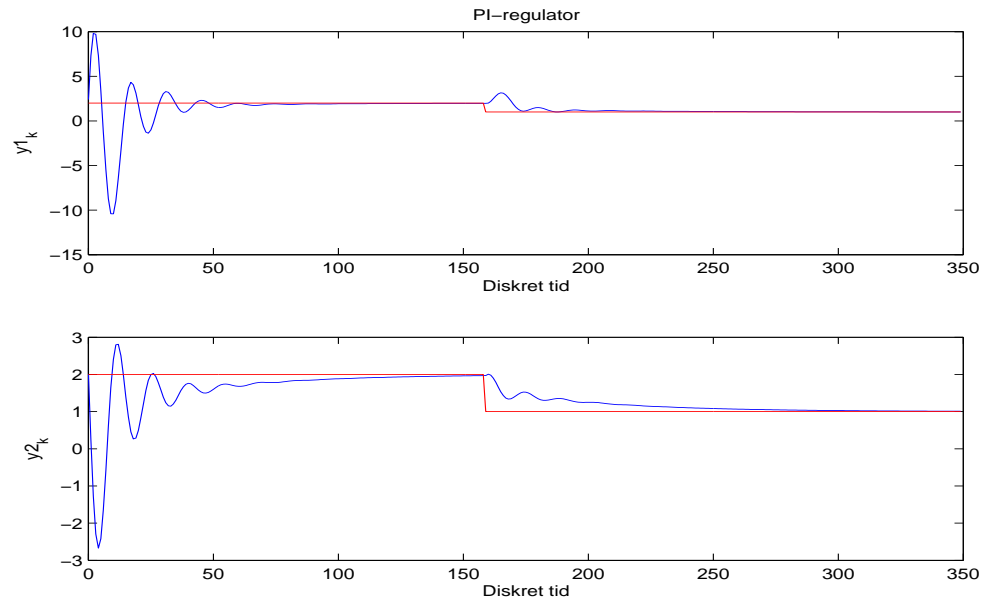
$$\gamma = 104,4149$$

$$G = \begin{bmatrix} -G_1 D & G_2 \end{bmatrix}$$

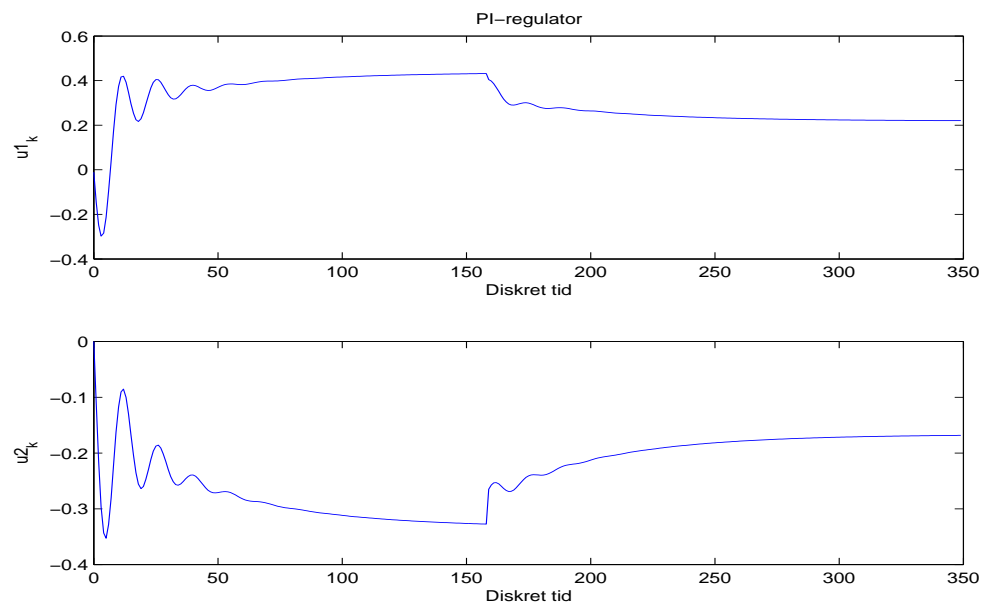
$$G_1 = \begin{bmatrix} 0,0270 & 0 \\ 0 & -0,0627 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 0,0066 & 0 \\ 0 & -0,0054 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,9262 \\ 0,9262 \\ 0,8926 \\ 0,8926 \\ 0,9779 \end{bmatrix}$$



**Figur 7.1.27:** Figuren viser PI-regulator og hvordan utgangene følger referansen.



**Figur 7.1.28:** Figuren viser pådragene til simuleringen.

### 7.1.4 Destillasjonskolonne

Dette er en kontinuerlig modell som beskriver en åttetrinns destillasjonskolonne. Det er laget en linærisert modell av denne. Regulatoren bygger på den diskrete linæriserte modellen, men simuleringene er testet mot den diskretiserte ulineære modellen. Den ulineære modellen ligger i MatLab skript-filen **fcoll.m**. For å finne de diskrete matrisene ble det brukt følgende formler

$$\begin{aligned} A_d &= e^{A_c h} \\ B_d &= A_c^{-1}(e^{A_c h} - I)B_c \end{aligned}$$

der  $h$  er samplingsintervallet. Den kontinuerlige lineære modellen er av formen

$$\begin{aligned} \dot{x} &= Ax + Bu + Cv \\ y &= Dx \end{aligned}$$

Denne modellen brukes til å se på ulike former for regulatorer. Det sees spesielt på ulike parringer av variable. Det vil si hva slags pådrag skal styre hvilken utgang. Det sees både på standard PI-regulatorer og PI-regulatorer på endringsform.

#### Standard PI-regulator form

Verdiene i eksemplene (7.1.16), (7.1.17), (7.1.18) og (7.1.19) genereres ved kjøring av MatLab skript-filen **Destkol.m**

#### Eksempel 7.1.16 *Optimalregulator med integralvirkning*

Det er her vist hvordan en optimalregulator med integralvirkning fungerer mot modellen. Dette er vist i figurene (7.1.29) og (7.1.30). Referansen gjør her et sprang fra  $\begin{bmatrix} 0,03 \\ 0,96 \end{bmatrix} \rightarrow \begin{bmatrix} 0,035 \\ 0,945 \end{bmatrix}$ . Figurene genereres ved kjøring av MatLab skript-filen **simulering15.m**

$$J(G) = 2,4818 \cdot 10^6$$

$$\gamma = 1$$

$$\begin{aligned} G_{opt} &= \begin{bmatrix} G_1 & G_2 \end{bmatrix} \\ G_1 &= \begin{bmatrix} G_{1A} & G_{1B} \end{bmatrix} \end{aligned}$$

$$G_{1A} = \begin{bmatrix} -125,7204 & -2,9657 & 2,6024 & 0,6125 \\ 226,1430 & 21,8063 & 5,1429 & 0,7936 \end{bmatrix}$$

$$G_{1B} = \begin{bmatrix} -3,2010 & -9,8755 & -27,9461 & -158,9449 \\ -2,5655 & -7,7096 & -20,8630 & -111,2584 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -5,9991 & -3,6828 \\ 10,2370 & -2,4137 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,7661 \\ 0,7661 \\ 0,8378 \\ 0,8900 \\ 0,9148 \\ 0,9848 \\ 0,9848 \\ 0,9710 \\ 0,9560 \\ 0,9510 \end{bmatrix}$$

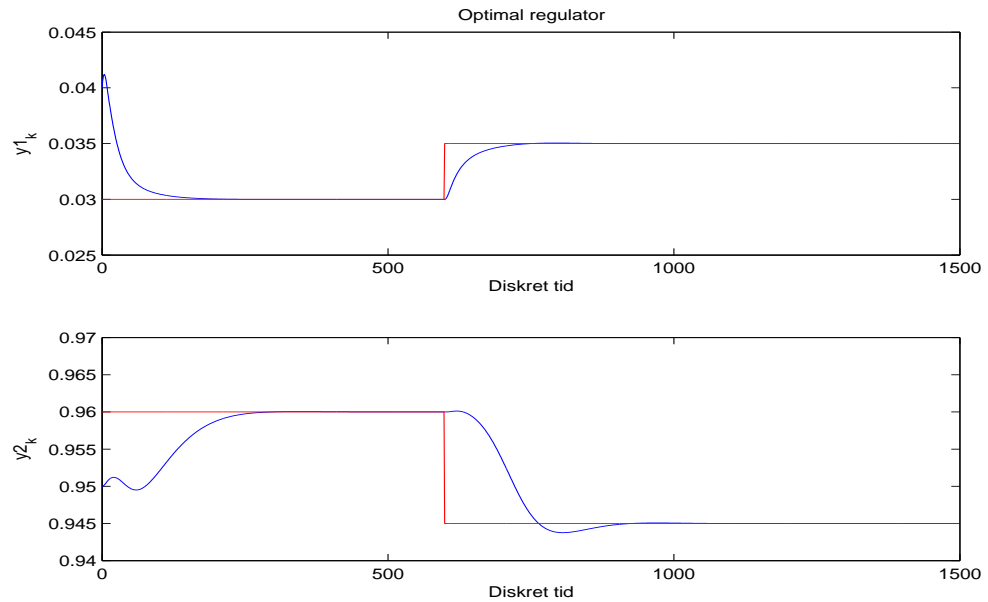
Det er brukt følgende vektning:

$$Q = \text{diag} \begin{bmatrix} 10^5 & 0 & 0 & 0 & 0 & 0 & 0 & 10^4 \end{bmatrix},$$

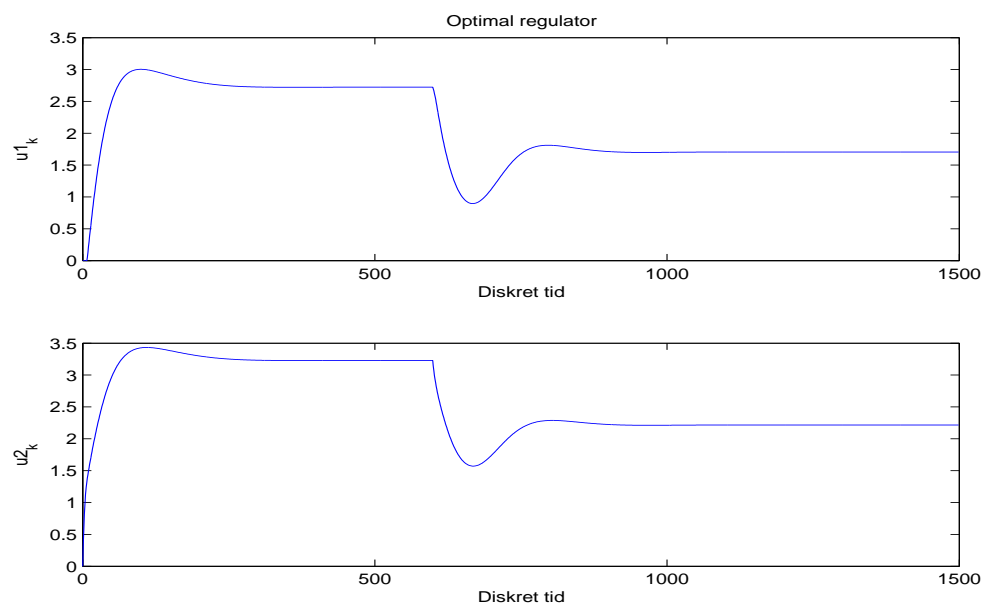
$$Q_z = \text{diag} \begin{bmatrix} 200 & 20 \end{bmatrix} \text{ og } P = I_{2 \times 2}$$

Augmenteringen som er brukt er som følger:

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ D & I \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k$$



**Figur 7.1.29:** Figuren viser optimalregulator og hvordan utgangene følger referansene.



**Figur 7.1.30:** Figuren viser pådragene til simuleringen.

**Eksempel 7.1.17 Multivariabel PI-reg med tilbakekopling fra  $y_k$** 

Det er her vist hvordan en multivariabel PI-regulator fungerer mot modellen. Dette er vist i figurene (7.1.31) og (7.1.32). Vektingen, augmentering og referansen er det samme som forrige eksempel. Figurene genereres ved kjøring av MatLab skript-filen *simulering16.m*

$$J(G) = 2,7757 \cdot 10^6$$

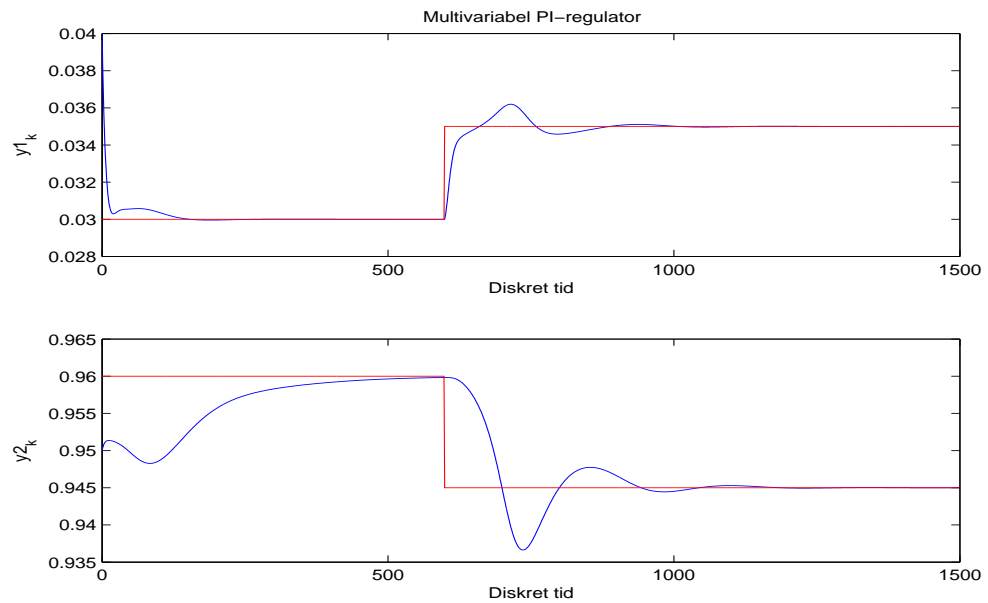
$$\gamma = 1,1184$$

$$G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$

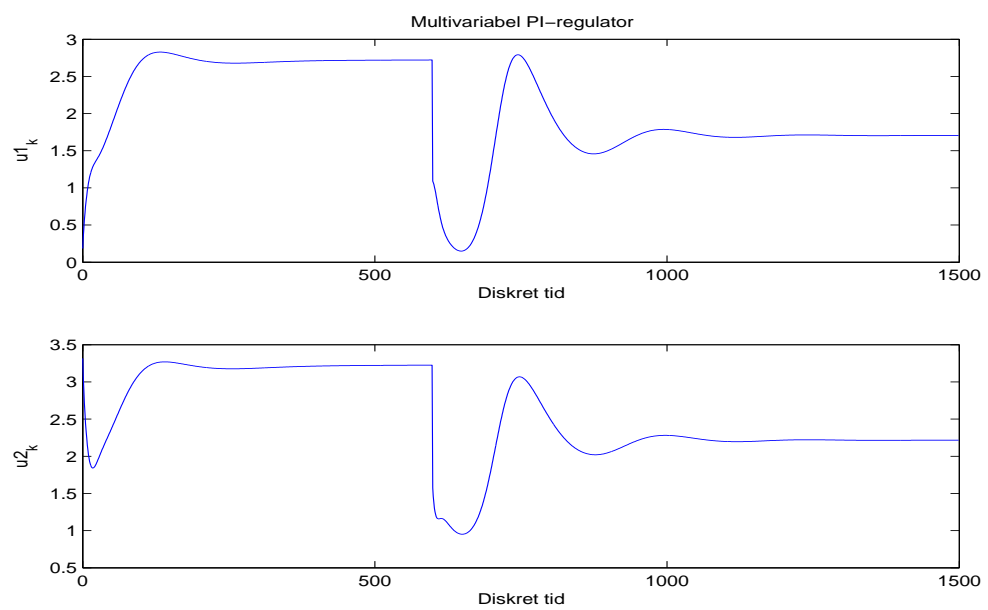
$$G_1 = \begin{bmatrix} -135,4370 & -153,8112 \\ 332,2798 & 0,6985 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -3,9914 & -1,3637 \\ 31,3481 & 0,0199 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,8186 \\ 0,8186 \\ 0,8066 \\ 0,9879 \\ 0,9879 \\ 0,9911 \\ 0,9520 \\ 0,8738 \\ 0,9257 \\ 0,8986 \end{bmatrix}$$



**Figur 7.1.31:** Figuren viser multivariabel PI-regulator og hvordan utgangene følger referansene.



**Figur 7.1.32:** Figuren viser pådragene til simuleringen.



**Eksempel 7.1.18 *PI-regulator med feil parring, struktur og tilbakekopling fra  $y_k$*** 

Det er her vist hvordan PI-regulator fungerer mot modellen. Dette er vist i figurene (7.1.33) og (7.1.34). Vektingen, augmentering og referansen er det samme som forrige eksempel. Parringen som er valgt er ikke den mest optimale. Dette kan vises i en RGA-analyse, se Di Ruscio (2002). Figurene genereres ved kjøring av MatLab skript-filen **simulering18.m**

$$J(G) = 3,3367 \cdot 10^6$$

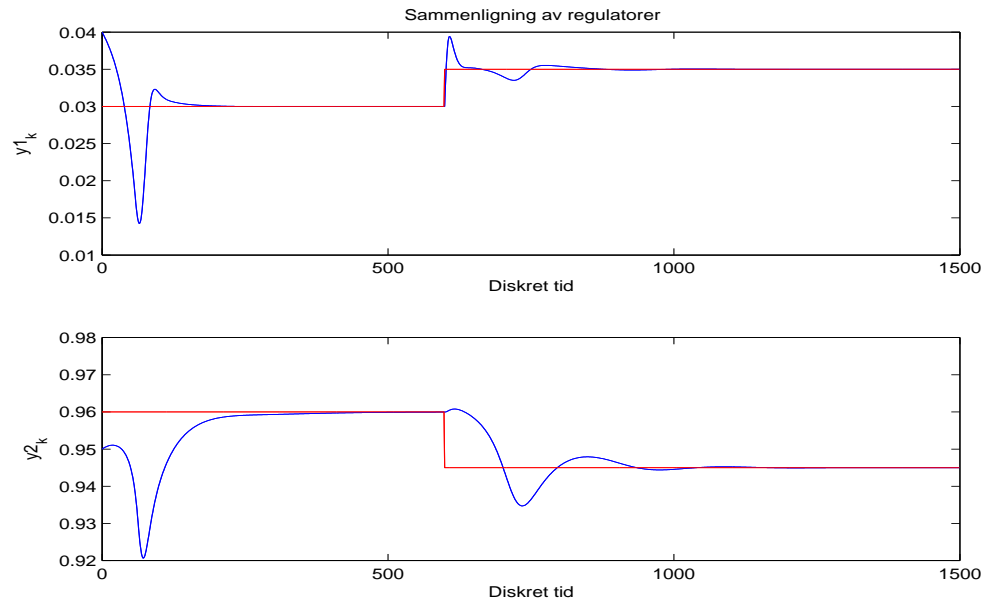
$$\gamma = 1,3444$$

$$G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$

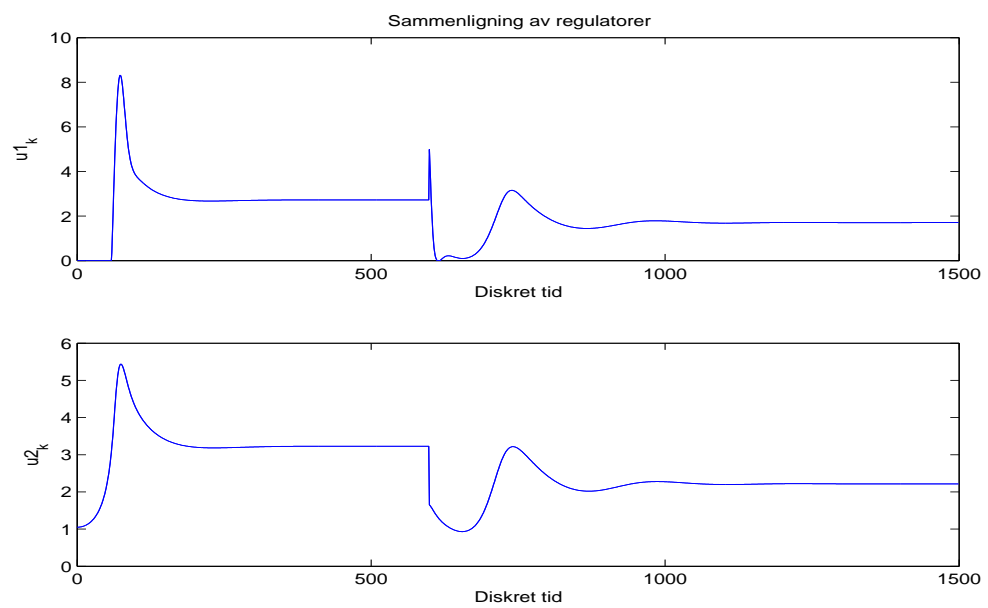
$$G_1 = \begin{bmatrix} -452,4523 & 0 \\ 0 & -105,0733 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -43,3080 & 0 \\ 0 & -1,1472 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,8559 \\ 0,8559 \\ 0,7967 \\ 0,9868 \\ 0,9868 \\ 0,9883 \\ 0,8673 \\ 0,9523 \\ 0,9005 \\ 0,9151 \end{bmatrix}$$



**Figur 7.1.33:** Figuren viser PI-regulator og hvordan utgangene følger referansene.



**Figur 7.1.34:** Figuren viser pådragene til simuleringen.

**Eksempel 7.1.19** *PI-regulator med riktig parring, struktur og tilbakekopling fra  $y_k$*

Det er her vist hvordan PI-regulator fungerer mot modellen. Dette er vist i figurene (7.1.35) og (7.1.36). Vektingen, augmentering og referansen er det samme som forrige eksempel. Parringen som er valgt her er den mest optimale. Dette kan vises i en RGA-analyse, se Di Ruscio (2002). Figurene genereres ved kjøring av MatLab skript-filen **simulering19.m**

$$J(G) = 2,8296 \cdot 10^6$$

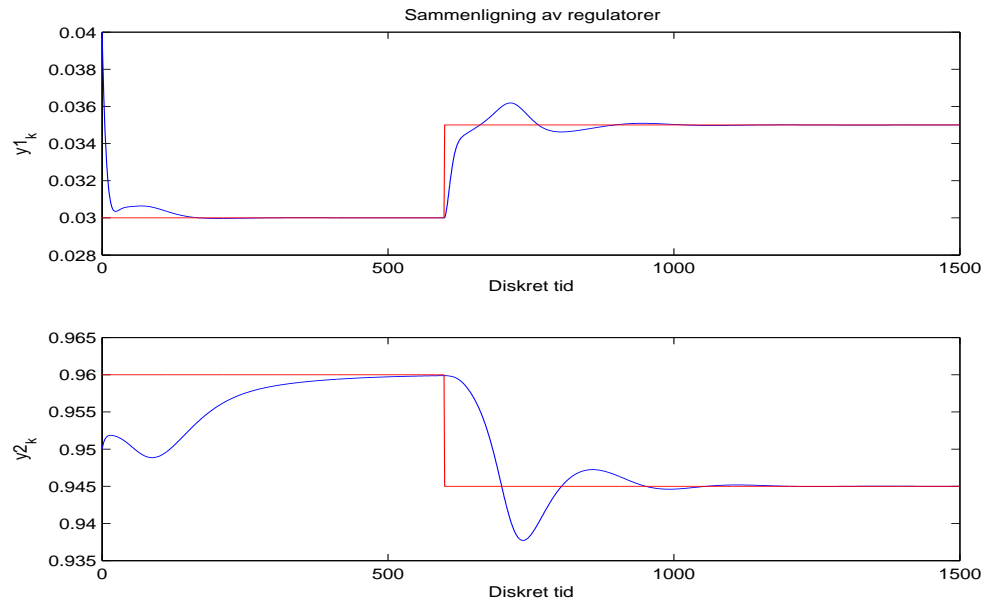
$$\gamma = 1,1401$$

$$G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$

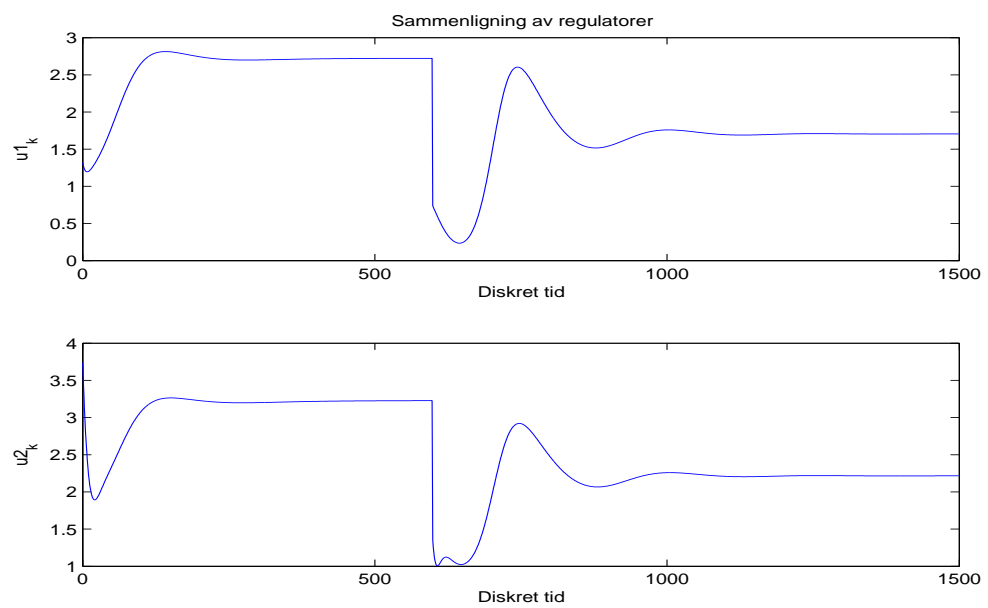
$$G_1 = \begin{bmatrix} 0 & -132,1839 \\ 373,6811 & 0 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 0 & -1,2488 \\ 26,0350 & 0 \end{bmatrix}$$

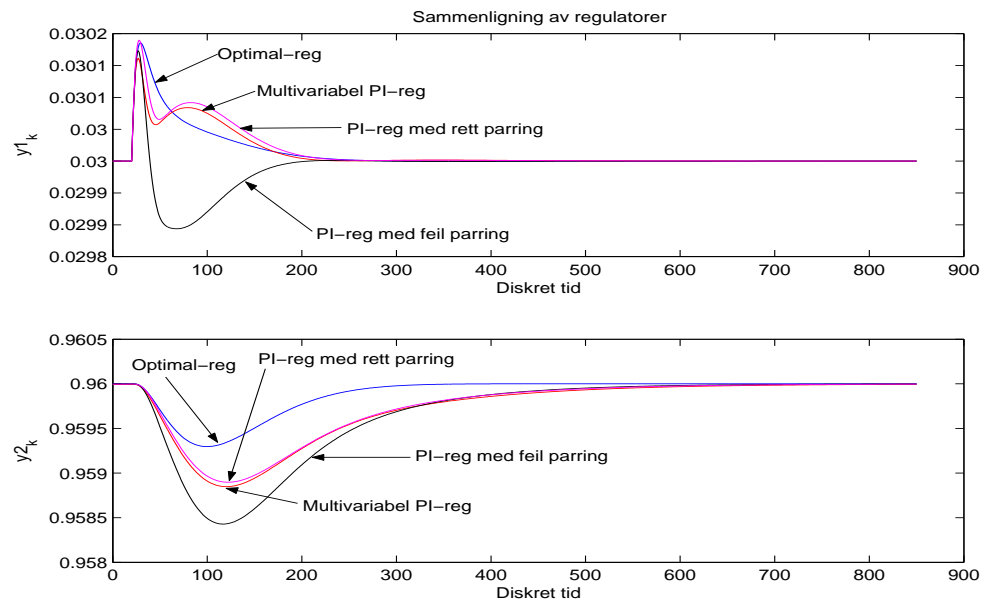
$$|eig(A + BG)| = \begin{bmatrix} 0,8441 \\ 0,8441 \\ 0,8016 \\ 0,9876 \\ 0,9876 \\ 0,9902 \\ 0,8718 \\ 0,9023 \\ 0,9310 \\ 0,9518 \end{bmatrix}$$



**Figur 7.1.35:** Figuren viser PI-regulator og hvordan utgangene følger referansene.



**Figur 7.1.36:** Figuren viser pådragene til simuleringen.



**Figur 7.1.37:** Figuren viser hvordan de ulike regulatorene reagerer på en endring i forstyrrelsen.

#### Eksempel 7.1.20 *Sammenligning av regulatorene med økning av støy*

I denne simuleringen er det satt inn en av økning støy i modellen. Fødestrømstilførselen økes med 10%. Dette er vist i figuren (7.1.37).

## PI-regulator på endringsform

### Eksempel 7.1.21 *Optimalregulator med integralvirkning*

Det er her vist hvordan en optimalregulator på endringsform fungerer mot modellen. Dette er vist i figurene (7.1.38) og (7.1.39). Referansen gjør her et sprang

$$\text{fra } \begin{bmatrix} 0,03 \\ 0,96 \end{bmatrix} \rightarrow \begin{bmatrix} 0,035 \\ 0,945 \end{bmatrix}.$$

$$J(G) = 3,9309 \cdot 10^6$$

$$\gamma = 1$$

$$G_{opt} = \begin{bmatrix} G_1 & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} G_{1A} & G_{1B} \end{bmatrix}$$

$$G_{1A} = \begin{bmatrix} -73,5281 & 0,3618 & 0,2215 & -4,2897 \\ 237,0163 & 29,3020 & 9,6740 & 2,0139 \end{bmatrix}$$

$$G_{1B} = \begin{bmatrix} -11,1328 & -17,5353 & -35,0884 & -174,8091 \\ -4,5818 & -9,5425 & -19,4342 & -90,8308 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -3,5992 & -3,8597 \\ 10,8995 & -1,8938 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,6600 \\ 0,7678 \\ 0,8481 \\ 0,8942 \\ 0,9203 \\ 0,9854 \\ 0,9854 \\ 0,9597 \\ 0,9597 \\ 0,9511 \end{bmatrix}$$

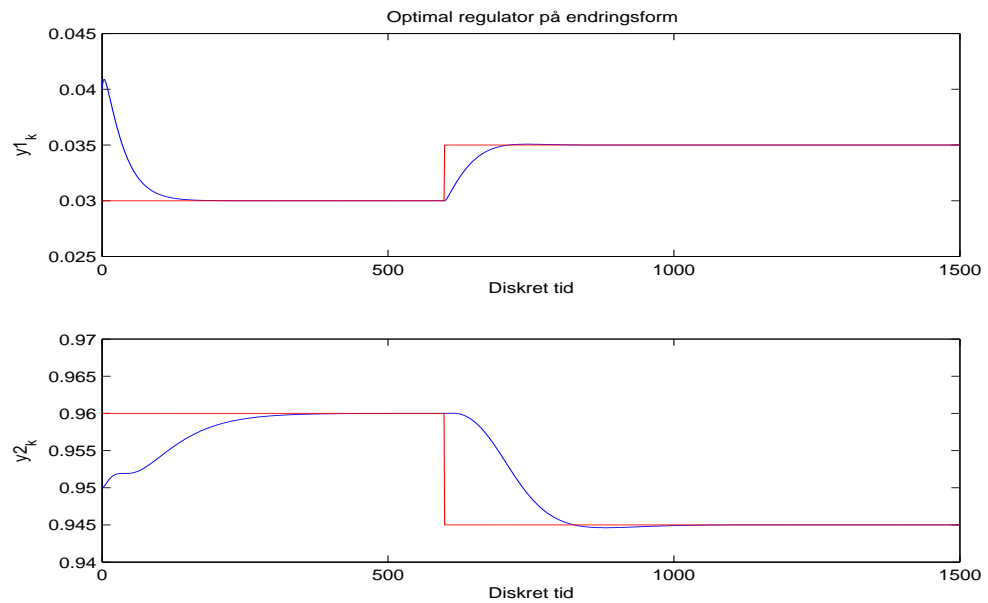
Vektingen er valgt litt annerledes enn de tidligere eksemplene. Det er her valgt å å vekte alle tilstandene, og ikke bare utgangene. Vektingen er:

$$Q = \text{diag} \begin{bmatrix} 10^5 & 10^3 & 10^3 & 10^3 & 10^3 & 10^3 & 10^3 & 10^4 \end{bmatrix},$$

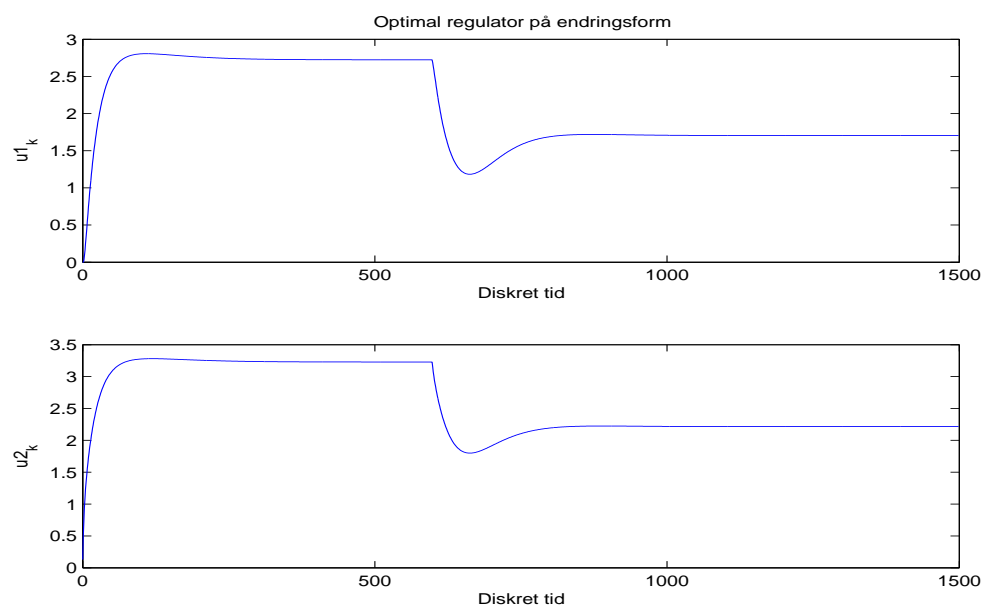
$$Q_z = \text{diag} \begin{bmatrix} 200 & 20 \end{bmatrix} \text{ og } P = I_{2 \times 2}$$

Augmenteringen som er brukt for å oppnå endringsform er:

$$\begin{bmatrix} \Delta x_{k+1} \\ y_k - r \end{bmatrix} = \begin{bmatrix} A & 0 \\ D & I \end{bmatrix} \begin{bmatrix} \Delta x_k \\ y_{k-1} - r \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \Delta u_k$$



**Figur 7.1.38:** Figuren viser optimalregulator på endringsform og hvordan utgangene følger referansene.



**Figur 7.1.39:** Figuren viser pådragene til simuleringen.

**Eksempel 7.1.22 Multivariabel PI-regulator med integralvirkning**

Det er her vist hvordan en multivariabel PI-regulator fungerer mot modellen. Dette er vist i figurene (7.1.40) og (7.1.41). Vektingen, augmentering og referansen er det samme som forrige eksempel.

$$J(G) = 4,5939 \cdot 10^6$$

$$\gamma = 1,1687$$

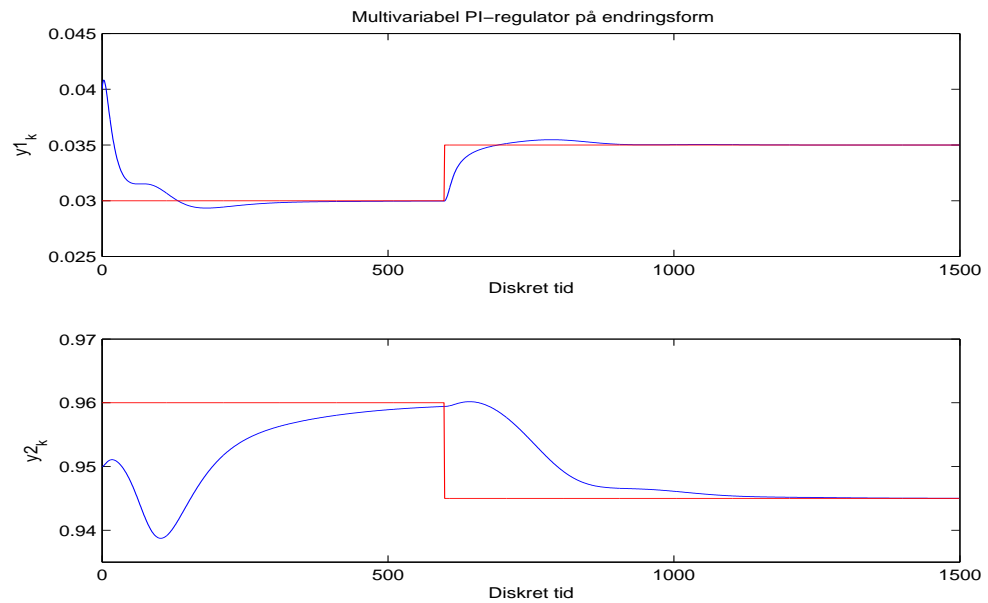
$$G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} -22,6962 & -116,3966 \\ 348,1070 & 1,0895 \end{bmatrix}$$

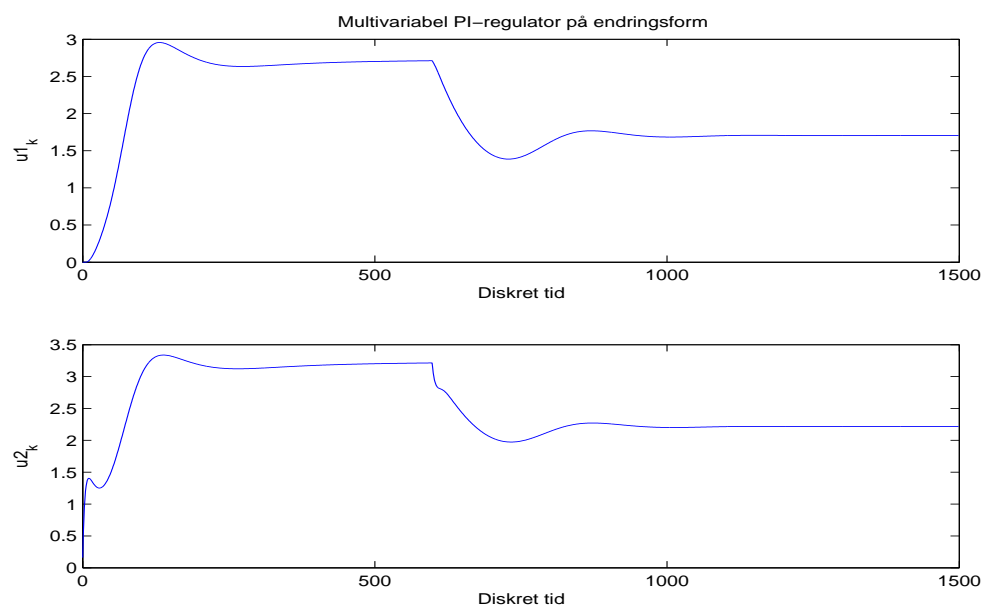
$$G_2 = \begin{bmatrix} -0,2256 & -0,8856 \\ 15,3534 & -0,6280 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,8205 \\ 0,8205 \\ 0,8114 \\ 0,8802 \\ 0,9080 \\ 0,9867 \\ 0,9867 \\ 0,9926 \\ 0,9540 \\ 0,9514 \end{bmatrix}$$





**Figur 7.1.40:** Figuren viser multivariabel PI-regulator og hvordan utgangene følger referansene.



**Figur 7.1.41:** Figuren viser pådragene til simuleringen.

**Eksempel 7.1.23** *PI-regulator med feil parring, struktur og tilbakekopling fra  $y_k$* 

Det er her vist hvordan PI-regulator fungerer mot modellen. Dette er vist i figurene (7.1.42) og (7.1.43). Vektingen, augmentering og referansen er det samme som forrige eksempel.

$$J(G) = 5,6974 \cdot 10^6$$

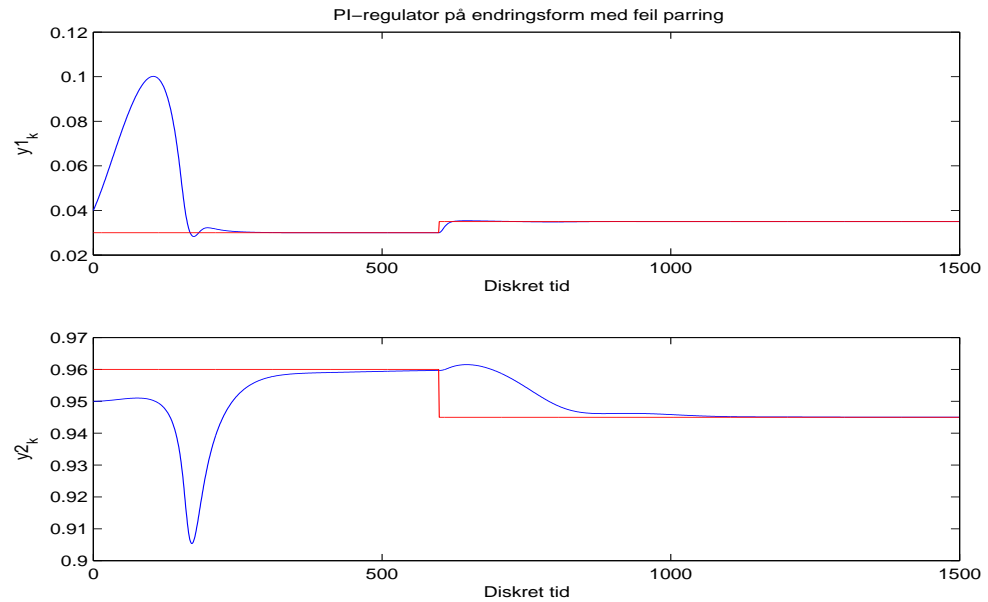
$$\gamma = 1,4494$$

$$G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$

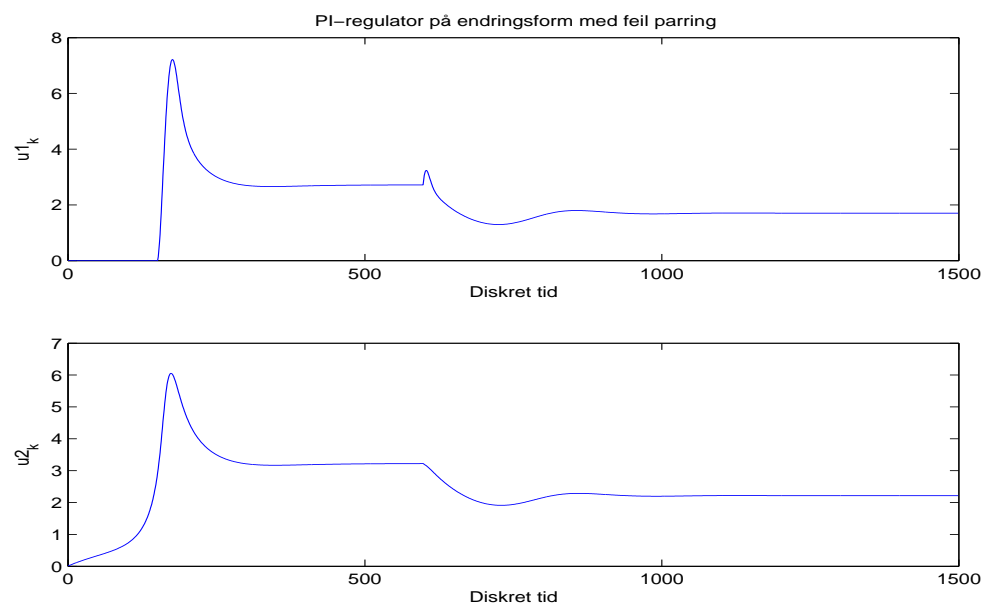
$$G_1 = \begin{bmatrix} -509,0041 & 0 \\ 0 & -89,0354 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -38,7979 & 0 \\ 0 & -0,7993 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,8157 \\ 0,8157 \\ 0,8115 \\ 0,9913 \\ 0,9853 \\ 0,9853 \\ 0,8824 \\ 0,9522 \\ 0,9096 \\ 0,9173 \end{bmatrix}$$



**Figur 7.1.42:** Figuren viser PI-regulator og hvordan utgangene følger referansene.



**Figur 7.1.43:** Figuren viser pådragene til simuleringen.

**Eksempel 7.1.24** *PI-regulator med riktig parring, struktur og tilbakekopling fra  $y_k$* 

Det er her vist hvordan PI-regulator fungerer mot modellen. Dette er vist i figurene (7.1.44) og (7.1.45). Vektingen, augmentering og referansen er det samme som forrige eksempel.

$$J(G) = 4,6168 \cdot 10^6$$

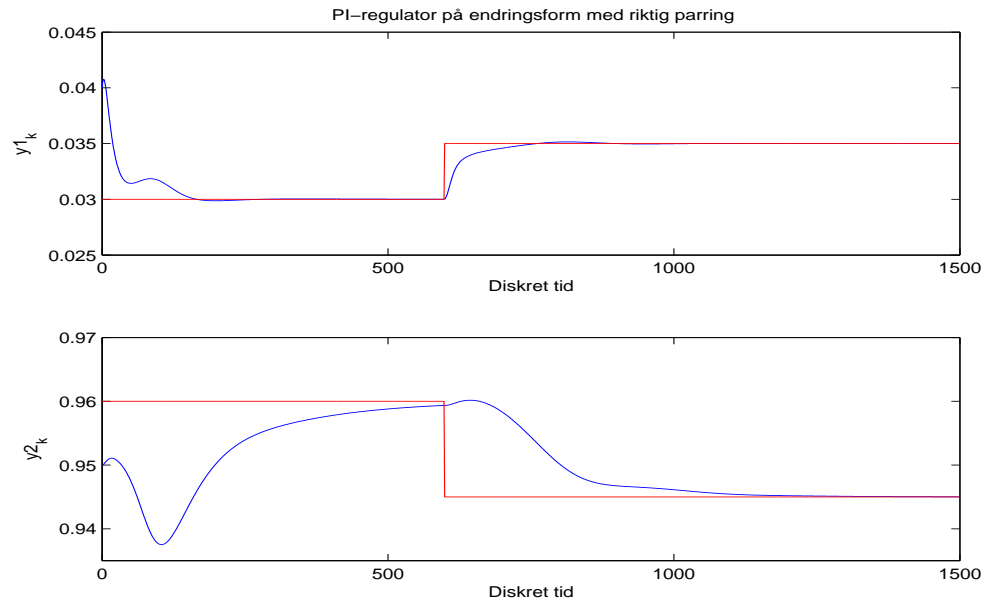
$$\gamma = 1,1745$$

$$G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$

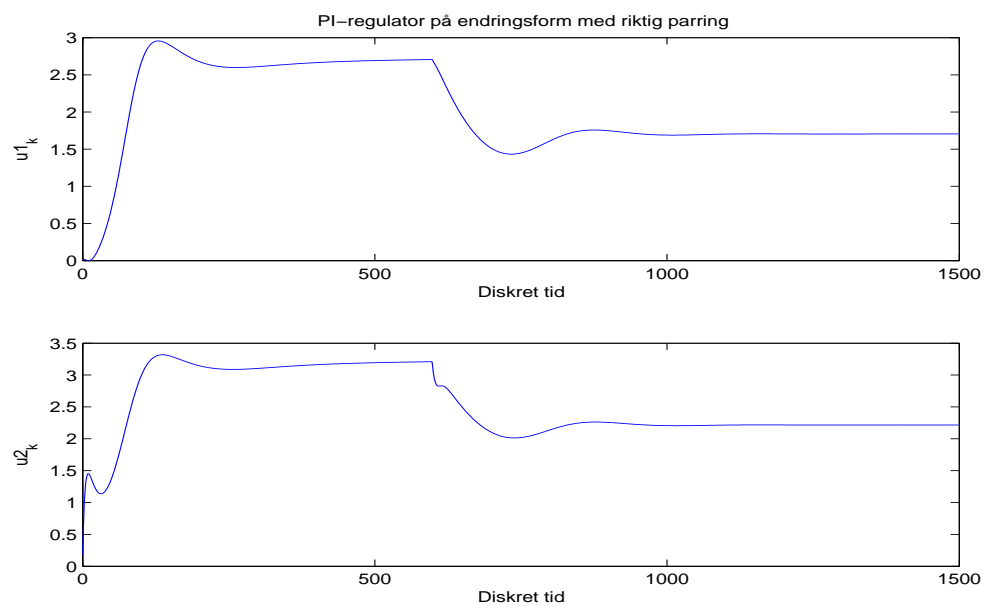
$$G_1 = \begin{bmatrix} 0 & -114,1143 \\ 355,6459 & 0 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 0 & -0,8813 \\ 17,8679 & 0 \end{bmatrix}$$

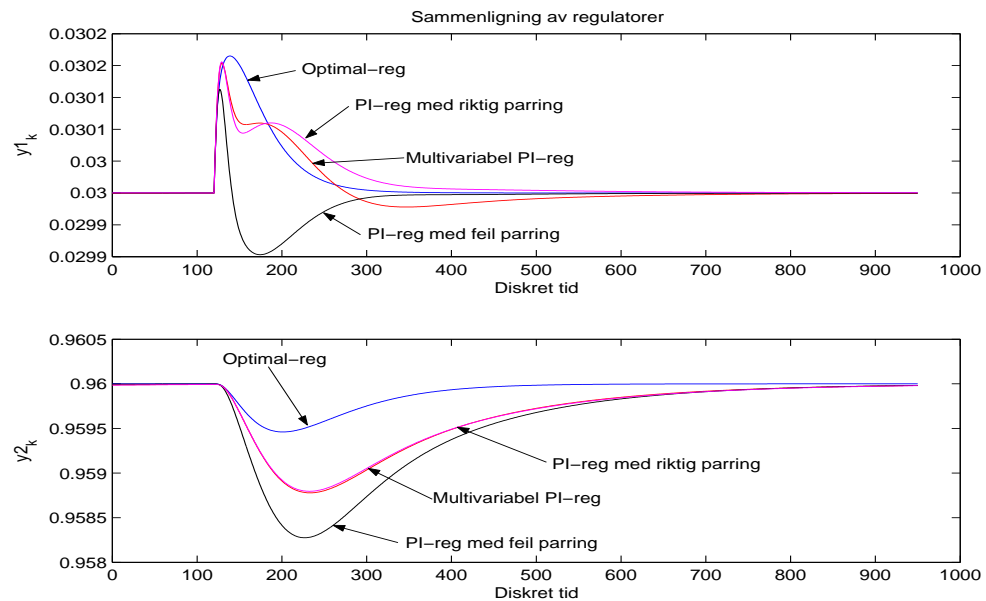
$$|eig(A + BG)| = \begin{bmatrix} 0,8337 \\ 0,8337 \\ 0,8064 \\ 0,9926 \\ 0,9863 \\ 0,9863 \\ 0,8773 \\ 0,9063 \\ 0,9517 \\ 0,9462 \end{bmatrix}$$



**Figur 7.1.44:** Figuren viser PI-regulator og hvordan utgangene følger referansene.



**Figur 7.1.45:** Figuren viser pådragene til simuleringen.



**Figur 7.1.46:** Figuren viser hvordan de ulike regulatorene reagerer på en endring i forstyrrelsen.

#### Eksempel 7.1.25 *Sammenligning av regulatorene med økning av støy*

I denne simuleringen er det satt inn en økning av støy i modellen. Fødestrømstilførselen økes med 10%. Dette er vist i figuren (7.1.46).

### 7.1.5 Vanntank

Dette er en modell laget av "dsr" som beskriver en vanntank. Modellen er som følger

$$x_{k+1} = Ax_k + Bu_k \quad (7.1.3)$$

$$y_k = Dx_k \quad (7.1.4)$$

Denne modellen brukes til å se en regulatorer på endringsform. Det er også sett på dette med tilbakekopling fra  $y_k$ , og hvordan dette får innvirkning på simuleringen. Verdiene i eksemplene (7.1.26) og (7.1.27) genereres ved kjøring av MatLab skript-filen **vanntank.m**

#### Eksempel 7.1.26 *Optimalregulator med integralvirkning*

Det er her vist hvordan en optimalregulator med integralvirkning fungerer mot modellen. Dette er vist i figuren (7.1.47). Referansen gjør her et sprang fra  $\begin{bmatrix} 16, 5252 \end{bmatrix} \rightarrow \begin{bmatrix} 20, 5252 \end{bmatrix}$ . Figurene genereres ved kjøring av MatLab skript-filen **simulering24.m**

$$J(G) = 1,3676 \cdot 10^3$$

$$\gamma = 1$$

$$G_{opt} = \begin{bmatrix} G_1 & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 11,8395 & 13,9676 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -3,8912 \end{bmatrix}$$

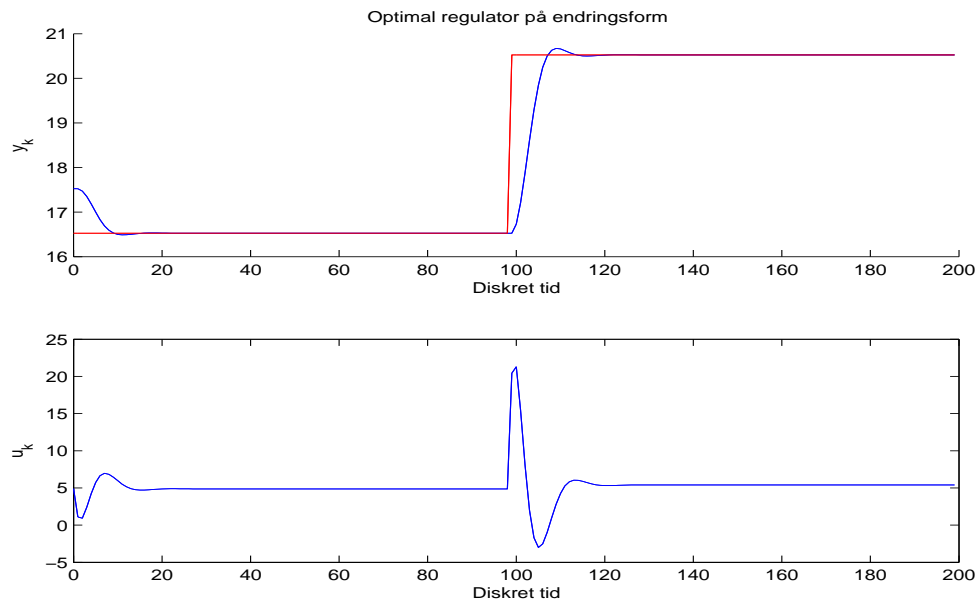
$$|eig(A + BG)| = \begin{bmatrix} 0,7516 \\ 0,7516 \\ 0,6508 \end{bmatrix}$$

Vektingen som er brukt er:

$$Q = 100D^T D, Q_z = 40 \text{ og } P = I$$

For å få modellen på endringsform er augmenteringen som er brukt følgende:

$$\begin{bmatrix} \Delta x_{k+1} \\ y_k - r \end{bmatrix} = \begin{bmatrix} A & 0 \\ D & I \end{bmatrix} \begin{bmatrix} \Delta x_k \\ y_{k-1} - r \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \Delta u_k$$



**Figur 7.1.47:** Figuren viser optimalregulator på endringsform og hvordan utgangene følger referansen.

### Eksempel 7.1.27 *Optimalregulator med integralvirkning og tilbakekopling fra $y_k$*

Det er her vist hvordan en optimalregulator med integralvirkning fungerer mot modellen. Dette er vist i figuren (7.1.48). Vektingen, augmentering og referansen er det samme som forrige eksempel. Figurene genereres ved kjøring av MatLab skript-filen *simulering25.m*

$$J(G) = 7,1058 \cdot 10^5$$

$$\gamma = 519,5762$$

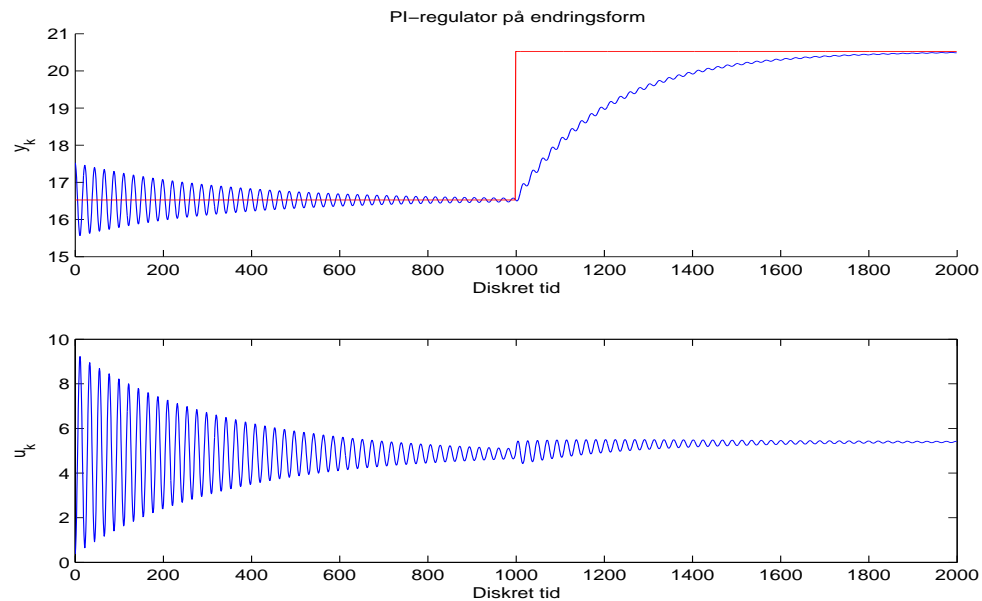
$$G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} -4,5316 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -0,0224 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,9970 \\ 0,9970 \\ 0,9952 \end{bmatrix}$$





**Figur 7.1.48:** Figuren viser PI-regulator på endringsform og hvordan utgangene følger referansen.

### Eksempel 7.1.28 *Optimalregulator med integralvirkning*

Som man kan se av figur (7.1.48) tar det lang tid før pådraget stiller seg inn. Det ble derfor forsøkt med en litt annen vektning for å prøve å finne en bedre regulatorinnstilling. Dette er vist i figur (7.1.49) Referansen gjør her et sprang fra  $\begin{bmatrix} 16,5252 \\ \end{bmatrix} \rightarrow \begin{bmatrix} 20,5252 \\ \end{bmatrix}$

$$J(G) = 3,1918 \cdot 10^5$$

$$\gamma = 1$$

$$G_{opt} = \begin{bmatrix} G_1 & G_2 \end{bmatrix}$$

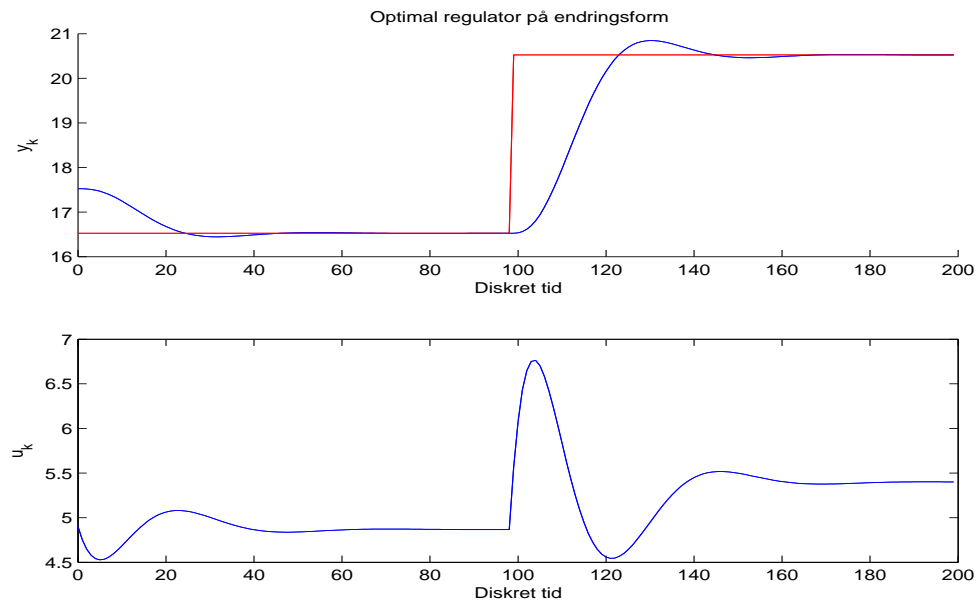
$$G_1 = \begin{bmatrix} 1,2335 & 8,4525 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -0,1749 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,9274 \\ 0,9274 \\ 0,8641 \end{bmatrix}$$

Vektingen er nå endret til:

$$Q = 100D^T D, Q_z = 4 \text{ og } P = 1000$$



**Figur 7.1.49:** Figuren viser optimalregulator og hvordan utgangene følger referansen.

**Eksempel 7.1.29** *Optimalregulator med integralvirkning og tilbakekopling fra  $y_k$*

Som man kan se av figur (7.1.50) stiller den seg raskere inn denne gangen. Men i forhold til optimalregulatoren, går det fortsatt tregt. Vektingen, augmentering og referansen er det samme som forrige eksempel.

$$J(G) = 3,7548 \cdot 10^6$$

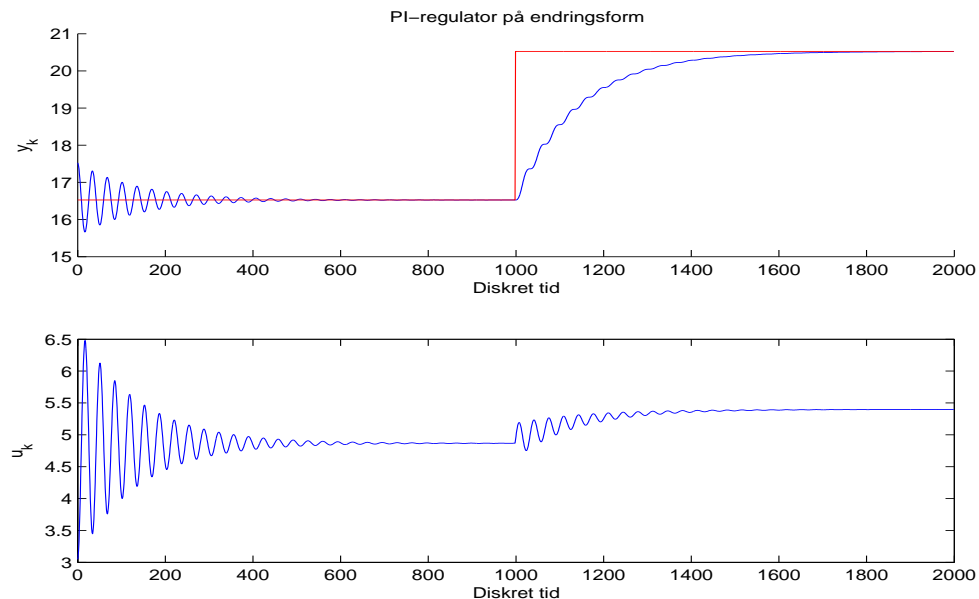
$$\gamma = 11,7639$$

$$G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} -1,8541 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -0,0137 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,9927 \\ 0,9927 \\ 0,9931 \end{bmatrix}$$



**Figur 7.1.50:** Figuren viser optimalregulator og hvordan utgangene følger referansen.

### 7.1.6 Unstable plant

Dette er en kontinuerlig modell som beskriver en ustabil prosess. Regulatoren bygger på den diskrete lineærisede modellen, der den kontinuerlige ble diskretisert på følgende måte

$$\begin{aligned} A_d &= e^{A_c h} \\ B_d &= A_c^{-1}(e^{A_c h} - I)B_c \end{aligned}$$

der  $h$  er samplingsintervallet. Den kontinuerlige lineære modellen er av formen

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Dx \end{aligned}$$

Denne modellen brukes til å se på ulike former for regulatorer. Disse eksemplene er tatt med for å vise resultatene for en prosess som er ustabil. Det er ikke vist figurer, kun verdiene i matrisene.

### Stabiliserende P-regulatorer

Verdiene i eksemplene genereres ved kjøring av MatLab skript-filen **stabilisering.m**. Det er her vist hvordan en ustabil prosess kan stabiliseres med forskjellige

regulatorer typer. I neste eksempel er det vist med forskjellige P-regulatorer. Det er brukt følgende vektning:  $Q = D^T D$  og  $P = I_{2 \times 2}$

**Eksempel 7.1.30 *Optimalregulator***

$$J(G) = 169,9897$$

$$\gamma = 1$$

$$G_{opt} = \begin{bmatrix} -0,0078 & -4,4511 & -0,0646 \\ -1,5260 & -1,4774 & -0,6354 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,9300 \\ 0,8224 \\ 0,8224 \end{bmatrix}$$

**Eksempel 7.1.31 *Multivariabel P-regulator med tilbakekopling fra  $y_k$***

$$J(G) = 177,4023$$

$$\gamma = 1,0436$$

$$G = G_y D$$

$$G_y = \begin{bmatrix} -0,8559 & 0,0621 \\ 0,0606 & 1,5868 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,9342 \\ 0,8381 \\ 0,8381 \end{bmatrix}$$

**Eksempel 7.1.32 *P-regulator med struktur og tilbakekopling fra  $y_k$***

$$J(G) = 177,6780$$

$$\gamma = 1,0452$$

$$G = G_y D$$

$$G_y = \begin{bmatrix} -0,8724 & 0 \\ 0 & 1,5270 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,9368 \\ 0,8380 \\ 0,8380 \end{bmatrix}$$

**Eksempel 7.1.33** *P-regulator med struktur og tilbakekopling fra  $y_k$* 

$$J(G) = 2,2678 \cdot 10^3$$

$$\gamma = 13,3406$$

$$G = G_y D$$

$$G_y = \begin{bmatrix} 0 & 0,0856 \\ -0,3651 & 0 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,9640 \\ 0,9786 \\ 0,9786 \end{bmatrix}$$

**Stabiliserende PI-regulatorer**

Verdiene i eksemplene genereres ved kjøring av MatLab skript-filen **stabiliseringint.m**. Det er her vist hvordan en ustabil prosess kan stabiliseres med forskjellige regulatorer typer. I neste eksempel er det vist med forskjellige PI-regulatorer. Augmenteringen som er brukt er som følger:

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} A & 0 \\ D & I \end{bmatrix} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k$$

Det er brukt følgende vekting:

$$Q = D^T D ,$$

$$Q_z = I_{2 \times 2} \text{ og } P = I_{2 \times 2}$$

**Eksempel 7.1.34** *Optimalregulator med integralvirkning*

$$J(G) = 794,6635$$

$$\gamma = 1$$

$$G_{opt} = \begin{bmatrix} G_1 & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} -0,6192 & -14,7757 & 1,3314 \\ -6,2489 & -5,3753 & -1,1335 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -0,6726 & 0,1089 \\ 0,1008 & 0,8204 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,6875 \\ 0,6875 \\ 0,8530 \\ 0,8530 \\ 0,8333 \end{bmatrix}$$

**Eksempel 7.1.35** *Multivariabel PI-regulator og tilbakekopling fra  $y_k$*

$$J(G) = 820,7772$$

$$\gamma = 1,0329$$

$$G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} -2,8063 & 0,7970 \\ 0,1873 & 6,3199 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -0,6557 & -0,1510 \\ 0,1002 & 0,6445 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,8584 \\ 0,8584 \\ 0,8846 \\ 0,6844 \\ 0,6844 \end{bmatrix}$$

**Eksempel 7.1.36** *PI-regulator med struktur og tilbakekopling fra  $y_k$*

$$J(G) = 831,6410$$

$$\gamma = 1,0465$$

$$G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} -3,0381 & 0 \\ 0 & 5,9029 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} -0,6711 & 0 \\ 0 & 0,5749 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,6588 \\ 0,6588 \\ 0,8589 \\ 0,8589 \\ 0,8724 \end{bmatrix}$$

**Eksempel 7.1.37** *PI-regulator med struktur og tilbakekopling fra  $y_k$*

$$J(G) = 2,2006 \cdot 10^4$$

$$\gamma = 27,6921$$

$$G = \begin{bmatrix} G_1 D & G_2 \end{bmatrix}$$
$$G_1 = \begin{bmatrix} 0 & 1,9969 \\ 12,9711 & 0 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 0 & 0,0315 \\ 0,0644 & 0 \end{bmatrix}$$

$$|eig(A + BG)| = \begin{bmatrix} 0,6419 \\ 0,9155 \\ 0,9155 \\ 0,9787 \\ 0,9950 \end{bmatrix}$$

# Kapittel 8

## Diskusjon

### 8.1 Optimal løsning

Løsninger av optimalreguleringsproblemet er det flere av. I denne rapporten er det løst som et optimaliseringsproblem. Dette gir mer frihet slik at løsningen kan formuleres med en spesifikk struktur og med tilbakekopling fra  $y_k$ .

Under selve optimaliseringen er  $r = 0$ . Dette medfører at løsningen egentlig er optimal med  $r = 0$  selv om regulatoren kan brukes med  $r \neq 0$ . Det kan vises i utledningen av det optimale pådrag at med  $r \neq 0$  fremkommer det et foroverkoplingssignal  $h$ , se Di Ruscio (2004). Det er for å unngå dette, at  $r$  ikke er med i selve optimaliseringen.

### 8.2 Syntese av PI-regulatorer

Ved design av PI-regulatorer for multivariable systemer kommer dette med parring av variable inn i bildet. Hvilken inngang skal brukes til å styre de forskjellige utgangene. Dette er et meget viktig valg å ta.

Metoden kan være nyttig å ta i bruk for eksempel på en destillasjonskolonne. Det er ikke enkelt å stille inn en kolonne med to PI-regulatorer. Man må kanskje sette den ene regulatoren i manuell, mens det gjøres innstillinger på den andre. Dette kan være tidkrevende. Men med en modell tilgjengelig kan det være fort gjort. Det forutsetter en god modell, og at prosessen arbeider rundt arbeidspunktet.



## 8.3 Eksemplene

Det er valgt forskjellige eksempler for å vise hvordan regulatordesignmetoden kan brukes. Flesteparten av systemene er uten direktevirkning, det vil se med  $E = 0$ . Selv om algoritmen takler også dette, er det som regel når tilstandsrommodellen blir generert ved bruk av en systemidentifikasjonmetode at  $E \neq 0$ .

## 8.4 Stabiliseringsproblemet

For å finne stabile startverdier til optimalregulatorer med tilbakekopling fra  $x_k$  kunne man så lenge  $A$ -matrisen var stabil og ingen integralvirkning var i bruk, velge startverdien  $G_0 = 0$ . Dette fordi det sikret en stabil lukket prosess  $A + BG_0$ . Dette gjaldt også med tilbakekopling fra  $y_k$ . Deretter var det opp til optimaliseringsalgoritmen til å finne løsningen. Det ble brukt MatLab funksjonen "fmin-search" for å løse problemet.

Med integralvirkning byr det på problemer for å finne en stabil startverdi. Den augmenterte modellen med positiv tilbakekopling ser slik ut

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \overbrace{\begin{bmatrix} A & 0 \\ D & I \end{bmatrix}}^{\tilde{A}} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \overbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}^{\tilde{B}} u_k \quad (8.4.1)$$

Dette medførte at  $\tilde{A}$  fikk egenverdier som lå på enhetssirkelen. Ved å velge  $G_0 = 0$  fikk man feilmelding fra MatLab funksjonen "dlyap" som løste Lyapunovligningen.

For å finne en løsning som gjorde at det lukkede systemet ble stabilt ble det gjort en liten transformasjon av  $\tilde{A}$ . Den augmenterte modellen ble seende slik ut

$$\begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \overbrace{\begin{bmatrix} A & 0 \\ D & I\epsilon \end{bmatrix}}^{\tilde{A}} \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \overbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}^{\tilde{B}} u_k \quad (8.4.2)$$

Der  $\epsilon$  kunne variere fra 0,90 til 0,95. Dette medførte at hvis  $A$  var en stabil matrise ville  $\tilde{A}$  ha alle egenverdiene på innsiden av enhetssirkelen. Dermed var det nok å velge  $G_0 = 0$ . For å sikre at løsningen skulle konvergere ble det brukt MatLab

funksjonen "fmincon" til dette. Det ble også satt på begrensninger på løsningen slik at

$$C(\theta) \leq 0 \quad (8.4.3)$$

$$L_B \leq \theta \leq U_B \quad (8.4.4)$$

Dette var særlig nyttig dersom det var mange variable som skulle optimaliseres. Når løsningen hadde konvertert ble denne sjekket for stabilitet med  $\epsilon = 1$ . Var denne stabil ble dette startverdien som ble brukt for å finne den endelige optimale løsningen. Til denne endelige optimaliseringen ble det brukt MatLab funksjonen "fminsearch".

Når det var mange variable som skulle optimaliseres, ble det med tilbakekopling fra  $x_k$ , brukt svaret til MatLab funksjonen "dlqr" som startverdi. I enkelte eksempler var det så mange parametre at en optimalisering ble særdeles vanskelig og tidkrevende. Dette var da en perfekt startverdi og løsningen konverterte raskt.

Fremgangsmetoden beskrevet over fungerte tilfredsstillende på de fleste eksemplene som er vist i denne rapporten. Men den forutsetter at  $A$ -matrisen er stabil. Hvis den ikke er stabil vil man få problemer å initialisere  $G_0$ .

Det har blitt sett på to måter å stabilisere ustabile prosesser. Med den ene metoden velges en  $\beta$  slik at det lukkede systemet  $A + BG_0 - \beta I$  er stabilt. Og til slutt  $\beta = 0$  når en stabiliserende  $G_0$  er funnet.

Algoritmen sjekker på absoluttverdiene til egenverdiene, og  $\beta$  beregnes ut fra dette. Men bare  $\beta$  velges slik at det lukkede systemet er stabilt gjør det ikke noe om  $\beta$  også får et bidrag av den imaginære delen av egenverdien.

Ved å trekke fra  $\beta$  i Lyapunov-ligningen, er det bare realdelen av egenverdiene som flytter seg. Det kan tenkes at den imaginære delen av egenverdien fortsatt vil føre til at absoluttverdien vil være på utsiden av enhetsirkelen. Dette har det ikke blitt sett på i dette prosjektet.

Det å velge en passende  $d\beta$  krevde litt prøving og feiling. Men når først dette var gjort, konverterte løsningen meget kjapt. Denne metoden ble testet på systemer både med integral- og direktevirkning, og løsningen konverterte. Metoden fungerte tilfredsstillende, og  $\beta$  gikk raskt mot 0.

Med den andre metoden, innfører man  $\delta$  i kriteriet og verdien beveger seg mot 1.

Denne metoden fungerte også bra. Og med riktig valg av  $d\delta$  fant algoritmen til slutt en stabiliserende regulator.

Det krevde litt prøving og feiling å finne en riktig  $d\delta$  slik at løsningen ikke hoppet for raskt avgårde. Men man måtte heller ikke velge for liten verdi. Med litt forskjellige verdier på  $d\delta$  konvergente alle løsningene. Metoden ble testet på de samme systemene som metoden med  $\beta$ .

Det er en sammenheng mellom  $\beta$  og  $\delta$ . Ved å betrakte

$$\delta A + BG = A + BG - \beta \quad (8.4.5)$$

kan det vises at det er en lineær sammenheng mellom disse.

$$\delta I = I - \beta A^{-1} \quad (8.4.6)$$

$$\beta I = (I - \delta I)A \quad (8.4.7)$$

Fra ligning (8.4.6) ser man når  $\delta = 1$  er  $\beta = 0$ . Og fra ligning (8.4.7) ser man når  $\beta = 0$  er  $\delta = 1$ . Dette resultatet er som forventet.

## 8.5 Bruk av systemidentifikasjonsmetoder

For å kunne beregne optimale pådrag, enten om de er generert av en PI-regulator eller en optimal regulator, kreves det en diskret modell av systemet. I denne rapporten er det sett på modeller av typen tilstandsrommodell. Denne modelltypen er en vanlig måte å representere innholdet i en modell på. Andre modelltyper kan være ARMAX-modeller eller impulsresponsmodeller. Disse modellene kan lages på forskjellige måter.

Man kan modellere prosesser ved hjelp av for eksempel massebalanser eller energibalanser. Dette gir ofte opphav til ulineære modeller. Derfor må disse lineæriseres rundt et arbeidspunkt. Dette er imidlertid en ganske tidkrevende prosess, dersom man starter helt fra bunn av. Utvikling av komplekse modeller er ikke en enkel affære. Det krever erfaring og innsikt. Og dessuten kreves det uttesting av parameterne i modellen mot prosessen. I en slik metode går det med mye tid og man er nødt til å vite nok om prosessen slik at man ikke utelater nødvendige tilstander eller parametre.

Men det finnes andre metoder, som for brukeren lager en raskere vei til en modell. Dette kan være metoder som "dsr" eller "n4sid", som lager en tilstandsrommodell. Det disse metodene krever er en tidsserie av data. Tidsserien består da inngangs- og utgangsdata. Det som er kjekt med dette er tiden man bruker på å utvikle modellen. Dette kan gå fort, dersom man har tidsseriene tilgjengelig. Men også disse metodene krever innsikt og viten om hva man holder på med. Man må forholde seg til ting som systemets orden, og andre parametre som for eksempel i "dsr", "L"- og "J"-parameterne. Disse metodene krever også sitt av brukeren. Og det er viktig at tidsseriene inneholder nok pertubasjon rundt arbeidspunktet slik at en god modell kan beregnes.

En ting som er viktig å huske på er dersom arbeidspunktet forandres, for eksempel med en endring av settpunktet, gjør dette sitt til at ikke modellen stemmer så godt overens med prosessen. Dette medfører også at regulator-parameterne ikke lenger er optimalt innstilt. Hva dette medfører for hver enkelt prosess er ikke godt å si. Men det er en ting som er svært viktig å være klar over dersom man skal regulere en ustabil prosess. I en prosess der det kan være ønskelig å bruke teknikken som er beskrevet i denne rapporten, vil det være naturlig om denne metoden ble utvidet videre til å gå rekursivt samtidig med prosessen. Dette krever også en kontinuerlig oppdatering av prosessen med for eksempel funksjonen "rdsr", som er en rekursiv "dsr". Dermed kan optimale PI-parametre beregnes underveis.

# Kapittel 9

## Konklusjon

Det er laget og implementert algoritmer i MatLab versjon 6.5 (R13) som genererer desentraliserte og suboptimale regulatorer for diskrete systemer. Løsningene finnes ved hjelp av optimaliseringsrutiner. Man har valget mellom å velge tilbakekopling fra tilstands- eller utgangsvektoren. Det er også mulig å velge struktur på regulatormatrisene. Dette har blitt gjort for systemer både med og uten direktevirkning i tilstandsrommodellen. Ved å bruke tilbakekopling fra utgangsvektoren er det blitt laget "optimale" pådrag generert av  $y_k$ . Ved å legge inn integralvirkning er det også blitt laget "optimale" PI-regulatorer. Det har blitt sett både på PI-regulator på standard- og endringsform.

Det er også laget to algoritmer som sikrer en stabil løsning for hvert steg i optimaliseringen. Disse kan brukes på ustabile prosesser, slik at løsningen konvergerer til et minimum.

Det har blitt diskutert muligheten til å bruke en systemidentifikasjonsmetode som for eksempel "dsr" til å estimere den diskrete tilstandsrommodellen. Det konkluderes med at disse metodene er "raskere", i den forstand at modellen kan lages kun ved hjelp av tidsserier med data. Og at de egner seg til bruk når prosessen ligger rundt arbeidspunktet.

På grunn av tidsnød har det ikke vært tid til å presentere de viktigste resultatene i en artikkel.

# Referanser

- Anderson, B. D. O. og Moore, J. B. (1971). *Linear Optimal Control*. Prentice-Hall, ISBN 0-13-638651-2
- Di Ruscio, D. (2004). *Optimal Model Based Control: System Analysis and Design*. Forelesningsnotater, Institutt for prosessregulering, Avdeling for teknologiske fag, Høgskolen i Telemark
- Di Ruscio, D. (2003a). *Subspace System Identification: Theory and applications*. Forelesningsnotater, Institutt for prosessregulering, Avdeling for teknologiske fag, Høgskolen i Telemark
- Di Ruscio, D. (2003b). *Discrete LQ optimal control with integral action*. Forelesningsnotat, Institutt for prosessregulering, Avdeling for teknologiske fag, Høgskolen i Telemark
- Di Ruscio, D. (2002). *System-teori, Tilstandsromanalyse og prosessregulering*. Forelesningsnotater, Institutt for prosessregulering, Avdeling for teknologiske fag, Høgskolen i Telemark
- Di Ruscio, D. (1993). *Topics in Model Based Control with Application to the Thermo Mechanical Pulping Process*. Dr. ing. thesis, Department of Engineering Cybernetics, The Norwegian Institute of Technology
- Di Ruscio, D. og Balchen, J. G. (1990). *An algorithm for design of decentralized suboptimal controllers with a specified structure*. Artikkel, Modeling, identification and control, Vol 11, No. 3, 155-167
- Maciejowski, J. M. (1989). *Multivariable Feedback Design*. Addison-Wesley Publishers Ltd., ISBN 0-201-18243-2

Skilbred, O. C., Nakjem, N. A., Nåden, C. og Thovsland, M. (2004). Regulering av totank-modell. Rapport i faget A2802 Prediktiv regulering med implementering, Institutt for prosessregulering, Avdeling for teknologiske fag, Høgskolen i Telemark

# Tillegg A

## Modeller

### A.1 SISO

Denne er hentet fra Di Ruscio (2004). Det er ingen oppgitte stasjonærverdier for denne modellen. Den diskrete modellen er av formen

$$x_{k+1} = Ax_k + Bu_k \quad (\text{A.1.1})$$

$$y_k = Dx_k + Eu_k \quad (\text{A.1.2})$$

der

$$A = 0.9 \quad (\text{A.1.3})$$

$$B = 0.5 \quad (\text{A.1.4})$$

$$D = 1 \quad (\text{A.1.5})$$

$$E = -1 \quad (\text{A.1.6})$$

### A.2 AIRC

Denne er hentet fra Maciejowski (1989). Den beskriver en lineærisert modell av vertikalplandynamikken til et fly. Det er ingen oppgitte stasjonærverdier for denne modellen. Den er diskretisert med en samplingstid  $h = 0,05$ . Det er blitt brukt følgende måte å diskretisere på

$$A_d = e^{A_c h} \quad (\text{A.2.1})$$

$$B_d = A_c^{-1}(e^{A_c h} - I)B_c \quad (\text{A.2.2})$$



Modellen er av formen

$$\dot{x} = Ax + Bu \quad (\text{A.2.3})$$

$$y = Dx \quad (\text{A.2.4})$$

der

$$A = \begin{bmatrix} 0 & 0 & 1,320 & 0 & -1 \\ 0 & -0,0538 & -0,1712 & 0 & 0,0705 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0,0485 & 0 & -0,8556 & -1,013 \\ 0 & -0,2909 & 0 & 1,0532 & -0,6859 \end{bmatrix} \quad (\text{A.2.5})$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ -0,120 & 1 & 0 \\ 0 & 0 & 0 \\ 4,4190 & 0 & -1,665 \\ 1,5750 & 0 & -0,0732 \end{bmatrix} \quad (\text{A.2.6})$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (\text{A.2.7})$$

## A.3 Destillasjonskolonne

Dette er hentet fra Di Ruscio og Balchen (1990). Det er hentet rett fra teksten og er derfor skrevet på engelsk. Modellen som lineæriseres er kontinuerlig. Den er diskretisert med en samplingstid  $h = 0,1$ . Det er blitt brukt følgende måte å diskretisere på

$$A_d = e^{A_c h} \quad (\text{A.3.1})$$

$$B_d = A_c^{-1}(e^{A_c h} - I)B_c \quad (\text{A.3.2})$$

Assumptions

- Constant molar flow
- No vapor holdup (immediate vapor response)

- Vapor-Liquid Equilibrium (VLE) and perfect mixing on all trays

Columns constants

- Number of equilibrium (theoretical) trays excluding the reboiler  $N = 6$
- Feed tray location  $N_F = 3$
- Liquid in reboiler  $M_{0B} = 10$  mol
- Liquid holdup on each tray  $M_{0i} = 5$  mol,  $i = 1, \dots, N$
- Liquid holdup in accumulator  $M_{0D} = 10$  mol
- Relative volatility  $\alpha = 2,993$
- Hydraulic tray constant  $K = 30 \frac{1}{min}$

Control input the system

- Reflux  $R = 2,0 \frac{mol}{min}$
- Vapor flow  $V = 2,5 \frac{mol}{min}$

Disturbances the system

- Feed flow  $F = 1,0 \frac{mol}{min}$
- Mole fraction of light component in feed  $x_F = 0,5$

Dette er en lineærisert modell av formen

$$\dot{x} = Ax + Bu + Cv \quad (\text{A.3.3})$$

$$y = Dx \quad (\text{A.3.4})$$

For å få til dette har det blitt lineærisert rundt følgende arbeidspunkter som er gitt i teksten over:

$$u_s = \begin{bmatrix} 2,0 \\ 2,5 \end{bmatrix} \quad (\text{A.3.5})$$

$$v_s = \begin{bmatrix} 1,0 \\ 0,5 \end{bmatrix} \quad (\text{A.3.6})$$

$$x_s = \begin{bmatrix} 0,04 & 0,099 & 0,213 & 0,379 & 0,569 & 0,757 & 0,889 & 0,960 \end{bmatrix}^T \quad (\text{A.3.7})$$

Vektorene som er avikene fra stasjonærverdiene er definert som følger:

$$x = (dx_1, \dots, dx_{N+1})^T, u = (dR, dV)^T, v = (dF, dx_F)^T$$

Dette gir da følgende matriser:

$$A = \begin{bmatrix} -0,69 & 0,30 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1,26 & -1,61 & 0,59 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1,02 & -1,31 & 0,59 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,72 & -1,06 & 0,39 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,48 & -0,72 & 0,39 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,32 & -0,63 & 0,39 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,23 & -0,59 & 0,39 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,10 & -0,25 \end{bmatrix} \quad (\text{A.3.8})$$

$$B = \begin{bmatrix} 0,006 & -0,007 \\ 0,022 & -0,027 \\ 0,033 & -0,039 \\ 0,037 & -0,039 \\ 0,037 & -0,029 \\ 0,026 & -0,021 \\ 0,014 & -0,011 \\ 0 & 0 \end{bmatrix} \quad (\text{A.3.9})$$

$$C = \begin{bmatrix} 0,0059 & 0 \\ 0,0224 & 0 \\ 0,0327 & 0 \\ 0,0236 & 0 \\ 0 & 0,1961 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (\text{A.3.10})$$

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{A.3.11})$$

## A.4 Example 3

Dette er en diskret modell av formen som er hentet fra Di Ruscio (2003a). Det er ingen oppgitte stasjonærverdier for denne modellen.

$$x_{k+1} = Ax_k + Bu_k \quad (\text{A.4.1})$$

$$y_k = Dx_k \quad (\text{A.4.2})$$

der

$$A = \begin{bmatrix} 1,5 & 1 & 0,1 \\ -0,7 & 0 & 0,1 \\ 0 & 0 & 0,85 \end{bmatrix} \quad (\text{A.4.3})$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \quad (\text{A.4.4})$$

$$D = \begin{bmatrix} 3 & 0 & -0,6 \\ 0 & 1 & 1 \end{bmatrix} \quad (\text{A.4.5})$$

## A.5 Vanntank

Denne modellen er hentet fra et prosjekt i faget A2802 Prediktiv regulering med implementering som undervises på masterstudiet Kybernetikk og industriell IT på Høgskolen i Telemark, avd for teknologiske fag, se Skilbred, Nakjem, Nåden og Thovsland (2004). Den er laget ved hjelp ”dsr”, brukt mot en totankmodell. Modellen er laget rundt stasjonærverdiene

$$u_s = 5 \quad (\text{A.5.1})$$

$$y_s = 17,5252 \quad (\text{A.5.2})$$

Modellen er av formen

$$x_{k+1} = Ax_k + Bu_k \quad (\text{A.5.3})$$

$$y_k = Dx_k \quad (\text{A.5.4})$$

Det er denne modellen som er simulert.  $u_s$  og  $y_s$  er da plussert på slik at plottene skal se riktige ut. Matrisene er som følger:

$$A = \begin{bmatrix} 0,9821 & 1,2064 \\ -0,0017 & 0,9870 \end{bmatrix} \quad (\text{A.5.5})$$

$$B = \begin{bmatrix} -0,0514 \\ -0,0241 \end{bmatrix} \quad (\text{A.5.6})$$

$$D = \begin{bmatrix} -0,5870 & 0,7023 \end{bmatrix} \quad (\text{A.5.7})$$

## A.6 Unstable plant

Dette er en kontinuerlig modell av formen som er hentet fra Di Ruscio (1990). Denne diskretiseres med  $h = 0,05$ . Diskretiseringen som er brukt er som følger

$$A_d = e^{A_c h} \quad (\text{A.6.1})$$

$$B_d = A_c^{-1}(e^{A_c h} - I)B_c \quad (\text{A.6.2})$$

$$\dot{x} = Ax + Bu \quad (\text{A.6.3})$$

$$y = Dx \quad (\text{A.6.4})$$

der

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 2 & -1 & -2 \end{bmatrix} \quad (\text{A.6.5})$$

$$B = \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (\text{A.6.6})$$

$$D = \begin{bmatrix} 0 & 5 & -1 \\ -1 & -1 & 0 \end{bmatrix} \quad (\text{A.6.7})$$

# Tillegg B

## Oversikt over innhold på CD

### B.1 Innhold

Det er vedlagt en del eksempler med MatLab-skript på CD'en. Disse omhandler noen utvalgte eksempler. Rapporten ligger også på CD'en. Hvert av eksemplene har en hovedfil som må kjøres. Denne filen står i parentes i listen under. Eksemplene ligger i forskjellige kataloger, og det anbefales å kopiere hele katalogen inn på PC'en før eksemplene kjøres. I hver av hovedfilene kan brukeren velge hvilken regulator som skal bli generert. Dette sees i hver enkelt hovedfil. Med eksemplet stabilisering av ustabil prosess er det lagt med begge metodene som inneholder stabilisering.

#### Oversikt over eksempler

- Destillasjonskolonne med standard PI-regulator (**Destkol.m**)
- SISO-prosess med direktevirkning (**SISO.m**)
- Vanntank med PI-regulator på endringsform (**vanntank.m**)
- Stabilisering av ustabil prosess
  1. Uten integralvirkning (**stabilisering.m**)
  2. Med integralvirkning (**stabiliseringint.m**)

Hver av disse eksemplene er vist i kapittel 7. Der er verdiene skrevet inn, og plottene til de ulike eksemplene vist.

# Tillegg C

## Oppgavebeskrivelse

### C.1 Oppgaven

På de neste sidene ligger oppgavebeskrivelsen.



**Høgskolen i Telemark**

Avdeling for teknologiske fag

## F4203 Hovedoppgave

**Oppgavens tittel: Design av desentraliserte og suboptimale regulatorer for diskrete systemer**

**For student:** Ole Christian Skilbred

**Faglærer (ved HiT):** David Di Ruscio

**Biveileder(e):**

**Utfyllende oppgavetekst:**

1. Sett deg inn i lineær kvadratisk (LQ) optimal regulering for diskrete systemer og spesielt LQ optimal regulering med integralvirkning og kryssledd i kriteriet.
2. Formuler løsningen som et optimaliseringsproblem der Riccatiligningen omskrives til en diskret Lyapunovligning og den optimale løsningen finnes som et optimaliseringsproblem. Sammenlign denne løsningen med den tradisjonelle løsningen og illustrer med eksempler.
3. Vis hvordan løsningen i punkt 2 kan benyttes til å finne suboptimale regulatorer av typen  $u_k = Gx_k$  og  $u_k = G y_k$  der regulatormatrisene kan ha spesifisert struktur. Algoritmen skal og kunne benyttes til syntese av PI-regulatorer for multivariable systemer.
4. Undersøk muligheten for å formulere problemet på endringsform slik at man på denne måten kan få integralvirkning i den resulterende regulatoren.
5. Dersom tiden tillater det så forsøk å presentere de viktigste resultatene i en kort artikkel på ca. 4-6 sider i samarbeid med faglærer. Denne bør skrives på engelsk.

Punktene over skal dokumenteres og illustreres med eksempler. Diskuter og muligheten for å bruke en systemidentifikasjonsmetode som f. eks. DSR til å estimere den diskrete tilstandsrommodellen som er utgangspunktet for regulatordesignmetodene.

**Bakgrunn for oppgaven:**

En lignende algoritme som beskrevet i teksten over er tidligere utviklet for kontinuerlige systemer. Et poeng med oppgaven er å utvikle en algoritme for design av suboptimale regulatorer for diskrete systemer.

I forbindelse med emnet Avansert reguleringsteknikk er det utviklet en metode for design av LQ optimale regulatorer med integralvirkning for diskrete systemer. Metoden er dokumentert i et notat som er tilgjengelig. En sammenligning av den optimale regulatoren viser at den er svært lik en tradisjonell PI-regulator. Imidlertid blir pådraget en funksjon av endringen i tilstandsvektoren

**Praktiske forhold (arbeidssted, arbeidsform, utstyr, m.m.):** Må ha tilgang til PC med MATLAB.





**Formell utkvittering av oppgaven (med endelig oppgavetekst, som definert ovenfor):**

Studentens signatur og dato:

Faglærers signatur og dato:

*Oppgaven regnes som tatt ut (og påbegynt) 14.01.2005 med mindre det er gjort en spesiell avtale (skriftlig) mellom studenten og avdelingen om en annen utaksdato (og en annen innleveringsdato enn 29.05. 2005).*