

Reglerteknik AK

Laboration 3

Positionsreglering av elastiskt kopplade massor med nätverksfördröjningar

Institutionen för reglerteknik
Lunds tekniska högskola
Senast ändrad september 2021

1. Inledning

I tidigare laborationer har vi studerat en process som har varit relativt enkel och som har kunnat regleras tillfredsställande med en enkel PID-regulator. I denna laboration ska vi titta på en process som är något mer komplicerad och som kräver en mer avancerad regulator. Regulatorn som vi kommer att använda är baserad på tillståndsåterkoppling och skattning av samtliga tillstånd från en mätbar utsignal. Vi ska även undersöka inverkan av tidsfördröjningar i reglerkretsen, samt experimentera med dödtidskompensering.

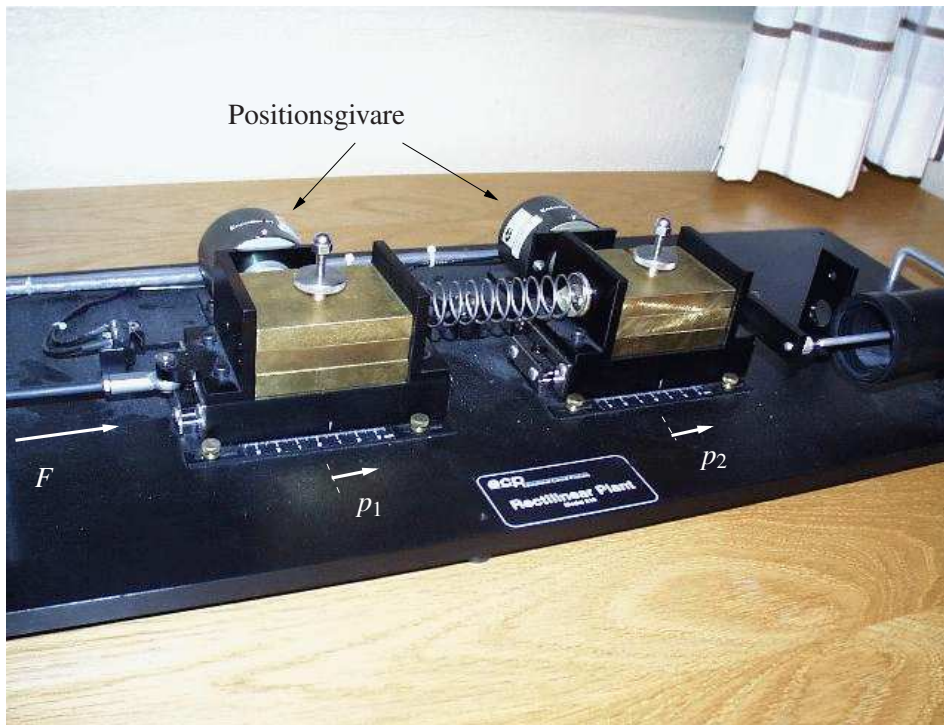
Förberedelser

- Läs igenom denna handledning.
- Repetera föreläsningarna om tillståndsåterkoppling, Kalmanfiltrering och dödtidskompensering. Du ska vid laborationens början kunna svara på följande frågor:
 - Vad innebär tillståndsåterkoppling? Förklara i ord!
 - Varför använder man ofta en observerare tillsammans med tillståndsåterkoppling?
 - Hur fungerar Otto Smith-regulatorn? Förklara i ord!
- Lös förberedelseuppgifterna 4.2, 5.2, 6.1 och 7.1.
- Studera MATLAB-filerna `define_process.m`, `design1.m`, `design2.m` och `design3.m` som finns sist i handledningen. Försök att relatera innehållet i filerna till uppgifterna i laborationshandledningen.

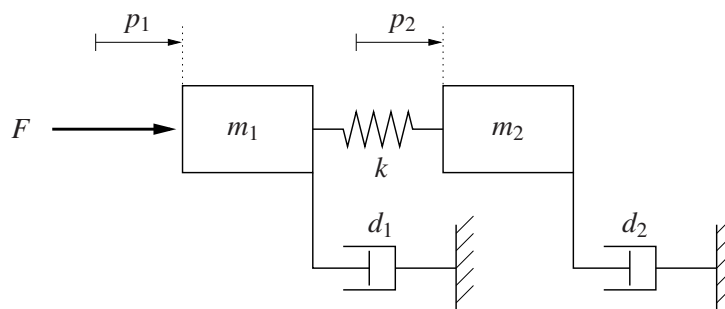
2. Processen

I figur 1 visas en bild av det flexibla servo som ska regleras. Processen består av två massor som är ihopkopplade med en fjäder. En principskiss av processen visas i figur 2. Massan på den ena sidan om fjädern kan flyttas med en motor. Vi kallar denna sida för motorsidan och den andra för lastsidan. Lägga märke till att av dämpningarna d_1 och d_2 är det endast d_2 som syns som en urskiljbar enhet på den verkliga processen. Dämpningar på andra ställen i processen kan dock "bakas ihop" och modelleras enligt figur 2.

Syftet med regleringen är att reglera positionen p_2 hos massan på lastsidan. På laborationsprocessen kan båda massornas positioner mätas, men i laborationen kommer vi att anta att vi endast kan mäta p_2 . De övriga tillstånden kommer att skattas med ett Kalmanfilter.



Figur 1 Det flexibla servot.



Figur 2 Principskiss av processen.

Kopplingsschema

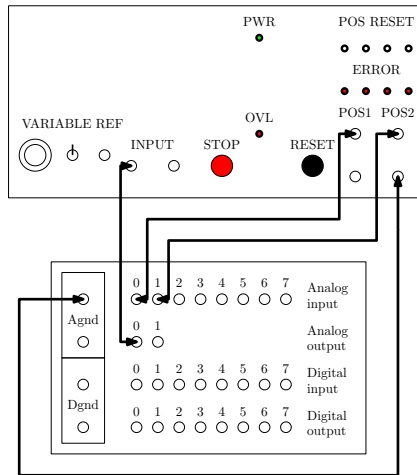
D/A-omvandlaren kopplas till processen enligt figur 3.

En linjär modell av processen

De två massorna har massan m_1 och m_2 . Kopplingen mellan massorna har fjäderkonstanten k . Dämpningen för massorna är d_1 respektive d_2 .

Den ena massan drivs av en DC-motor med borstlös rotor som drivs av en strömåterkopplad förstärkare. Dynamik i motor och förstärkare försummas. Motorns kraft blir proportionell mot förstärkarens inspanning u enligt

$$F = k_m u$$



Figur 3 Kopplingsschema för processen.

En kraftbalans ger följande dynamiska modell:

$$m_1 \frac{d^2 p_1}{dt^2} = -d_1 \frac{dp_1}{dt} - k(p_1 - p_2) + F$$

$$m_2 \frac{d^2 p_2}{dt^2} = -d_2 \frac{dp_2}{dt} + k(p_1 - p_2)$$

Introducera tillståndsvektorn $x = \begin{pmatrix} p_1 & \dot{p}_1 & p_2 & \dot{p}_2 \end{pmatrix}^T$. Processen kan då skrivas på tillståndsform som

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{1}$$

där

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m_1} & -\frac{d_1}{m_1} & \frac{k}{m_1} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{k}{m_2} & 0 & -\frac{k}{m_2} & -\frac{d_2}{m_2} \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \frac{k_m}{m_1} \\ 0 \\ 0 \end{pmatrix} \tag{2}$$

$$C = \begin{pmatrix} 0 & 0 & k_y & 0 \end{pmatrix}$$

För en verklig labbprocess har man mätt och uppskattat följande värden på konstanter och koefficienter

$$\begin{aligned} m_1 &= 2.29 \text{ kg} \\ m_2 &= 2.044 \text{ kg} \\ d_1 &= 3.12 \text{ N/m/s} \\ d_2 &= 3.73 \text{ N/m/s} \\ k &= 400 \text{ N/m} \\ k_m &= 2.96 \text{ N/V} \\ k_y &= 280 \text{ V/m} \end{aligned}$$

Undersökning av processen

Uppgift 2.1 Logga in och starta MATLAB genom att öppna en terminal och skriva `cd; ./labstart`. Definiera processen `Gp` genom att köra filen `define_process.m`:

```
>> define_process
```

Räkna ut systemets poler med

```
>> pole(Gp)
```

Var ligger polerna? Är systemet stabilt? Asymptotiskt stabilt? Knuffa till den verkliga processen med handen och studera stabiliteten. Stämmer det verkliga beteendet med analysen?

Uppgift 2.2 Rita processens Bodediagram med

```
>> bode(Gp)
```

Om amplituden i Bodediagrammet visas i decibel kan detta ändras genom att köra kommandot

```
>> ctrlpref
```

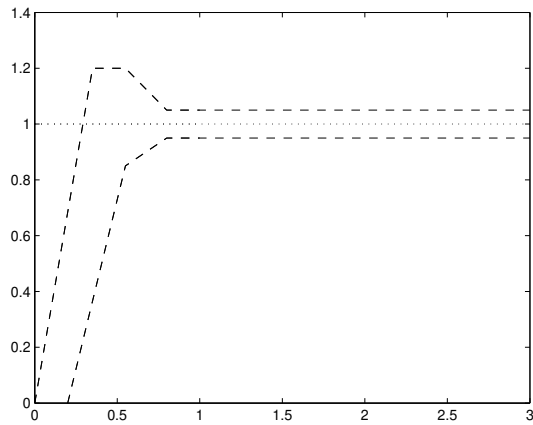
och ändra "Magnitude" från dB till absolute.

Notera resonanstoppen i amplitudkurvan. Vid vilken frekvens ligger den? Vilket samband finns mellan resonanstoppens läge och polernas läge? Skaka om processen för hand och försök uppskatta resonansfrekvensen. Stämmer detta med modellen?

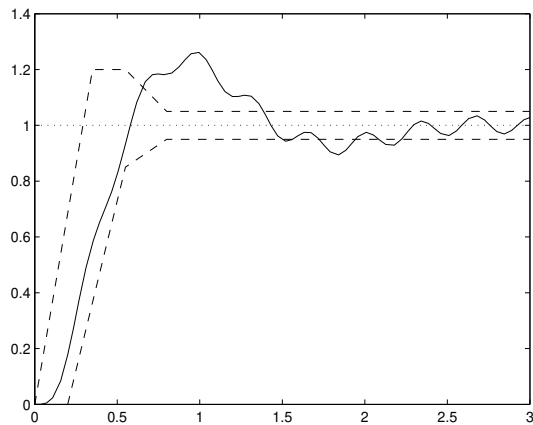
3. Specifikationer

Specifikationerna är de krav som det reglerade systemet ska uppfylla. I detta fall har vi valt att specificera det slutna systemet i tidsplanet. Vi önskar ett väl dämpat stegsvar med en stigtid på mellan 0.2 och 0.4 sekunder, se figur 4. Samtidigt får inte styrsignalen vara för våldsam, eftersom motorn i så fall slits. För att få viss robusthet mot modellfel och tidsfördröjningar vill vi dessutom ha en fasmarginal på minst 30° .

Processens oscillativa egenskaper gör det omöjligt att uppfylla specifikationerna med en PID-regulator. Ett försök med en PD-regulator visas i figur 5. Stegsvaret är snabbt nog, men regulatormen klarar inte av att dämpa ut svängningarna.



Figur 4 Stegsvaret ska ligga inom det markerade området.



Figur 5 Stegsvaret för en PD-regulator med $K = 0.07$ och $T_d = 0.1$.

4. Tillståndsåterkoppling

För att kunna förändra processens dynamik godtyckligt använder vi tillståndsåterkoppling. Först kontrollerar vi att systemet är styrbart.

Uppgift 4.1 Ställ upp och beräkna rangen (d.v.s. antalet linjärt oberoende kolonner) av systemets styrbarhetsmatris:

```
>> Wc = [ ... ]
>> rank(Wc)
```

Notera att matriserna A och B redan finns definierade.

Är systemet styrbart?

Om vi till en början antar att hela tillståndsvektorn x kan mätas så kan vi använda styrlagen

$$u = -Kx + k_r r \quad (3)$$

Här är K är en radvektor, r är referensvärdet och k_r är en skalär, se figur 6.

Uppgift 4.2 (Förberedelseuppgift) Visa att det slutna systemet kan skrivas på formen

$$\begin{aligned} \dot{x} &= A_{cl}x + B_{cl}r \\ y &= C_{cl}x \end{aligned}$$

när styrlagen (3) används på processen (1). Vad blir A_{cl} , B_{cl} och C_{cl} ? Hur ska k_r väljas för att den statiska förstärkningen från r till y ska bli 1? Uttryck svaren med matrisbeteckningarna från tillståndsmodellen (1), dvs. använd **inte** de explicita uttrycken från (2). \diamond

Med hjälp av tillståndsåterkopplingen kan vi placera det slutna systemets poler godtyckligt. I praktiken finns det dock begränsningar, bl.a. i form av begränsad styrauktoritet. Lite förenklat kan man säga att ju mer man flyttar en pol från dess ursprungliga läge, desto större styrsignal kommer att behövas.

Den önskade polplaceringen kan uttryckas med hjälp av ett fjärde ordningens karakteristiskt polynom (se figur 7):

$$(s^2 + 2\zeta_a\omega_a s + \omega_a^2)(s^2 + 2\zeta_b\omega_b s + \omega_b^2)$$

Givet en polplacering kan återkopplingsvektorn K sedan enkelt beräknas med kommandot `place` (se `design1.m`).

Uppgift 4.3 Editera filen `design1.m` genom att i MATLAB köra:

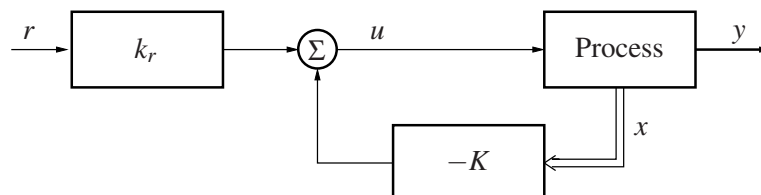
```
>> edit design1.m
```

Placera det slutna systemets poler genom att fylla i lämpliga värden på ω_a , ζ_a , ω_b och ζ_b . Beräkna sedan regulatorn genom att köra filen i MATLAB:

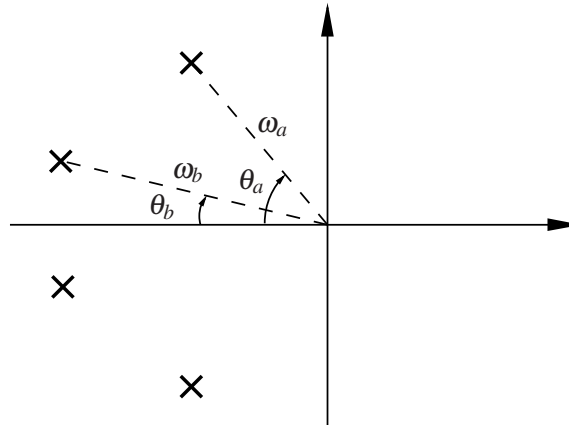
```
>> design1
```

Öppna Simulink-modellen `modell1.mdl` genom att skriva

```
>> modell1
```



Figur 6 Tillståndsåterkoppling.



Figur 7 Polplacering enligt det karakteristiska polynomet $(s^2 + 2\zeta_a\omega_a s + \omega_a^2)(s^2 + 2\zeta_b\omega_b s + \omega_b^2)$, där $\zeta_a = \cos \theta_a$ och $\zeta_b = \cos \theta_b$.

Simulera det återkopplade systemet och se om det uppfyller specifikationen på stegsvaret. Kommandot `specs` kan användas efter en simulering för att jämföra resultatet mot specifikationen i figur 3. Ändra designparametrarna och upprepa proceduren tills önskat beteende erhålls.

Vad är en lämplig polplacering för tillståndsåterkopplingen?

Uppgift 4.4 Varför prövar vi inte denna regulator på den riktiga processen?

5. Observerare

I praktiken kan vi inte mäta processens alla tillstånd utan bara utsignalen y . Vi skaffar oss då istället en modell av processen och matar modellen med samma insignal. Avvikelsen mellan modellens utsignal och processens utsignal används för att korrigera modellens tillstånd så att dessa närmar sig processens. En sådan anordning kallas *observerare* eller *Kalmanfilter*.

Observeraren beskrivs av

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu + L(y - C\hat{x}) \quad (4)$$

där \hat{x} betecknar de skattade tillstånden. Kolonnvektorn L kan väljas så att observerarens tillstånd närmar sig processens tillstånd godtyckligt snabbt, förutsatt att systemet är observerbart.

Uppgift 5.1 Ställ upp och beräkna rangen av systemets observerbarhetsmatris:

```
>> Wo = [ ... ]
>> rank(Wo)
```

Är systemet observerbart?

Med hjälp av Kalmanfiltret kan vi återkoppla från skattade tillstånd istället för från riktiga tillstånd, se figur 8. Den nya styrlagen blir

$$u = -K\hat{x} + k_r r \quad (5)$$

Uppgift 5.2 (Förberedelseuppgift) Utgående från (4) och (5), visa att regulatoren baserad på återkoppling från skattade tillstånd kan skrivas på formen

$$\begin{aligned} \frac{d\hat{x}}{dt} &= A_R \hat{x} + B_{R_y} y + B_{R_r} r \\ u &= C_R \hat{x} + D_{R_y} y + D_{R_r} r \end{aligned}$$

Vad blir A_R , B_{R_y} , B_{R_r} , C_R , D_{R_y} och D_{R_r} ? Uttryck svaren med matrisbetckningarna från (1), (4) och (5). \diamond

Eftersom processen har fyra tillstånd så kommer observeraren också att ha fyra tillstånd. Vi specificerar observerarens poler enligt det karakteristiska polynomet

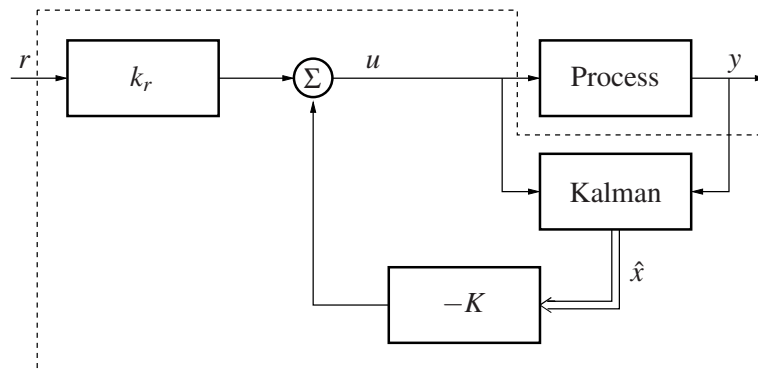
$$(s^2 + 2\zeta_c \omega_c s + \omega_c^2)(s^2 + 2\zeta_d \omega_d s + \omega_d^2)$$

Ett lämpligt val av poler beror bl.a. på mängden mätbrus, storleken på modellfelet och om initialtillståndet är känt. Snabba poler ger hög förstärkning av mätbrus, medan långsamma poler ger långsam konvergens hos skattningen. Som utgångspunkt kan man använda tumregeln som säger att observerarens poler placeras så att de blir 1.5–2 gånger snabbare än tillståndsåterkopplings poler.

Uppgift 5.3 Editera filen `design2.m` och för först in de värden på ω_a , ζ_a , ω_b och ζ_b ni kom fram till i avsnitt 4. Fyll sedan i några lämpliga värden på ω_c , ζ_c , ω_d och ζ_d . Beräkna den fullständiga regulatoren (tillståndsåterkoppling + observerare) genom att köra filen:

```
>> design2
```

Öppna sedan Simulink-modellen `model2.mdl` genom att skriva



Figur 8 Återkoppling från skattade tillstånd. Regulatoren består av allt innanför den streckade linjen.


```
>> model2
```

Simulera det återkopplade systemet och se om det uppfyller specifikationerna. Ändra designparametrarna och upprepa proceduren tills önskat beteende erhålls. (Ändra om nödvändigt även polplaceringen för tillståndsåterkopplingen.) Plotta även kretsöverföringens Bodediagram med

```
>> margin(Gp*Gr)
```

och kontrollera fasmarginalen. Försök även här att få en fasmarginal på ungefär 40° . Vad är en lämplig polplacering för observeraren?

Uppgift 5.4 Plotta regulatorns Bodediagram med

```
>> bode(Gr)
```

Vilken förstärkning har regulatorn för låga frekvenser? Vad innebär detta för regulatorns förmåga att undertrycka konstanta laststörningar?

Uppgift 5.5 Prova regulatorn på den riktiga processen, se avsnitt A. Hur skiljer sig det riktiga stegsvaret från det simulerade stegsvaret? Vad är det som begränsar prestandan?

Uppgift 5.6 På den verkliga processen kan man faktiskt mäta både x_1 (första massans position) och x_3 (andra massans position), även om vi i laborationen antar att endast x_3 finns tillgänglig för återkoppling. Vi kan dock använda båda mätsignalerna för att undersöka hur väl Kalmanfiltret fungerar. Studera hur väl de skattade tillstånden \hat{x}_1 och \hat{x}_3 stämmer överens med de verkliga processtillstånden x_1 och x_3 . För vilket tillstånd är skattningen bäst?

6. Integralverkan

För att eliminera stationära fel inför vi integralverkan i regulatoren, se figur 9. Integratorn införs som extra tillstånd x_i enligt

$$\begin{aligned}x_i &= \int (r - y) dt \\ \dot{x}_i &= r - y = r - Cx\end{aligned}\quad (6)$$

Om man introducerar den utökade tillståndsvektorn

$$x_e = \begin{pmatrix} x \\ x_i \end{pmatrix}$$

kan det utökade systemet (d.v.s. processen och integratorn) skrivas

$$\begin{aligned}\dot{x}_e &= \underbrace{\begin{pmatrix} A & 0 \\ -C & 0 \end{pmatrix}}_{A_e} x_e + \underbrace{\begin{pmatrix} B \\ 0 \end{pmatrix}}_{B_e} u + \underbrace{\begin{pmatrix} 0 \\ 1 \end{pmatrix}}_{B_r} r \\ y &= \underbrace{\begin{pmatrix} C & 0 \end{pmatrix}}_{C_e} x_e\end{aligned}$$

Om vi tillfälligtvis åter antar att hela tillståndsvektorn kan mätas kan vi återkoppla från både processens tillstånd och integraltillståndet enligt

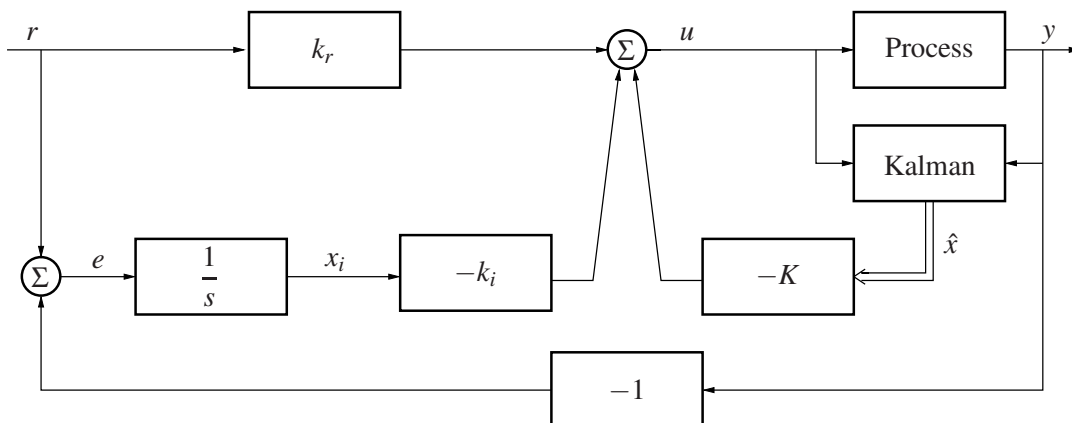
$$u = -Kx - k_i x_i + k_r r = -K_e x_e + k_r r$$

där

$$K_e = \begin{pmatrix} K & k_i \end{pmatrix}$$

Det slutna systemet blir

$$\begin{aligned}\dot{x}_e &= (A_e - B_e K_e) x_e + (B_e k_r + B_r) r \\ y &= C_e x_e\end{aligned}$$



Figur 9 Återkoppling från skattade tillstånd med integralverkan.

Eftersom det utökade systemet har fem tillstånd så specificerar vi nu tillståndsåterkopplingens poler enligt ett femte ordningens karakteristiskt polynom:

$$(s^2 + 2\zeta_a\omega_a s + \omega_a^2)(s^2 + 2\zeta_b\omega_b s + \omega_b^2)(s + \omega_e)$$

Liksom tidigare kan vi i praktiken inte mäta processens tillstånd utan återkopplar istället från de skattade tillstånden och integratorn enligt styrlagen

$$u = -K\hat{x} - k_i x_i + k_r r \quad (7)$$

Uppgift 6.1 (Förberedelseuppgift) Utgående från (4), (6) och (7), visa att regulatorn baserad på tillståndsåterkoppling från skattade tillstånd med integralverkan kan skrivas på formen

$$\begin{aligned} \begin{pmatrix} \frac{d\hat{x}}{dt} \\ \frac{dx_i}{dt} \end{pmatrix} &= \underbrace{\begin{pmatrix} * & * \\ * & * \end{pmatrix}}_{A_R} \begin{pmatrix} \hat{x} \\ x_i \end{pmatrix} + \underbrace{\begin{pmatrix} * \\ * \end{pmatrix}}_{B_{Ry}} y + \underbrace{\begin{pmatrix} * \\ * \end{pmatrix}}_{B_{Rr}} r \\ u &= \underbrace{\begin{pmatrix} * & * \end{pmatrix}}_{C_R} \begin{pmatrix} \hat{x} \\ x_i \end{pmatrix} + \underbrace{\begin{pmatrix} * \\ * \end{pmatrix}}_{D_{Ry}} y + \underbrace{\begin{pmatrix} * \\ * \end{pmatrix}}_{D_{Rr}} r \end{aligned}$$

Vad blir A_R , B_{Ry} , B_{Rr} , C_R , D_{Ry} och D_{Rr} ? ◇

Uppgift 6.2 Editera filen `design3.m` och för först in era tidigare värden på ω_a , ζ_a , ω_b , ζ_b , ω_c , ζ_c , ω_d och ζ_d . Fyll även i ett lämpligt värde på ω_e . Beräkna den fullständiga regulatorn (tillståndsåterkoppling + integrator + observerare) genom att köra filen:

```
>> design3
```

Öppna sedan Simulink-modellen `model3.mdl` genom att skriva

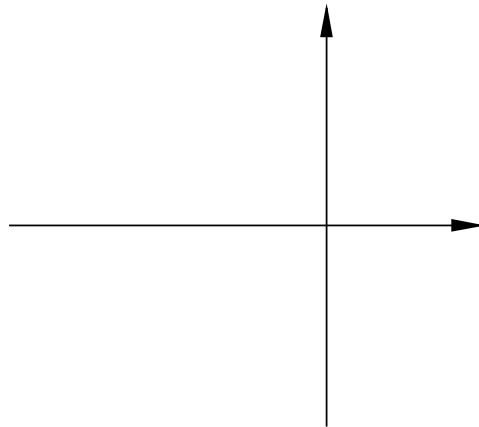
```
>> model3
```

Simulera det återkopplade systemet och se om det uppfyller specifikationerna. Ändra designparametrarna och upprepa proceduren tills stegsvarsspecifikationen är uppfylld. Plotta även kretsöverföringens Bodediagram med

```
>> margin(Gp*Gr)
```

och kontrollera att specifikationen på fasmarginalen är uppfylld, $\varphi_m \geq 30^\circ$. Vad är en lämplig polplacering?

Uppgift 6.3 Rita in det slutna systemets *samtliga* poler i singularitetsdiagrammet nedan:



Uppgift 6.4 Plotta regulatorns Bodediagram med

```
>> bode(Gr)
```

Hur kan man se att regulatorn nu har integralverkan?

Uppgift 6.5 Prova regulatorn på den riktiga processen. Hur skiljer sig resultatet från uppgift 5.5?

När man har integralverkan i regulatorn så behövs inte längre termen $k_r r$ i styrlagen för att få korrekt statisk förstärkning – detta tar integratorn hand om. Istället kan parametern k_r användas för att trimma stegsvaret vid börvärdesändringar. Som vi ser i (7) så innebär $k_r \neq 0$ en direktkoppling mellan referensvärdet och styrsignalen. Ett värde $k_r > 0$ kan alltså användas för att ge processen en extra ”skjuts” vid steg i börvärdet. (Detta kan vara speciellt användbart vid reglering av den verkliga processen, som har mycket friktion.)

Uppgift 6.6 Vilket värde har k_r nu? Ändra värdet på kr i `design3.m` och gör nya experiment på den verkliga processen. Vad är ett lagom värde på k_r ?

7. Reglering med tidsfördröjningar

Moderna reglersystem är ofta distribuerade, så att sensorer, ställdon och regulator befinner sig på olika noder i ett nätverk. Fördelen med distribuerade system är framför allt flexibilitet. En nackdel ur reglerperspektiv är de tidsfördröjningar som uppstår när en reglerkrets sluts över ett nätverk. Om fördröjningarna är långa i förhållande till slutna systemets snabbhet så kan prestandan minska betydligt.

I detta avsnitt fortsätter vi att arbeta med den regulator ni tog fram i förra avsnittet, men lägger till en dödtid i reglerkretsen genom att fördröja styrsignalen.

Uppgift 7.1 Vad är dödtidsmarginalen för ett system med fasmarginalen ϕ_m och skärfrekvensen ω_c ?

Uppgift 7.2 Beräkna dödtidsmarginalen (eng. *delay margin*) för regulatorn ni tog fram i uppgift 6.2 med hjälp av kommandot

```
>> allmargin(Gp*Gr)
```

(Om kretsöverföringens amplitudkurva har flera överkorsningar så kan det finnas flera skärfrekvenser och fasmarginaler. Det korrekta värdet är då det *minsta* värde på dödtidsmarginalen som rapporteras av `allmargin`.)

Reglering med dödtid

Processen som regleras är långsam i förhållande till hastigheten i typiska lokala nätverk för reglering. För att komma upp i fördröjningar som har menlig inverkan på prestandan ska vi istället försöka reglera processen via en omväg.

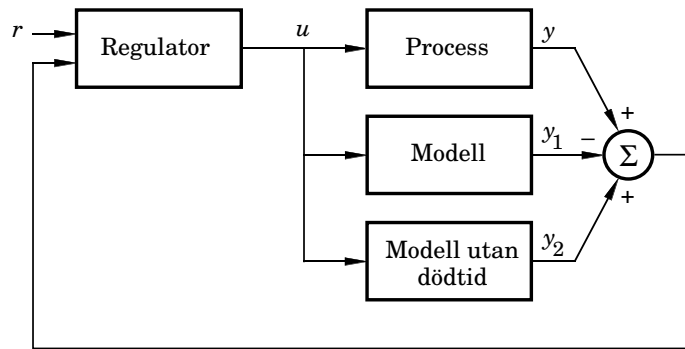
Vi antar nu att processens mätsignal samplas med periodtiden $h = 10$ ms. Mätsignalen fördröjs sedan med hjälp av en modell av nätverksfördröjningar. Den här modellen kan ni själva påverka genom att bestämma hur lång dödtid som ska finnas i ert nätverk. Efter ni har bestämt hur lång dödtid ni vill ha i nätverket kommer modellen att lägga till brus på den valda fördröjningen.

Om dödtiden är känd och någorlunda konstant kan man kompensera för den genom att använda en Otto Smith-regulator, se figur 10. Principen för Otto Smith-regulatorn är att styra processen efter en simulerad modell utan dödtid. Den verkliga utsignalen släcks ut av en simulerad modell med dödtid. Detta kräver så klart en god modell av den verkliga processen. Dessutom krävs det att processmodellen är stabil.

Uppgift 7.3 Öppna Simulink-modellen `mode14.mdl` genom att skriva

```
>> mode14
```

Vi börjar med att undersöka hur prestandan blir utan dödtidskompensering. Mata in några olika dödtider i intervallet 5 ms till 25 ms och simulera systemet. Anteckna resultaten ni ser. Hur långa fördröjningar klarar systemet av utan att bli instabilt? Stämmer resultatet med analysen i uppgift 7.2?



Figur 10 Principen för Otto Smith-regulatorn.

(Samplingen av mätsignalen och utställningen av styrsignalen görs i början på varje sampel. Detta gör att den verkliga fördröjningen kommer att avrundas till närmsta större multipel av 10 ms. Dessutom kan samplingen i sig sägas introducera i genomsnitt en halv sampels extra fördröjning, d.v.s. ytterligare 5 ms dödtid.)

Uppgift 7.4 Mata in en fördröjning på ca 30–50 ms. Aktivera Otto Smith-regulatorn genom att mata in den uppskattade tidsfördröjningen i dödtidsblocket. Hur väl fungerar dödtidskompenseringen?

Uppgift 7.5 Prova även Otto Smith-regulatorn på den riktiga processen. Hur skiljer sig det riktiga stegsvaret från det simulerade stegsvaret?

Uppgift 7.6 Försök att mata in några riktigt långa fördröjningar. Hur långa dödtider kan man kompensera för på den simulerade respektive den verkliga processen?

8. Sammanfattning

Denna sammanfattning är tänkt att samla relevanta frågor som ni ska kunna besvara efter avklarade experiment. Laborationshandledaren kommer att gå igenom er sammanfattning innan ni kan bli godkända på laborationen.

Uppgift 8.1 Nämn minst tre begränsningar hos den verkliga processen som inte fångas av den matematiska modellen.

Uppgift 8.2 Det flexibla servot är en kraftigt resonant (svängig) process. Hur kunde man se detta dels i processens Bodediagram och dels i processens singularitetsdiagram?

Uppgift 8.3 Varför prövade vi aldrig regulatort baserad på ren tillståndsåterkoppling (avsnitt 4) på den riktiga processen?

Uppgift 8.4 Hur kan man använda tillståndsåterkoppling om inte alla tillstånd är mätbara?

Uppgift 8.5 Vid reglering baserad på tillståndsåterkoppling från skattade tillstånd (avsnitt 5), hur många poler har det slutna systemet totalt?

Uppgift 8.6 Hur många tillstånd innehöll regulatort med integralverkan (avsnitt 6)? Vilka?

Uppgift 8.7 Hur kan man se i regulatorns Bodediagram om den har integralverkan eller inte?

Uppgift 8.8 Varför går det inte i praktiken att kompensera för hur långa fördröjningar som helst?

A. Handhavande av processen

På processen finns knappar för att nollställa positionsgivarna. Ställ massorna i position 0 och tryck på RESET och de två knapparna POS. Om massorna åker mot något av stoppblocken så utlöses säkerhetssystemet och strömmen stängs av till motorn. Säkerhetssystemet återställs genom att trycka på RESET-knappen. Detta kan bara göras när insignalen till processen är noll.

Ställ alltid massorna i nollpositionen och nollställ positionsgivarna före varje experiment för att minska transienterna i början av experimentet.

OBS. Motorn som driver massorna är väldigt stark och det är därför viktigt att ni hanterar processen med respekt och tänker på var ni har händerna när ni utför ett experiment.

B. MATLAB-filer

define_process.m

```
% Skapa linjär modell av processen

m1 = 2.29; m2 = 2.044;           % massor
d1 = 3.12; d2 = 3.73;           % dämpningskonstanter
k = 400;                         % fjäderkonstant
km = 2.96;                       % motorkonstant
ky = 280;                         % mätkonstant

A = [0 1 0 0; -k/m1 -d1/m1 k/m1 0; 0 0 0 1; k/m2 0 -k/m2 -d2/m2];
B = [0; km/m1; 0; 0];
C = [0 0 ky 0];
D = 0;

Gp = ss(A,B,C,D);                % skapa tillståndsbeskrivning av processen
```

design1.m — Beräkning av regulator baserad på ren tillståndsåterkoppling

```
% Design av tillståndsåterkoppling

omegaa = ...;                    % ena polparets hastighet
zetaaa = ...;                    % ena polparets dämpning
omegab = ...;                    % andra polparets hastighet
zetaab = ...;                    % andra polparets dämpning

pc = conv([1 2*omegaa*zetaaa omegaa^2],[1 2*omegab*zetaab omegab^2]);
poles1 = roots(pc);

K = place(A,B,poles1);           % beräkna återkopplingsvektorn K

kr = 1/(C*inv(-A+B*K)*B);        % beräkna kr så att statiska
% förstärkningen r->y blir 1
```


design2.m — Beräkning av regulator baserad på tillståndsåterkoppling från observerare

```
% Design av tillståndsåterkoppling

omegaa = ...;           % ena polparets hastighet
zetaaa = ...;          % ena polparets dämpning
omegab = ...;          % andra polparets hastighet
zetab = ...;           % andra polparets dämpning

pc = conv([1 2*omegaa*zetaaa omegaa^2],[1 2*omegab*zetab omegab^2]);
poles1 = roots(pc);

K = place(A,B,poles1); % beräkna återkopplingsvektorn K
kr = 1/(C*inv(-A+B*K)*B); % beräkna kr så att statiska
                           % förstärkningen r->y blir 1

% Design av observerare

omegac = ...;           % ena polparets hastighet
zetaac = ...;          % ena polparets dämpning
omegad = ...;          % andra polparets hastighet
zetad = ...;           % andra polparets dämpning

po = conv([1 2*omegac*zetaac omegac^2],[1 2*omegad*zetad omegad^2]);
poles2 = roots(po);

L = place(A',C',poles2)'; % beräkna Kalmanförstärkningen L

% Beräkning av regulatorn (observerare + tillståndsåterkoppling)

AR = A-B*K-L*C;
BRy = L;
BRr = B*kr;
CR = -K;
DRy = 0;
DRr = kr;

Gr = -ss(AR, BRy, CR, DRy); % överföringsfunktion från -y till u
```

design3.m — Beräkning av regulator baserad på tillståndsåterkoppling från observerare med integralverkan

```
% Design av tillståndsåterkoppling med integralverkan

Ae = [A zeros(4,1); -C 0];           % A-matris för utökade systemet
Be = [B; 0];                         % B-matris för utökade systemet

omegaa = 19;                          % ena polparets hastighet
zetaaa = 0.8;                          % ena polparets dämpning
omegab = 15;                           % andra polparets hastighet
zetaab = 0.8;                           % andra polparets dämpning
omegae = 2;                            % femte polens hastighet

pc = conv([1 2*omegaa*zetaaa omegaa^2],[1 2*omegab*zetaab omegab^2]);
pc = conv(pc, [1 omegae]);
poles1 = roots(pc);

Ke = place(Ae,Be,poles1);             % beräkna återkopplingsvektorn Ke
K = Ke(1:4);
ki = Ke(5);
kr = 0;                                % direktkoppling från referensvärdet

% Design av observerare

omegac = 38;                           % ena polparets hastighet
zetaac = 0.8;                           % ena polparets dämpning
omegad = 30;                            % andra polparets hastighet
zetaad = 0.8;                            % andra polparets dämpning

po = conv([1 2*omegac*zetaac omegac^2],[1 2*omegad*zetaad omegad^2]);
poles2 = roots(po);

L = place(A',C',poles2)';             % beräkna Kalmanförstärkningen L

% Beräkning av regulatorn (observerare + tillståndsåterkoppling med integralverkan)

AR = [A-B*K-L*C -B*ki; zeros(1,4) 0];
BRy = [L; -1];
BRr = [B*kr; 1];
CR = [-K -ki];
DRy = 0;
DRr = kr;

Gr = -ss(AR,BRy,CR,DRy);              % överföringsfunktion från -y till u
```