Laborationer i Processidentifiering

Johansson, Rolf; Holmberg, Ulf; Lilja, Mats; Lundh, Michael

1988

Document Version:
Förlagets slutgiltiga version

Link to publication

Citation for published version (APA):
Johansson, R., Holmberg, U., Lilja, M., & Lundh, M. (1988). *Laborationer i Processidentifiering*. (Research Reports TFRT-3198). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:
4

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

CODEN: LUTFD2/(TFRT-3198)/1-53/(1988)

Laborationer i Processidentifiering

Rolf Johansson
Ulf Holmberg
Mats Lilja
Michael Lundh

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Januari 1988

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
REPORT

Date of issue
January 1988

Document Number
CODEN:LUTFD2/(TFRT-3198)/1-53/(1988)

Author(s)
Rolf Johansson
Ulf Holmberg
Mats Lilja
Michael Lundh

Supervisor

Sponsoring organisation

Title and subtitle
Laborationer i Processidentifiering

Abstract

Five laboratory exercises are presented in this report. The exercises cover frequency analysis, recursive and real time identification, interactive system identification, and process identification as a tool for control design. The material has been developed for a graduate course in system identification.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
Swedish

Number of pages
53

Recipient's notes

Security classification

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

FÖRORD

Föreliggande arbete innehåller fyra laborationer och en terminalövning i processidentifiering. Dessa kurslaborationer ingår i kursen *Processidentifiering*, som givits första gången under höstterminen 1987 såsom reguljär fortsättningskurs och forskarutbildningskurs vid Reglerteknik, LTH. Kursdeltagarna förutsättes ha goda förkunskaper i reglerteori och stokastiska processer. Laborationerna har även med vissa variationer använts inom externkurser, som givits av institutionen för Reglerteknik, LTH.

Följande laborationer ingår:

1. *Frekvensanalys*

   Avsikten med denna laboration är att bestämma frekvenssvaret och Bode--diagrammet för en valfri laboratorieprocess. Laborationen utnyttjar en kommersiell frekvensanalysator med ett överordnat användargränssnitt för bekvämare experimentplanering.

2. *Interaktiv identifiering*

   Denna laboration består av två terminalövningar med Idpac. Det visas hur man kan identifiera system med ML-metoden. Uppgifterna omfattar också exempelvis filhantering och genering av dataserier. I den andra övningen finns uppgifter med spektralanalys, diskret Fouriertransform, korrelationsanalys, koherensanalys och olika sätt att beräkna överföringsfunktioner samt validering.

3. *Rekursiv Identifiering och Adaptiv Reglering.*

   Laborationen avser täcka det avsnitt av kursen, som berör realtidsidentifiering och rekursiva metoder av främst ARMAX-modeller. Mycket uppmärksamhet ägnas åt experimentell observation av parameterskattningarnas noggrannhet och konvergenshastighet. Problem med identifiering i sluten reglerslinga demonstreras. Det nära släktskapet mellan rekursiv identifiering och adaptiv reglering påvisas i de senare, valfria delarna av laborationen.

   Laborationen utnyttjar en regulatorimplementering baserad på IBM-AT med program skrivna i Modula-2. Detta system beskrives detaljerat i Johansson, R. (1987): *Adaptor - A System for Process Identification and Adaptive Control – An Implementation Report*, Rapport 3190, Institutionen för Reglerteknik, LTH, Lund. Laborationen utnyttjar i den här befintliga formen en laboratorieprocess med vattentankar för nivåreglering.

4. *Syntesorienterad identifiering*

   För dimensioneringsuppgifter är man speciellt intresserad av att erhålla hanterliga, lägre ordnings modeller, som stämmer bra i medelhöga och lägre frekvenser. Högre ordnings dynamik och brus liksom olinjäriteter försummas i görligaste mån. Med begränsad modellnoggrannhet blir det ofta lämpligt att välja en robust designmetod. Under laborationen genomföres hela den interaktiva och iterativa arbetsgången från datainsamling och identifiering till regulatordesign och reglering.

Laborationen utnyttjar en laboratorieprocess i form av en pendelupphängd platta, som påverkas av en luftström från en fläkt. Konstruktionen har tillkommit med stark inspiration från B. Schmidtbauers "resonant aerodynamic object", som finns beskriven i Schmidtbauer, B. (1986): *A low-cost development tool for microprocessor based controller design*, IFAC Low Cost Automation, Valencia, Spain, pp.293-298.

5. *Matlab-uppgifter i Processidentifiering*

   Syftet med denna övning är att visa, hur man kan numeriskt kan lösa många problem, som ofta angrips analytiskt i litteraturen. Övningen utgör därmed ett komplement till kursens övningspass. Uppgifterna anknyter främst till övningsuppgifter i Söderström, T. (1984): *Lecture Notes in Identification*, Rapport UPTEC 8468R, Reglerteknik och Systemanalys, Uppsala Universitet.

Jag vill tacka Ulf Holmberg, Mats Lilja och Michael Lundh för bidrag, som till stor del tillkommit under sena nattimmar. Vidare vill jag tacka Karl Johan Åström, Per Hagander och Björn Wittenmark för kloka synpunkter under olika skeden av kursens utveckling.

Den är min förhoppning, att dessa laborationer förmår ge sina utövare inom reglerteknikämnet och annorstädes en starkare experimentell medvetenhet.

Rolf Johansson (Red.)

Frekvensanalys

Michael Lundh

Institutionen för Reglerteknik
Lunds Tekniska Högskola
September 1987

1. INLEDNING

Avsikten med laborationen är att experimentellt bestämma frekvenssvaret $G(i\omega)$ för en process t.ex. bommen.

Utrustning

För att genomföra laborationen finns en Solartron 1250 frekvensanalysator. Denna är ansluten till en Apple dator, för att man skall få ett användarvänligt mätsystem.

Processen

Processen väljer Du själv. Exempel på lämpliga processer är bommen (överföringsfunktionen till vinkelläge) och DC-servot (överföringsfunktionen till vinkelhastighet). Tankarna bör undvikas om ni inte vill hålla på ett helt dygn.

Förberedelser

Läs igenom kapitel 2 (speciellt avsnitt 2.2) och lös de uppgifter som betecknats med *Förberedelseuppgift* i laborationshandledningen. Bestäm även vilken process Du vill göra frekvensanalys på.
Lösningar skall kunna redovisas vid laborationens början, då även några frågor skall besvaras enskilt.

2. METOD

Vid frekvensanalys drivs systemet av en sinusformad insignal $u(t) = u_0 \sin(\omega t)$. Då transienten avklingat erhålls utsignalen $y(t) = |G(i\omega)| u_0 \sin(\omega t + arg G(i\omega))$. Experimentet upprepas sedan för olika frekvenser så att bodediagram kan uppritas.
Det är emellertid svårt att med någon större noggrannhet direkt mäta amplituderna hos y och u samt deras fasskillnad. En bättre metod är att under en viss experimenttid T för varje frekvens ω bilda integralerna

$$Y_s(T) = \int_0^T y(t) sin(\omega t) dt$$
$$Y_c(T) = \int_0^T y(t) cos(\omega t) dt$$

Väljer man experimenttiden som ett helt antal perioder av signalen $u(t)$, dvs $T = 2\pi n/\omega$ erhålls

$$Y_s(T) = \frac{T}{2} u_0 \mathrm{Re} G(i\omega)$$
$$Y_c(T) = \frac{T}{2} u_0 \mathrm{Im} G(i\omega)$$

Ur dessa storheter erhålls amplitud och fasvridning hos $G(i\omega)$.

$$|G(i\omega)| = \frac{2}{T u_0} \sqrt{Y_s^2(T) + Y_c^2(T)}$$
$$\arg G(i\omega) = \arctan \frac{Y_c(T)}{Y_s(T)}$$

Denna korrelationsmetod kan tolkas som bandpassfiltrering av $y(t)$ med ω som passbandets mittfrekvens och bredden på detta är proportionell mot $1/T$.

3. EXPERIMENTFÖRBEREDELSER

Frekvensanalysatorn som används utför beräkningarna ovan. Ett antal experimentbetingelser måste emellertid bestämmas och matas in till mätsystemet.

Frekvensintervall

Bestäm det frekvensintervall som skall undersökas. För att göra detta måste man ha kännedom om systemets tidskonstanter.
DC-servots stigtid är omkring 10 s och bommens omkring 1 s.

Förberedelseuppgift:
Bestäm lämpligt frekvensintervall över 3 dekader.

Mättid

Brusnivån bestämmer hur lång mättid som behövs. Lång mättid ger låg känslighet för brus. Mättiden är ett jämnt antal perioder av testfrekvensen.

Amplitud

Bestäm insignalens amplitud u_0. Välj denna så att systemet är linjärt, dvs inga signaler får bli så stora att de uppnår någon begränsning. Frekvensanalysen kan upprepas för några olika amplituder för att avgöra om systemet är linjärt. Lämplig amplitud för DC-servot är 0.15 V. För bommen 0.5 V.

Väntetid

Bestäm hur länge man skall vänta innan man börjar mäta. Fördröjningen från det att man exciterar systemet med en signal av viss frekvens tills dess att man börjar mäta bestäms av hur snabbt transienterna dör ut. Vid systemets resonansfrekvens kan det ta lång tid.

Förberedelseuppgift:
Skatta hur lång tid hela ert experiment kommer att ta. Om tiden blir orimligt lång modifiera mättid och väntetid

Eliminering av vissa olinjäriteter

Vilofriktion hos ett servo kan vid hastighetsmätning elimineras genom att låta insignalen svänga kring en nivå så att hastigheten alltid är positiv, $u(t) = u_B + u_0 \sin(\omega t)$. Metoden är ej tillämpbar vid lägesmätning. Lägg på en bias på 1.05 V då DC-servots hastighet mäts.

4. EXPERIMENT

Själva frekvensanlysen går till på följande sätt.
Sätt på Solartron 1250. Anslut kanal 1 och generatorn till processens insignal (glöm ej jordanslutningen). Anslut kanal 2 till processens utsignal.
Sätt i disketten och slå på Appledatorn. Programmet är självstartande. Svara N på första frågan.
Operatörskommunikationen är menybaserad. Från huvudmenyn väljer man någon av följande tre undermenyer.

4 FRA-SETUP (setupmeny)

8 ALTER/CREATE M.D.B (utför mätning)

9 PLOT M.D.B (plotta bodediagram)

Setupmeny

I denna meny bestäms alla experimentbetingelser. Följande undermenyer används.

Generator (1) Här väljer man amplitud (u_0), bias (u_B) och vågform (sinus). Frekvensen väljs ej här!

Analyzer (5) Här sätter man parametrar för integrationen. Man kan integrera över ett visst antal perioder eller en viss tid. Man bestämmer hur lång fördröjningen skall vara innan man börjar integrera. HARMONIC sätts till 1. Välj mätområden. COUPLING för kanalerna är DC och vi utnyttjar anslutningarna på framsidan.

Sweep (6) Här väljer man de frekvenser analysen skall göras för. Välj max. och min. frekvens. Välj logaritmisk inkrementering av frekvensen och antalet steg per dekad. Slå RETURN på frågan om linjärt inkrement.

Display (7) Här defineras vilken kanal u respektive y anslutits till. Välj även vilka koordinater som skall beräknas (LOG(R),THETA).

Utför mätningen

Under meny rubriken ALTER/CREATE M.D.B. (Measurement Data Base) utföres analysen. Sätt nivän 0 dB till 1 och välj svepet logaritmiskt ökande. Få generatorn att stoppa vid 0 grader.

Plotta Bodediagram

Slutligen återstår bara att plotta bodediagrammet. Kommandomoden erhålls vid RETURN. Plotta även argumentkurvan (kommando 6). Skärmens aktuella bild skrivs ut på skrivare med ESC Q. Gör inte detta i onödan eftersom det tar lång tid.

5. VALIDERING

Utnyttja det erhållna bodediagrammet för att bestämma en PID-regulator. Prova denna regulator !

6. AVSLUTANDE DISKUSSION

Då bodediagrammet erhållits kan man bestämma några karakteristika för processen.

- Vilken ordning är processen av?
- Var ligger dess poler?
- Vad är statiska förstärkningen?

Interaktiv Identifiering

Ulf Holmberg
Mats Lilja

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Oktober 1987

LUNDS TEKNISKA HÖGSKOLA
REGLERTEKNIK
AUTOMATIC CONTROL
LUND INSTITUTE OF TECHNOLOGY

IDPAC-övning 1

Ulf Holmberg

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Box 118
221 00 Lund

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S–221 00 Lund SWEDEN

Visitors:
Reglerteknik
Ole Römers väg 1
Lund

Telephone:
+46 46–10 70 00
Telex: S–33533 luniver
Telefax: +46 46–13 81 18

Den här övningen skall visa hur man kan använda *Idpac* för identifiering av system med framför allt Maximum Likelihood metoden. Men för att snabbt komma in i *Idpac's* värld har det även blandats in uppgifter som generering av dataserier, plottningar, globala parametrar, filhantering m. m..

Det kan vara illustrativt att själv generera data från ett känt system. Vi kan då i slutändan jämföra våra parameterskattningar med det verkliga systemets parametrar. I systemfilen *process* finns följande ARMAX-modell.

$$A(q^{-1}) = 1 - 2.851q^{-1} + 2.717q^{-2} - 0.856q^{-3}$$
$$B(q^{-1}) = q^{-1}(1 + q^{-1} + q^{-2})$$
$$C(q^{-1}) = 1 + 0.7q^{-1} + 0.2q^{-2}$$

Samplingsintervallet är 0.1s och den stokastiska processen $\{e(t)\}$ är normalfördelat vitt brus med medelvärde 0 och standardavvikelse 1.

Uppgift

Generera in- och utsignal till ovanstående system och försök identifiera fram systemet från dessa dataserier med $ML - metoden$.

Lösningsguide:

Generering av dataserier: Använd kommandot *insi* för att generera processerna $\{u(t)\}$ och $\{e(t)\}$. Låt experimenttiden vara $30s$ d.v.s. välj antalet data till 300. Insignalen $u(t)$ kan vi låta vara en $PRBS$-signal med grundfrekvens $1s$. Sätt den globala parametern $tick.$ = samplingsintervallet.

$$Let \quad tick. = 0.1$$

I filhuvudet på alla datafiler står en massa information om filen, bl.a. antalet kolonner, antalet rader, samplingsintervall, filtyp, kommando som genererat filen m. m. . Denna information kommer du åt med kommandot *fhead*. Bekanta dig redan nu med det kommandot.

Ett annat kommando man har stor nytta av är *plot*. Gå in i grafisk mod på terminalen och skriv

$$plot(300) \quad u$$

Talet inom parantes anger hur många punkter som plottas upp på en gång. Detta kan också bestämmas med en global parameter *nplx.* som från början har värdet 100. Om vi vill ha tidskalan uttryckt i sekunder istället för antalet data kan vi använda *turn*. Pröva följande sekvens.

$$turn \quad time \quad s$$
$$let \quad nplx. = 30$$
$$plot \quad u/e$$

Vi är nu färdiga för att generera utsignalen till systemet. Detta gör vi med kommandot *dsim*.

$$dsim \quad y < process \quad u \quad e \quad 300$$

ML-identifiering: Innan vi nu börjar med själva identifieringen måste vi lägga in-utsignalparet (u, y) i en och samma fil. Detta gör vi med *move*.

$$move \quad wrk < u$$
$$move \quad wrk(2) < y$$

Filen *wrk* består nu av 2 kolonner, den första är insignal och den andra utsignal. Antag nu att vi inte vet någonting om systemet. Vi börjar därför med att ML-identifiera en första ordningens ARMAX-modell.

$$ml \quad p1 < wrk \quad 1$$

Förlustfunktion, AIC samt parameterskattningar kommer nu visas på skärmen. Skriv upp dessa värden och jämför med skattningar av högre ordning.

$$ml \quad p2 < wrk \quad 2$$
$$ml \quad p3 < wrk \quad 3$$
$$ml \quad p4 < wrk \quad 4$$

Vilken modellordning skall vi välja på grundval av AIC? Stämmer dessa parametrar med den verkliga processens parametrar?

Ett annat bra sätt att se om det finns mer information att krama ur datan är att titta på prediktionsfelen eller som de också kallas residualerna. Om allt står rätt till skall dessa uppföra sig som störningsprocessen $\{e(t)\}$, d.v.s. som vitt brus. I annat fall finns det mer dynamik att modellera och anledning att pröva en högre ordningens modell. Titta på residualerna för ovanstående identifieringsresultat.

$$resid \quad e1 < p1 \quad wrk \quad 100$$

.

.

Eftersom modellen som genererade datan är känd kan vi räkna ut dess utsignal då störningsprocessen $\{e(t)\}$ är noll.

$$deter \quad yd < process \quad u$$

Om vi gör detsamma med vårt skattade system av tredje ordningen

$$deter \quad y3 < p3 \quad u$$

så kan vi jämföra de båda kurvorna på samma graf

$$plot \quad yd \quad y3/u$$

Bias: ARMAX-modellen gäller normalt bara för avvikelser kring ett jämviktsläge. När man utför ett verkligt experiment är det därför lämligt att låta styrsignalen vara konstant under en relativt lång tid före och efter själva experimentet så att jämviktsläget blir definierat. Undersök hur pass allvarligt t.ex. en bias påverkar skattningarna. För att lägga till en konstant term till y kan du använda *sclop*.

$$sclop \quad y < y + 10000$$

Gör ML-identifiering som tidigare. Vilken ordning väljer du nu?

Polynomiska trender i data kan enkelt tas bort m. h. a. kommandot *trend*. Eftersom en konstant är ett nollte ordningens polynom tar vi bort en bias på följande sätt:

$$trend \quad y < y \quad 0$$

Ta bort biasen med trend-kommandot och gör en ny identifiering. Jämför resultatet med det du fick utan biasen. Varför blir det inte lika bra nu?

Outliers:(störningar, transmissionsfel, konversionsfel m.m.) Med kommandot *plmag* kan man ändra enskilda punkter i en dataserie. Det finns ett numera ett trevligt sätt att använda detta kommando. Man kan nämligen utnyttja ett nytt kommando *gin* (som subkommando i *plmag* och i macroform *Galter*), som gör det möjligt att använda hårkors. Skriv *help gin* för mer information. Använd nu *plmag* och macrot *Galter* för att ändra data i $\{y(t)\}$ till s.k. ourliers. Utför sedan en ML-identifiering för att se hur pass känliga skattningarna är för ourliers. Pröva även att sätta den globala parametern *liml.* = 1. Detta innebär att residualerna begränsas till $3 * \lambda$ (λ = standardavv. för residualerna).

Extrauppgift — *Minstakvadratmetoden*

I systemfilen *process2* finns följande ARMAX-modell.

$$\begin{aligned} A(q^{-1}) &= 1 + 0.5q^{-1} - 0.25q^{-3} \\ B(q^{-1}) &= q^{-1}(1 + q^{-1} + q^{-2}) \\ C(q^{-1}) &= 1 + 0.7q^{-1} + 0.2q^{-2} \end{aligned}$$

Använd $\{u(t)\}$ och $\{e(t)\}$ som du har redan för att generera utsignal $\{y(t)\}$ till *process*2.

Vi skall nu se vad som händer om man försöker använda minstakvadratmetoden MK på ett system av formen

$$A(q^{-1})y = B(q^{-1})u + C(q^{-1})e$$

MK räknar som om brusprocessen vore vitt brus, d.v.s. som om systemet istället vore

$$\widehat{A}(q^{-1})y = \widehat{B}(q^{-1})u + e$$

Parametrarna kommer därför konvergera mot

$$\begin{aligned} \widehat{A}(q^{-1}) &= A(q^{-1})/C(q^{-1}) = 1 - 0.2q^{-1} - 0.06q^{-2} - 0.17q^{-3}\dots \\ \widehat{B}(q^{-1}) &= B(q^{-1})/C(q^{-1}) = q^{-1}(1 + 0.3q^{-1} + 0.59q^{-2}\dots) \end{aligned}$$

Ordningen blir alltså oändligt stor. Men koefficienterna avtar i regel så i praktiken får vi ett system med ändlig ordning. Parametrarna kommer då likna de trunkerade serierna ovan.

Det finns ett speciellt kommando för minstakvadratmetoden, LS, men denna kräver en strukturfil och är rätt krånglig att använda. Vi kan istället använda det nu redan kända och kraftfulla kommandot ML. Anledningen är att ML alltid börjar med en minstakvadratskattning i första iterationen. Men antalet iterationer kan vi förfara över genom att sätta den globala parametern $itml. = 1$. ML-kommandot kommer då fungera som LS även om den använder en annan numerisk algoritm.

För att vi inte skall glömma bort att återställa $itml. = 20$ (som är standardvärdet) kan vi skriva ett specialmacro för minstakvadratmetoden.

$$\begin{array}{l} macro \quad MK \quad syst \quad < \quad data \quad n \\ let \quad itml. = 1 \\ ml \quad syst < data \quad n \\ let \quad itml. = 20 \\ end \end{array}$$

Använd nu kommandot MK för att skatta processen.

$$\begin{array}{l} mk \quad m1 < wrk \quad 1 \\ mk \quad m2 < wrk \quad 2 \\ o.s.v. \end{array}$$

Vad väljer du för ordningstal nu?

LUNDS TEKNISKA HÖGSKOLA
REGLERTEKNIK
AUTOMATIC CONTROL
LUND INSTITUTE OF TECHNOLOGY

IDPAC-övning 2

Mats Lilja

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Box 118
221 00 Lund

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund SWEDEN

Visitors:
Reglerteknik
Ole Römers väg 1
Lund

Telephone:
+46 46-10 70 00
Telex: S-33533 luniver
Telefax: +46 46-13 81 18

1. IDPAC-Övning 2

I denna övning visas olika metoder för skattning av signal-spektra och skattning av överföringsfunktioner. I början av övningen berörs även skattning av korrelationsfunktioner.

Filen `dat1` innehåller data från tre olika signaler vid 200 ekvidistanta tidpunkter. Plotta de tre signalerna (`plot(200) dat1(1)/dat1(2)/dat1(3)` förslagsvis). Beräkna och plotta de skattade autokorrelationsfunktionerna enligt

```
> acof aco1 < dat1(1) 20
> acof aco2 < dat1(2) 20
> acof aco3 < dat1(3) 20
> switch graph on
> plot (21) aco1 aco2 aco3
```

Antalet 'lags' valdes i detta fall till 20, vilket överensstämmer med tum-regeln att välja detta tal till ca 10-15% av dataseriens längd. Använd nu `aspec`-kommandot för att titta på motsvarande autospektra:

```
> aspec asp1 < dat1(1) 20
> aspec asp2 < dat1(2) 20
> aspec asp3 < dat1(3) 20
> bode asp1 asp2 asp3
```

Fundera över skillnaderna i de tre signalernas spektra (Ledning: signalerna är färgade brus $(1 - aq^{-1})y(t) = e(t)$, $a = 0, 0.7$ resp. 0.95).

Nästa datafil (`dat2`) innehåller $2^8 = 256$ datapunkter. Pröva ånyo kommandona `acof` och `aspec`. Lägg märke till hur antalet lags inverkar på resultatet (pröva t.e.x. för 4, 16 och 64 lags). En alternativ metod att skatta spektrum, är att använda kommandot `dft` (FFT – Fast Forier Transform) direkt på data-mängden:

```
> dft(pow) dft2 < dat2
> bode dft2
```

Optionen "pow" ger effektspektrum. En ytterligare metod består i att göra en minsta-kvadrat-skattning av ett fiktivt system, drivet av vitt brus, där den aktuella signalen är utsignal. Här används makrot `mk`:

```
> macro mk model < data n
> let itml.  = 0
> ml model < data n
> let itml.  = 20
> end
```

I makrot utnyttjas maximum-likelihood kommandot `ML`, vilket använder en enkel minsta-kvadrat-skattning som initial-skattning. Genom att nollställa den globala variabeln `itml.` så görs inga iterationer. Variabeln återställs igen till sitt gamla värde (default 20 iterationer). Skatta spektrumet genom att ta fram minsta-kvadrat-skattningar av ordning 1,2 och 3:

```
> mk mod1 < dat2 1
```

```
> sptrf sp1 < mod1 c / a
> bode(ao) sp1
```

osv. Man kan givetvis ersätta MK med ML. Observera att dat2 bara innehåller en kolonn (utsignalen), så man antar endast en *okänd* insignal (brus) till systemet.

Härnäst skall vi, med olika metoder, skatta överföringsfunktionen hos det system, vars insignal resp. utsignal ligger i filen dat3. Systemet har dessutom en okänd signal (färgat brus) som insignal. Använd först ccof-kommandot för att skatta korskorrelationen mellan insignal och utsignal. Skatta sedan motsvarande korsspektrum (cspec). En skattning av överföringsfunktionen och motsvarande bodediagram erhålles nu genom kommandona (300 datapunkter)

```
> cspec cspec30 < dat3(1) dat3(2) 30
> aspec aspec30 < dat3(1) 30
> frop trf3 < cspec30 / aspec30
> bode trf3
```

Det 'sanna' systemet heter Sys3 och ligger på en allmän area. In- utsignal-sambandet ges av

$$
\begin{aligned}
(1 - 1.8q^{-1} + 1.54q^{-2} - 0.592q^{-3})y(t) = {} & (q^{-1} - 0.9q^{-2} + 0.196q^{-3})u(t) \\
& + 0.2(1 - 0.5q^{-1} + 0.24q^{-2} + 0.37q^{-3})e(t)
\end{aligned}
$$

där $e \in \mathrm{N}(0,1)$ och $u(t)$ är en PRBS-signal. Överföringsfunktionens frekvenskurva kan beräknas och plottas med

```
> sptrf true3 < sys3 b / a
> bode true3
```

Som en jämförelse kan man skatta ML-modeller av olika ordningar och beräkna motsvarande överföringsfunktioner. Gör detta för några olika val av modellordningar.

Till sist skall vi titta på ett exempel på hur man kan kontrollera huruvida det är lönt att använda en linjär modell för givna data. Detta kan göras mha den sk koherensfunktionen ψ_{xy} för två signaler x och y:

$$
\psi_{xy}(\omega) := \frac{|\phi_{xy}(\omega)|}{\sqrt{\phi_x(\omega)\phi_y(\omega)}}
$$

ϕ_{xy} är korsspektrumet mellan x och y osv. Om x och y är in- resp. utsignal för ett linjärt system så gäller att $\psi_{xy}(\omega) = 1$ för alla frekvenser ω. I datafilen dat4 finns två signaler lagrade. Plotta dessa. Flytta över signalerna till varsin fil, ta bort medelvärdet och använd makrot COHER:

```
> trend x < dat4(1) 0
> trend y < dat4(2) 0
> coher cohxy30 < x y 30
```

Antalet lags valdes till 30 eftersom dataseriernas längd var 300. Vore det någon ide' att använda minsta-kvadrat identifiering här (t.ex.)? Jämför med koherensfunktionen för signalerna i dat3. Insignalen i dat3 består av en PRBS-signal med grundperioden 5 s. Hur återspeglas detta i koherensspektrumet?

Rekursiv Identifiering och Adaptiv Reglering

Rolf Johansson

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Oktober 1987

1. INTRODUKTION

Denna laboration avser att belysa rekursiv identifiering av en enkel process, som kan beskrivas som ett dynamiskt system.

I den första uppgiften testas den rekursiva identifieringens egenskaper, när några av dess varianter prövas. Därefter skall ett systemet identifieras, samtidigt som en regulator är inkopplad. Vidare kommer ett experiment med regulatordimensionering baserad på de skattade parametrarna. Slutligen visas adaptiv reglering, där de skattade parametrarna direkt användes i regulatorn.

Rekvisita:

- Tankprocess
- IBM-PC AT
- Adaptor - Program för adaptiv reglering
- Börvärdespotentiometer

Tankprocessen utgöres av två seriekopplade tankar med fria utlopp, där man styr pumpen för inflödet till första tanken (grön kontakt) och mäter nivån i vardera tanken (röda kontakter). Laborationen skall genomföras med *Adaptor* - ett program, som implementerar en självinställande regulator på en IBM-PC AT processdator.

2. BESKRIVNING AV REGULATORPROGRAMMET

En kortfattad beskrivning av de skattnings- och regulatoralgoritmer, som används, ges nedan. Inga egentliga härledningar ges här. Den, som är intresserad av detaljer, hänvisas till att studera litteratur med avsnitt om rekursiv identifiering exempelvis Ljung, Söderström: *Theory and Practice of Recursive Identification*, MIT Press, 1984.

Regulatorprogrammet ADAPTOR innehåller ett antal olika algoritmer för identifiering med minstakvadrat-anpassning av ARMAX-modeller till mätdata. Det finns också rutiner för filtrering av de mätdata, som senare används i MK-anpassningen. Vidare finns det en algoritm för en allmän linjär regulator för tidsdiskret reglering. Detta möjliggör undersökningar av identifiering med slutna reglerslingor. Vidare kan man på detta sätt testa regulatorer, som framtages med enkla beräkningar baserade på de erhållna skattningarna.

Det finns vidare en adaptiv regulator för polplacering av en pol. En annan algoritm har dessutom givits den speciella egenskapen, att integralverkan obligatoriskt ingår. Det finns också indirekt (explicit) adaptiv reglering och en självinställande regulator ad modum Clark-Gawthrop.

Följande beteckningar används nedan:

u : Styrsignal

y : Mätsignal

u_c : Referensvärde

y_m : Referensmodellens utsignal

v : Mätbar störning; Framkopplingssignal

d : Stegformad störning; Lågfrekvent störning

ν : Störning

A Processens polpolynom
B Processens nollställespolynom
b_0 Förstärkning
k Processens tidfördröjning
R Regulatorpolynom m.a.p. u
S Regulatorpolynom m.a.p. y
T Regulatorpolynom m.a.p. u_c
Δ Differensbildning; $1 - q^{-1}$
A_0 Observatörspolynom $1 - tq^{-1}$
A_M Önskat polpolynom $1 - pq^{-1}$
θ Parametervektor
φ Regressionvektor
ε Prediktionsfel

Identifieringsalgoritmen

En kort beskrivning av algoritmen ges av följande uttryck.

Algoritm utan differensbildning

— Processmodell:

$$A(q^{-1})y(t) = q^{-k}B(q^{-1})u(t) + C(q^{-1})v(t) + d$$
$$A(q^{-1}) = 1 + a_1q^{-1} + \ldots + a_{n_A}q^{-n_A}$$
$$B(q^{-1}) = b_0 + b_1q^{-1} + \ldots + b_{nB}q^{-n_B}$$
$$C(q^{-1}) = c_0 + c_1q^{-1} + \ldots + c_{n_C}q^{-n_C}$$

— Regulatormodell:

$$R(q^{-1})u(t) = -S(q^{-1})y(t) + T(q^{-1})u_c(t) - V(q^{-1})v(t)$$
$$R(q^{-1}) = r_0 + r_1q^{-1} + \ldots + r_{n_R}q^{-n_R}$$
$$S(q^{-1}) = s_0 + s_1q^{-1} + \ldots + s_{n_S}q^{-n_S}$$
$$T(q^{-1}) = t_0 + t_1q^{-1} + \ldots + t_{n_T}q^{-n_T}$$
$$V(q^{-1}) = v_0 + v_1q^{-1} + \ldots + v_{n_V}q^{-n_V}$$

— Slutna systemets polpolynom:

$$R(q^{-1})A(q^{-1}) + S(q^{-1})q^{-k}B(q^{-1})$$

— Datafiltrering:

$$y_f(t) = \left(\frac{F(1)}{F(q^{-1})}\right)(y(t) - y_0); \qquad u_f(t) = \left(\frac{F(1)}{F(q^{-1})}\right)(u(t) - u_0);$$
$$F(q^{-1}) = 1 + f_1q^{-1} + f_2q^{-2} + \ldots + f_{n_F}q^{-n_F}$$

— Parameterskattningsmodell:

$$y_f(t) = \Big(1 - A(q^{-1})\Big) y_f(t) + q^{-k} B(q^{-1}) u_f(t) + C(q^{-1}) v_f(t)$$

$$y_f(t) = \theta^T \varphi(t)$$

$$\theta = \begin{pmatrix} -a_1 & -a_2 & \cdots & -a_{n_A} & b_0 & \cdots & b_{n_B} & c_0 & \cdots & c_{n_C} \end{pmatrix}^T$$

$$\varphi(t) = \begin{pmatrix} y_f(t-1) & \cdots & y_f(t-n_A) & u_f(t-k) & \cdots & v_f(t) & \cdots \end{pmatrix}^T$$

— Rekursiv minstakvadrat-metod:

$$\varepsilon(t) = y_f(t) - \widehat{\theta}^T(t-1)\varphi(t)$$

$$P(t) = \frac{1}{\lambda}\left(P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda + \varphi^T(t)P(t-1)\varphi(t)}\right)$$

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + P(t)\varphi(t)\varepsilon(t)$$

Algoritm med differensbildning

— Processmodell:

$$A(q^{-1})\Delta y(t) = q^{-k} B(q^{-1}) \Delta u(t) + C(q^{-1}) \Delta v(t)$$

— Regulatormodell:

$$R(q^{-1})u(t) = -S(q^{-1})y(t) + T(q^{-1})u_c(t) - V(q^{-1})v(t)$$

— Slutna systemets polpolynom:

$$R(q^{-1})A(q^{-1}) + S(q^{-1})q^{-k}B(q^{-1})$$

— Datafiltrering:

$$\Delta y_f(t) = \frac{F(1)\Delta}{F(q^{-1})} y(t); \qquad \Delta u_f(t) = \frac{F(1)\Delta}{F(q^{-1})} u(t); \qquad etc.$$

— Parameterskattningsmodell:

$$\Delta y_f(t) = (1 - A(q^{-1}))\Delta y_f(t) + B(q^{-1})\Delta u_f(t) + C(q^{-1})\Delta v_f(t)$$

$$\Delta y_f(t) = \theta^T \varphi(t)$$

$$\theta = \begin{pmatrix} -a_1 & -a_2 & \cdots & -a_{n_A} & b_0 & b_1 & \cdots & b_{n_B} & c_0 & \cdots & c_{n_C} \end{pmatrix}^T$$

$$\varphi(t) = \begin{pmatrix} \Delta y_f(t-1) & \cdots & \Delta y_f(t-n_A) & \Delta u_f(t-k) & \cdots & \Delta v_f(t) & \cdots \end{pmatrix}^T$$

— Rekursiv minstakvadrat-metod:

$$\varepsilon(t) = \Delta y_f(t) - \widehat{\theta}^T(t-1)\varphi(t)$$

$$P(t) = \frac{1}{\lambda}\left(P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda + \varphi^T(t)P(t-1)\varphi(t)}\right)$$

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + P(t)\varphi(t)\varepsilon(t)$$

Handhavande

I programmet utnyttjas analoga in- och utgångar till IBM-PC/AT enligt följande:

AD 0: Referensvärde u_c

AD 1: Mätvärde y

AD 2: Framkopplingssignal v (mätbar störning)

DA 0: Styrsignal u

Programmet *Adaptor* startas på IBM-PC/AT med kommandot:

```
C:\use\ident>m2 adaptor
```

och svarar strax med en pil "→" Om man då besvarar denna med `INDIRE` och med att därefter att trycka på tangentbordets "return" så erhålles följande tablå med regulatorns status.

```
==========================================================
RECURSIVE LEAST SQUARES ESTIMATION
----------------------------------------------------------
Regulatormode: Stop
Estim = false        Diff = false      FF = false
Tsamp =    1.0000    Ulow =  0.0000    Uhigh = 1.0000

Regulator
 nR = 0      R =     1.0000
 nS = 0      S =     1.0000
 nT = 1      T =     0.0000    0.0000
 nV = -1      V =


Estimator
 nA = 1      A =     1.0000    0.0000
 nB = 0      B =     0.0000
 nC = -1      C =

 nF = 0      F =     1.0000
 k = 1       Lambda =     0.9900

----------------------------------------------------------
```

Programmets parametrar kan nu ändras via tangentbordet med ett antal kommandon, som finns närmare beskrivna i appendix. Följande exempel visar en möjlig följd av kommandon.

Exempel 1:

Betrakta följande kommandosekvens

```
->MANUAL
->nS 1
->S 2.0 1.0
```

```
->k 2
->TSAMP 3
->RUN
->ESTIM T
```

Denna kommandosekvens inleds med begäran om manuell reglering och fortsätter med att gradtalet på polynomet $S(q^{-1})$ sätts lika med ett. Därefter ges koefficienterna i polynomet S sina värden 2.0 och 1.0. I den tredje satsen ändras k till två, varefter samplingsintervallet `TSAMP` sättes till 3 sekunder. Med kommandot `RUN` startas den digitala regulatorn, och med `ESTIM T` körs slutligen parameterskattningen igång.

3. IDENTIFIERING AV ÖVRE TANKENS DYNAMIK

Laborationen skall genomföras för undersökning av den övre tankens dynamik. Vi betraktar styrningen av pumpen som insignal och nivån i övre tanken såsom utsignal.

Med frekvensanalys har man för en av laborationsuppställningarna konstaterat att processen väl kan beskrivas av den tidskontinuerliga överföringsfunktionen

$$G(s) = \frac{b}{s+a} = \frac{0.05}{s+0.0148} = \frac{3.38}{1+s67.7}$$

kring vattennivån 10 cm i övre tanken. Observera att detta gäller för en av uppställningarna och att det kan finnas vissa variationer mellan de olika exemplaren. En tidsdiskret modell med samplingsintervallet h $[s]$ får då formen

$$y(t+h) = a_1 y(t) + b_0 u(t)$$

eller

$$y(t) = \frac{b_0}{q+a_1} u(t) = \frac{b_0 q^{-1}}{1+a_1 q^{-1}} u(t)$$

med

$$a_1 = -\exp(-ah) \qquad b_0 = \frac{b}{a}(1-\exp(-ah))$$

Här nedan följer en lista på den tidsdiskreta överföringsfunktionens koefficienter för några olika val av samplingstiden h.

Parameterskattningar med samplingsperiod h								
h	0.1	0.3	1.0	3.0	10.0	30.0	100.0	300.0
$\widehat{a}_1$	-0.999	-0.996	-0.985	-0.957	-0.863	-0.642	-0.228	-0.012
$\widehat{b}_0$	50E-4	15E-3	0.050	0.147	0.465	1.21	2.61	3.34

Dessa värden kan tjäna som jämförelsematerial till de resultat, som vi kommer att erhålla.

Vid alla försök gäller att man skall genomföra "små" variationer av nivån kring 10 cm. Detta kan att göras med manuell reglering eller med hjälp av olika regulatorer. Olika varianter kommer att studeras.

Vid alla försök bör man också notera vilka parametrar, som används av regulatorn. Om man tillfälligt vill spara undan regulatorer eller parameterskattningar, så kan detta göras med kommandot `SAVE`. De sparade skattningarna med regulator återhämtas med kommandot `LOAD`.

Försök 1

Håll nivån vid 10 cm genom manuell reglering, som erhålles genom kommandot

```
->MANUAL
```

varefter börvärdesspaken justeras kring lämpligt värde. Invänta önskad stationär nivå.

1a. Genomför en rekursiv identifiering utan differensbildning och notera resultaten för några samplingsintervall i tabellen nedan.

```
->nA 1
->nB 0
->k 1
->diff F
->tsamp 10
->estim T
```

Parameterskattningar med samplingsperiod h						
	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$
$\widehat{a}_1$						
$\widehat{b}_0$						

— Slutsatser:

1b. Genomför en rekursiv identifiering utan differensbildning, men där jämviktsnivåerna subtraherats bort. Detta göres genom att invänta stationaritet vid önskad nivå, varefter kommandot `EQUIL` göres. Då inhämtas de stationära signalnivåerna till programmet. Notera resultaten för några samplingsintervall i tabellen nedan.

```
->equil
->diff F
->tsamp 10
->estim T
```

Parameterskattningar med samplingsperiod h						
	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$
$\widehat{a}_1$						
$\widehat{b}_0$						

— Slutsatser:

1c. Genomför samma experiment men med differensbildning (`diff T`) av regressorvariablerna u och y. Brusförstärkningen minskar, om vi dessutom väljer filtret F för regressorerna (jfr sid 4) som ett lågpassfilter.

```
->diff T
->nf 1
->f 1 -0.9
```

Parameterskattningar med samplingsperiod h						
	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$
$\widehat{a}_1$						
$\widehat{b}_0$						

— Slutsatser:

Försök 2

En regulator beskriver ett samband mellan in- och utsignaler. Detta samband gäller utöver de samband som processmodellen anger. Detta kan under vissa omständigheter ställa till problem vid identifiering. Vi skall därför undersöka, om vi får andra parameterskattningar, när vi samtidigt har en regulator inkopplad.

2a. Genomför nu samma experiment som i 1b men med en proportionalregulator, som håller nivån 10 cm. Välj exempelvis följande regulator.

```
->nR 0
->R 1
->nS 0
->S 5
->nT 0
->T 5
->run
->estim T
```

Parameterskattningar med samplingsperiod h						
	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$	$h = \ldots$
$\widehat{a}_1$						
$\widehat{b}_0$						

— Slutsatser:

2b. Försök som i 2a men med variation av referensvärdet.

Parameterskattningar med samplingsperiod h						
	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$
$\widehat{a}_1$						
$\widehat{b}_0$						

— Slutsatser:

2c. Försök som i 2a men med en litet sämre regulator enligt kommandosekvensen.

```
->nR 2
->R 1 -1.7 0.72
->nS 0
->S 1
->nT 0
->T 1
```

Parameterskattningar med samplingsperiod h						
	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$
$\widehat{a}_1$						
$\widehat{b}_0$						

— Slutsatser:

Försök 3

Ett syfte med identifiering kan vara att utnyttja parameterskattningar för att beräkna och att dimensionera regulatorer. Låt processmodellen liksom tidigare vara

$$y(t+h) = -a_1 y(t) + b_0 u(t) + d$$

där d är en okänd störning, som tros variera långsamt. I detta försök skall vi beräkna en tidsdiskret PI-regulator

$$(1 - q^{-1})u(t) = -(s_0 + s_1 q^{-1})y(t) + t_0 u_c(t)$$

så att processen väl följer referensvärdet. Polplacering i origo kan ge den önskade följeverkan, samtidigt som integralverkan kompenserar för d. Eftersom en sådan dimensionering är tämligen känslig för parameterfel, så torde den kunna vara ett bra test på parameterskattningarnas kvalitet. Den önskade regulatorn fås exempelvis för parametervalen

$$s_0 = \frac{1}{b_0}(1 - a_1) \qquad s_1 = \frac{a_1}{b_0} \qquad t_0 = \frac{1}{b_0}$$

varvid det slutna systemet blir

$$y(t + h) = u_c(t) + (1 - q^{-1})d$$

Med nominella parametervärden enligt tabell sid 7 får man då med `TSAMP 10`

$$s_0 = 4.01 \qquad s_1 = -1.86 \qquad t_0 = 2.15$$

och med `TSAMP 3` erhålles i stället

$$s_0 = 13.31 \qquad s_1 = -6.51 \qquad t_0 = 6.80$$

3a. Testa om detta visar sig stämma för några olika av de parameterskattningar, som Du har förtroende för. Anteckna i de nedanstående tabeller de bästa processparametrarna och motsvarande regulatorparametrar.

Parameterskattningar med samplingsperiod h						
	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$
$\widehat{a}_1$						
$\widehat{b}_0$						

Beräknade regulatorparametrar för samplingsperiod h						
	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$	$h = \dots$
s_0						
s_1						
t_0						

— Slutsatser:

4. ADAPTIV REGLERING

Om man infogar en beräkningsrutin, som efter varje rekursion av parameterskattningen beräknar en ny regulatorinställning, så har man konstruerat en *adaptiv* regulator. I programmet kan detta begäras genom följande kommandosekvens, där en referensmodells poler anges jämte en observatör.

```
->tune t
->Am 1 0
->A0 1 -0.3
->run
```

— Slutsatser:

5. DIREKT ADAPTIV REGLERING

En version med s.k. direkt adaptiv reglering förevisas av handledaren. Begreppet *direkt adaptiv regulator* avser en beräkningsgång, där regulatorparametrarna kan identifieras direkt. Därvid slipper man en regulatorberäkning under varje samplingsintervall.

```
->stop
->direct
->nr 1
->r 1 0
->ns 1
->s 0 0
->estim t
->tune t
->run
```

En kortfattad beskrivning av de regulatoralgoritmer som används ges i ett appendix. Inga egentliga härledningar ges här.

Regulatorprogrammet *Adaptor* innehåller ett antal olika algoritmer för adaptiv reglering. En utav dessa utgöres av en adaptiv regulator för polplacering av en pol. En annan algoritm har dessutom givits den speciella egenskapen att integralverkan obligatoriskt ingår. Det också indirekt (explicit) adaptiv reglering och en självinställande regulator *ad modum* Clark-Gawthrop.

Försök

Det brukar rekommenderas i litteraturen att man vid försöken startar i stationärt tillstånd med nivån ungefär lika med börvärdet. Man genomför därefter "små" variationer av nivån kring 10 cm. Detta åstadkommes enklast genom manuell reglering. Detta är dock ej helt nödvändigt, och det kan vara instruktivt att även göra någon "kallstart". Notera genomgående vilka parametrar, som används av regulatorn.

Transienta egenskaper

5a. Välj

```
->TSAMP 2
->AM 1 -0.3
->INTEG T
```

varmed vi begär att erhålla samplingsperioden 3 sekunder, en önskad pol i 0.3 och integralverkan. Studera ett enkelt adapteringsförlopp och notera vilka parametrar som erhålles. Följande kommandosekvens kan användas:

```
->MANUAL
->ESTIM T
->TUNE T
->RUN
```

Spara undan de resulterande parametrarna med hjälp av kommandot `SAVE` för senare bruk. Välj därefter

```
->TSAMP  3
->Am  1 -0.3
->INTEG F
```

och studera ett adapteringsförlopp, då integralverkan ej inkorporeras.

— Slutsatser:

Egenskaper hos integralverkan

5b. Välj samma `TSAMP` och `Am` som i (a). Låt först `INTEG F`. Introducera en laststörning med en hävert eller med hjälp av utloppskranen från den övre tanken. Kontrollera och notera regulatorparametrarna före och efter störningen.

Upprepa experimentet med integralverkan `INTEG TRUE`. Om `SAVE` utförts i a), så kan denna regulator återhämtas med kommandot `LOAD`.

— Slutsatser:

Specifikationsparametrars betydelse

5c. Testa hur `TSAMP` och `Am` skall väljas i förhållande till varandra. Låt `INTEG T`.

Val av `TSAMP` och `Am` anger kraven på lösningstid och bandbredd hos det slutna systemet. Antag att referensmodellens polpolynom är $A_m(q^{-1}) = 1 - a_m q^{-1}$ för ett visst val av samplingstiden `TSAMP`. Polen a_m hos A_m motsvarar då den tidskontinuerliga tidskonstanten $\tau = -\mathtt{TSAMP}/\log(a_m)$. I fallet $a_m = 0$ så blir lösningstiden $k \cdot \mathtt{TSAMP}$.

Genomför experiment med små börvärdesändringar, och se hur snabbt regulatorn uppnår stationärt tillstånd för olika värden på `Am` och `TSAMP`.

Adapteringstider med samplingsperiod h						
TSAMP h	$h = 1.0$	$h = 3.0$	$h = 10.0$	$h = \ldots$	$h = \ldots$	$h = \ldots$
$a_m = 0.0$						
$a_m = 0.3$						
$a_m = 0.7$						

— Slutsatser:

APPENDIX - Adaptiv regulatoralgoritmer

Adaptiv regulator utan integralverkan

- Processmodell:

$$A(q^{-1})y(t) = \underset{\uparrow}{b_0}q^{-k}B(q^{-1})u(t) + d(t)$$

- Regulatormodell:

$$R(q^{-1})\underset{\uparrow}{u(t)} = -S(q^{-1})y(t) + T(q^{-1})u_c(t)$$

- Referensmodell:

$$\underset{\uparrow}{y_m(t)} = q^{-k}\frac{A_M(1)}{A_M(q^{-1})}u_c(t); \qquad A_M(q^{-1}) = 1 - a_{m_1}q^{-1}$$

- Regulatorekvation:

$$\underset{\uparrow}{R(q^{-1})}A(q^{-1}) + \underset{\uparrow}{S(q^{-1})}b_0q^{-k}B(q^{-1}) = b_0A_0(q^{-1})A_M(q^{-1})B(q^{-1})$$

$$\underset{\uparrow}{T(q^{-1})} = A_0(q^{-1})A_M(1); \qquad A_0(q^{-1}) = 1 - a_{0_1}q^{-1}$$

- Parameterskattningsmodell:

$$y(t) = \underset{\uparrow}{R(q^{-1})}\left(\frac{q^{-k}}{A_0(q^{-1})A_M(q^{-1})}u(t)\right) + \underset{\uparrow}{S(q^{-1})}\left(\frac{q^{-k}}{A_0(q^{-1})A_M(q^{-1})}y(t)\right) = \theta^T\varphi(t)$$

- Identifieringsmetod:

$$\underset{\uparrow}{\varepsilon(t)} = y(t) - \widehat{\theta}^T(t-1)\varphi(t)$$

$$\underset{\uparrow}{P(t)} = \frac{1}{\lambda}\left(P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda + \varphi^T(t)P(t-1)\varphi(t)}\right)$$

$$\underset{\uparrow}{\widehat{\theta}(t)} = \widehat{\theta}(t-1) + P(t)\varphi(t)\varepsilon(t)$$

Adaptiv regulator med integralverkan

- Processmodell:

$$A(q^{-1})y(t) = b_0 q^{-k} B(q^{-1})u(t) + d(t)$$

$$A(q^{-1})\Delta y(t) = b_0 q^{-k} B(q^{-1})\Delta u(t) + \nu(t)$$

- Regulatormodell:

$$R(q^{-1})\Delta u(t) = -S(q^{-1})\Delta y(t) + T(q^{-1})\Delta u_c(t) + \alpha(u_c(t) - y(t))$$

$$\alpha = A_0(1)A_M(1)$$

- Referensmodell:

$$y_m(t) = q^{-k}\frac{A_M(1)}{A_M(q^{-1})}u_c(t); \qquad A_M(q^{-1}) = 1 - a_{m_1}q^{-1}$$

- Regulatorekvation:

$$\Big(R(q^{-1})\Delta\Big)A(q^{-1}) + \Big(\quad S(q^{-1})\Delta + \alpha\Big)b_0 q^{-k}B(q^{-1}) = b_0 A_0(q^{-1})A_M(q^{-1})B(q^{-1})$$

$$T(q^{-1})\Delta + \alpha = A_0(q^{-1})A_M(1)$$

- Parameterskattningsmodell:

$$y - \frac{\alpha q^{-k}}{A_0 A_M}y = R\left(\frac{\Delta q^{-k}}{A_0 A_M}u(t)\right) + \quad S\left(\frac{\Delta q^{-k}}{A_0 A_M}y(t)\right) = \theta^T\varphi(t)$$

- Identifieringsmetod:

$$\varepsilon(t) = y(t) - \frac{\alpha}{A_0 A_M}y(t-k) - \widehat{\theta}^T(t-1)\varphi(t)$$

$$P(t) = \frac{1}{\lambda}\left(P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda + \varphi^T(t)P(t-1)\varphi(t)}\right)$$

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + P(t)\varphi(t)\varepsilon(t)$$

APPENDIX

Programmet för den rekursiva identifieringen exekveras på en IBM-PC AT. Den exekverbara filen kallas ADAPTOR.LOD och kan hämtas in via diskett placerad i diskettläsaren. Datorn startas via huvudströmbrytare på höger sida. Bildskärmen har en särskild strömbrytare på sin frontpanel. Programmet startas med kommandot:

C:\ >ADAPTOR

och svarar strax med en pil "→". En tablå med regulatorns status erhålles genom att skriva `INDIRE` och därefter trycka på tangentbordets "return".

```
==========================================================
RECURSIVE LEAST SQUARES ESTIMATION
----------------------------------------------------------
Regulatormode: Stop
Estim = false        Diff = false      FF = false
Tsamp =   1.0000     Ulow =  0.0000    Uhigh = 1.0000

Regulator
 nR = 0      R =     1.0000
 nS = 0      S =     1.0000
 nT = 1      T =     0.0000    0.0000
 nV = -1      V =


Estimator
 nA = 1      A =     1.0000    0.0000
 nB = 0      B =     0.0000
 nC = -1      C =

 nF = 0      F =     1.0000
 k = 1       Lambda =     0.9900

----------------------------------------------------------
```

Programmets parametrar kan nu ändras via tangentbordet med ett antal kommandon, som finns närmare beskrivna nedan. Följande exempel visar en möjlig följd av kommandon.

Exempel 1

Betrakta kommandosekvensen

```
->nS 1
->S 2.0 1.0
->k 2
->tsamp 3
->run
->estim TRUE
```

Denna sekvens inleds med att gradtalet på polynomet $S(q^{-1})$ sätts lika med ett. Därefter ges koefficienterna i polynomet S sina värden 2.0 och 1.0. I den tredje satsen ändras prediktionshorisonten k till två steg varefter samplingsintervallet TSAMP sättes till 3 sekunder. Med kommandot RUN startas den digitala regulatorn och med ESTIM T körs slutligen parameterskattningen igång.

Om man begär

```
->direct
```

så erhålles följande alternativa tablå

```
==========================================================
ADAPTIVE REGULATOR
----------------------------------------------------------
Regulator Mode: STOP
                                          tsamp = 1.000
Estim = FALSE          Integ= TRUE        ulow  = 0.000
Tune = FALSE           FF   = FALSE       uhigh = 1.000

Regulator
 nR = 0     R =    1.0000
 nS = 0     S =    1.0000
 nT = 1     T =    0.0000   0.0000
 nV = -1     V =


Estimator
 nR = 0     R =    1.0000
 nS = 0     S =    1.0000
 nV = -1    V =
 nF = 2     F =    1.0000  -1.0000 0.2100
 nA0= 1     A0=    1.0000  -0.7000
 nAm= 1     Am=    1.0000  -0.3000

 k = 1   Lambda =    0.9900

----------------------------------------------------------
```

Nedan följer en översikt över tillgängliga kommandon

RUN	Start av regulatorn
MANUAL	Manuell reglering med $u := u_c$
STOP	Regulator, parameterskattning stänges av
EXIT	Regulator stänges av; Exekvering avbrytes; Återvänder till DOS
ESTIM 1	Parameterskattning påbörjas/avbrytes
TUNE 1	Regulatortrimning påbörjas/avbrytes
DIFF 1	Differensbildning av data i skattningsalgoritmen till/från
INTEG 1	Integralverkan i adaptiva regulatorn till/från
FF 1	Framkoppling till/från

TSAMP r	Samplingsperiod i sekunder
ULOW r	Undre begränsning för styrsignalen u
UHIGH r	Övre begränsning för styrsignalen u
RO r	Koefficient för styrsignalviktning enligt Clark-Gawthrop
DIRECT l	Direkt adaptiv reglering till/från
INDIRE l	Indirekt adaptiv reglering till/från
CLARKG l	Clark-Gawthrop viktning till/från
NR i	Regulatorpolynomet R:s gradtal
NS i	Regulatorpolynomet S:s gradtal
NT i	Regulatorpolynomet T:s gradtal
NV i	Regulatorpolynomet V:s gradtal
R p	Regulatorpolynomet R sättes till värdet p
S p	Regulatorpolynomet S sättes till värdet p
T p	Regulatorpolynomet T sättes till värdet p
V p	Regulatorpolynomet V sättes till värdet p
NA i	Modellpolynomet A:s gradtal
NB i	Modellpolynomet B:s gradtal
NC i	Modellpolynomet C:s gradtal
NF i	Filterpolynomet F:s gradtal
NA0 i	Observatörspolynomet A0:s gradtal
NAm i	Referensmodellens polpolynom Am:s gradtal
A p	Modellpolynomet A sättes till värdet p
B p	Modellpolynomet B sättes till värdet p
C p	Modellpolynomet C sättes till värdet p
F p	Filterpolynomet F sättes till värdet p
A0 p	Observatörspolynomet A0 sättes till värdet p
Am p	Referensmodellens polpolynom *Am* sättes till värdet p
K i	Prediktionshorisont; Antal samplingsperioder
LAMBDA r	Glömskefaktor i intervallet (0,1)
SHOW	Parameterstatus visas
BACKUP	Reservregulatorns parameterstatus visas
LOAD	Reservregulatorn/estimatorn aktiveras
SAVE	Den aktiva regulatorn/estimatorn kopieras och sparas
SHOW SIG	Signaler u, y, uc, v visas m.a.p tiden
SHOW A	Skattade parametrar i A-polynomet visas
SHOW B	Skattade parametrar i B-polynomet visas
SHOW C	Skattade parametrar i C-polynomet visas
P	P-matrisen redovisas
P0 r	P-matrisens initialvärde sättes
EQUIL	Jämviktsnivåer läses in

EQUIL0	Jämviktsnivåer nollställes
AMP r	Amplitud för intern referensvärdeskälla läses in
PER r	Periodtid [s] för intern fyrkantvåg läses
MEAN r	Medelvärde för intern signalgenerator läses in
DELAY i	Intern fördröjning av styrsignal; Antal samplingsperioder
FILTER l	Lågpassfiltrering/Högpassfiltrering
EXTREF l	Referensvärde tages via AD-omvandlare 0
INTREF l	Internt referensvärde slages till/från

där

r	reellt tal t.ex. 0.99
i	heltal t.ex. 5
p	polynomkoefficienter t.ex. 0.5 1.5
l	logisk variabel dvs. T (true) eller F (false)

Syntesorienterad Identifiering

Ulf Holmberg

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Oktober 1987

Syntesorienterad Identifiering

Inom reglertekniken är vi intresserade av att ha en modell som stämmer bra i ett lägre frekvensområde. Högre ordnings dynamik från brus och olinjäriteter försummas. Kvar har vi således en lägre ordnings modell som vi kan analysera och göra regulatordesign på. Eftersom vi har osäkerhet och begränsad giltighet i vår modell bör vi använda en robust designmetod, dvs där en liten ändring av modellen endast obetydligt påverkar regulatorparametrarna. I laborationen skall vi ta oss ifrån ett identifieringsexperiment, med datainsamling och identifiering, till regulatordesign och reglering.

1. Processen

Processen är konstruerad för att vara svår att reglera, d.v.s. där en PID ej kommer fungera. Den består av en rektangulär platta som är upphängd i ena kanten. Där finns en tyngd på ena sidan vilket gör att plattan hänger lite snett. Vinkeln som plattan har men vertikalplanet mäts och skall regleras. Plattan påverkas aerodynamiskt av en fläkt, placerad på lämligt avstånd ifrån plattan. Processen innehåller således följade otäckheter:

— Fläktmotorns tidskonstant

— Fläktvindens tidsfördröjning

— Resonans (plattan fungerar som en pendel)

— Högfrekvent brus orsakat av turbulens

2. Identifieringsexperiment

Vi skall börja med att samla data; insignal u, och utsignal y; från processen. En signalgenerator skriven i Modula2 skall användas. Man kan här generera en PRBS signal kring nivån `mean` med amplituden `amp` samt logga `(y,u)` i två kolonner på en fil, `Data.mat`. Denna filen kan sedan laddas in i `matlab` där vi skall göra identifieringen. Programmet startas med kommandot

```
m2   signal
```

De möjliga kommandona är nu `{log,quit,h,mean,amp}`.

log	Startar loggning av data `(y,u)` (max 400 st.)
quit	Lämnar programmet
h	Ändrar samplingsintervallet från
h=1.000	1 sekund (tidigare värde)
h=...	till ...

På samma sätt kan man sätta medelvärdet `mean` och amplituden `amp`. Använd samplingsintervallet $h = 0.1s$ samt amplituden $amp = 0.1$. Medelvärdet är lättare att ställa in med en potentiometerstyrd extern bias-signal. Vi skall identifiera systemet kring vinkeln noll. Detta motsvarar ett visst pådrag från fläkten eftersom plattan hänger snett ifrån början.

3. Identifiering

Vi skall nu använda Matlab's identifieringspaket för att ta fram en ARMAX-modell. Börja med att ladda in datafilen:

```
load   data
```

Ta sedan bort bias ur data.

```
z=[data(:,1)-mean(data(:,1)), data(:,2)-mean(data(:,2))];
```

Använd sedan funktionen `armax(z,[na,nb,nc,k])` för att skatta en ARMAX-modell i bakåtskiftoperatorn enligt:

$$(1 + a_1 q^{-1} + \cdots a_{na} q^{-na}) y(t) = (b_k q^{-k} + \cdots b_{nb-1} q^{-(nb-1)-k}) u(t) + \\ + (1 + c_1 q^{-1} + \cdots c_{nc} q^{-nc}) e(t)$$

Lägg märke till att `nb` inte anger B-polynomets ordningstal utan antalet b-parametrar. Parametern `k` anger tidfördröjningen i systemet genom att de k första b-parametrarna fixeras till noll. Kommandot `armax` ger resultatet på en speciell representation, `theta-form`. Skattningarna av parametrarna presenteras tillsammans med sina respektive varianser, samt förlustfunktion och AIC med kommandot `present(·)`. Exempel:

```
th3222=armax(z,[3,2,2,2]);
present(th3222)
```

Jämför t.ex. AIC för skattningar av olika ordningar och tidsfördröjningar för att komma fram till en vettig modell. Ett bra sätt att verifiera modellen är att jämföra dess utsignal med den riktiga utsignalen. Detta kan göras på följade sätt:

```
ym=idsim(z(:,2),th3222);
t=[1:1:length(z(:,1))];
plot(t,z(:,1),t,ym);
```

När vi är nöjda med modellen skall vi plocka ut polynomen A, B, C och överföra dem till framåtskiftoperatorn innan vi gör syntes.

```
[a,b,c,d,f]=polyform(th3222);
```

För exemplet ovan behöver vi ta bort de inledande nollorna i b samt lägga till en nolla i c för att få motsvarande framåtskift-representation.

```
b(1:2)=[];
c=[c 0];
```

(C-polynomet behövs inte i vår designmetod.)

4. Regulatordesign

Designmetoden vi kommer använda är vanlig polplacering, men för att lättare förstå var vi skall placera polerna ur robusthetssynpunkt transformerar vi först över modellen på kontinuerlig tid. När vi sedan genomfört designen i kontinuerlig tid, vilket är en iterativ procedur, översätter vi specifikationerna till diskret tid och utför designen där. Designmetoden ser ut på följande sätt:

1. Överför systemet på kontinuerlig tid där det är lättare att intuitivt bestämma polplacering och avgöra robusthet. Kommandot

```
[Bc,Ac]=desample(Bd,Ad,h);
```

överför den diskreta representationen (Bd, Ad) med samplingsintervall h till kontinuerlig motsvarighet (Bc, Ac).

2. Välj önskat karakteristiskt polynom $A_m(s)$ och observerarpolynom $A_o(s)$. För att regulatorn skall bli kausal måste vi välja

$$\text{grad}A_o \geq 2\text{grad}A - \text{grad}A_m - \text{grad}B^+ - 1$$

där B^+ är den (stabila) delen av B som förkortas bort. Låt oss för enkelhetsskull (och för säkerhetsskull) inte förkorta några nollställen samt välj $\text{grad}A_m = \text{grad}A \equiv n$. Då får vi villkoret

$$\text{grad}A_o \geq n - 1$$

och om vi vill ha integrator med kommer villkoret modifieras till

$$\text{grad}A_o \geq n$$

Placera nollställena för enkelhetsskull jämnt fördelade på ett cirkelsegment i vänstra halvplanet med radien ω_m (och t.ex. $\omega_o = 2 \cdot \omega_m$) där halva öppningsvinkeln hos segmentet är fixerat till 45°. (En variant på Butterworth konfiguration.)

```
Amc=butterworth(n,wm,45);
```

```
Aoc=butterworth(n-1,2*wm,45);
```

Vi har nu endast en parameter, ω_m, att spela med i vår specifikation av det slutna systemet. Om det senare visar sig att vår regulator inte är robust (punkt 4) behöver vi bara modifiera ω_m och göra om designen. (Om ni hinner kan ni även pröva andra nollställeskonfigurationer hos A_m och A_o, t.ex. enkelpoler på reella axeln.)

3. Beräkna R, S, T-polynomen genom att lösa DAB-ekvationen

(Diophantus-Aryabhatta-Bezout)

$$AR_1 + B^- S = A_m A_o$$

där $B = B^+ B^-$ och $B_m = B'_m B^-$. Regulatorn ges sedan av

$$\begin{cases} R = R_1 B^+ \\ S = S \\ T = t_0 A_o B'_m \end{cases}$$

Detta görs med funktionen

[R,S,T]=rstpoly($A, B^+, B^-, A_m, B'_m, A_o, A_r$);

Polynomet A_r tillfogas A i DAB-ekvationen vilket gör att man kan välja integralverkan. I det kontinuerliga fallet blir alltå $A_r(s) = s = [1 \quad 0]$ om man vill ha integrator och $A_r = 1$ utan integrator. Utan integrator blir anropet:

```
[r,s,t]=rstpoly(Ac,1,Bc,Amc,1,Aoc,1);
```

4. *Robusthetstest* : Plotta Nyquistkurvan för öppna systemet $G_o(s) = \frac{B(s)S(s)}{A(s)R(s)}$. Om vi gjort en design som har en Nyquistkurva nära -1 kommer visserligen regulatorn alltid stabilisera vår modell, men kurvan för det sanna öppna systemet kan mycket väl gå på fel sida om -1 och alltså ge ett instabilt slutet system. Följaktligen bör man se till att kurvan inte går för nära den kritiska punkten -1. På så vis får vi en robust design och det gör inte så mycket att vi har lite felaktiga parameterskattningar. Alltså, om Nyquistkurvan är långt ifrån -1, acceptera regulatorn såsom robust och gå vidare till nästa punkt, i annat fall välj ett nytt ω_m och börja om ifrån punkt 2. Plottning av $G_o(i\omega)$ kan göras med kommandona

```
w=logspace(0,2);

[re,im]=Nyquist(conv(Bc,S),conv(Ac,R),w);

plot(re,im)
```

5. Översätt den analoga designen till diskret tid: Vi vet nu vad vi kan kräva av regulatorn för att få en robust design. Denna information finns nu i $A_m(s)$ och $A_o(s)$. Översätt dessa polynom till motsvarande polynom i diskret tid; $A_m(q)$ och $A_o(q)$. Gör därefter om polynomsyntesen i diskret tid för att få en diskret regulator $R(q)$,$S(q)$ och $T(q)$.

```
Amd=real(poly(exp(roots(Amc)*h)));

Aod=real(poly(exp(roots(Aoc)*h)));

[r,s,t]=rstpoly(Ad,1,Bd,Amd,1,Aod,1,'discrete')
```

Observera att om vi hade valt integralverkan så skall $A_r(q) = q - 1 = [1 \quad -1]$.

5. Reglering

Starta programmet `Toolbox` (genom att skriva just "toolbox"). Detta programmet exekverar en regulator av RST-typ. Det finns beskrivet kortfattat i Appendix. Välj först samplingsintervall, `TSAMP`, till 0.1 och horisontella plottiden, `HPT`, till $40s$. Gör därefter `save` så att detta verligen lagras in. Välj därefter referenssignalen till fyrkantvåg med amplitud 0.05 och period $20s$ (och gör `save`). Skriv sedan in dina parametrar i regulatormoden (och gör `save`). Gå till sist in i plot-moden och starta regulatorn.

MATLAB-uppgifter i Processidentifiering

Ulf Holmberg

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Oktober 1987

Matlab-Övning

Syftet med denna övning är att visa hur enkelt man med några få Matlab-kommandon kan lösa (numeriskt) några av övningsuppgifterna i Söderströms bok. För att få ett stort utbyte av övningen krävs dock att man har räknat uppgifterna för hand innan så man kan jämföra sina analytiska asymptotiska resultat med Matlabs numeriska svar. Uppgifterna 8, 9, 10 och 18 behandlas. Notationen $\mathcal{S}, \mathcal{M}, \mathcal{I}, \mathcal{X}$ används för att definiera problemen. Dessa betecknar följande: $\mathcal{S}$: systemet, $\mathcal{M}$: modellstrukturen, $\mathcal{I}$: identifieringsmetoden och $\mathcal{X}$: experimentvillkoret.

1. Uppgift 8.

$\mathcal{S}$	$y(t) + a_0 y(t-1) = e(t)$
$\mathcal{M}_1$	$y(t) + a y(t-1) = e(t)$
$\mathcal{M}_2$	$y(t) + a_1 y(t-1) + a_2 y(t-2) = e(t)$
$\mathcal{I}$	Minstakvadratmetoden

På räkneövningarna räknade vi ut att båda modellstukturerna, $\mathcal{M}_1$ och $\mathcal{M}_2$, gav konsistenta skattningar. Däremot gav den överparametriserade stukturen $\mathcal{M}_2$ större varians. Vi fick att

$$\begin{aligned} \mathtt{Var}(\hat{a}) &= \frac{1}{N}(1 - a_0^2) \\ \mathtt{Var}(\hat{a}_1) &= \frac{1}{N} \end{aligned}$$

Verifiera detta i Matlab. Använd `help` för information om nedanstående kommandon om nödvändigt.

```
rand('normal');
e=rand(1000,1);
a=[1 0.5];
c=1;
th=mktheta(a,[],c);
y=idsim(e,th);
th1=arx(y,1);
present(th1);
Sta=sqrt((1-0.5*0.5)/1000);
th2=arx(y,2);
present(th2);
Sta1=sqrt(1/1000);
```

2. Uppgift 9.

$\mathcal{S}$ $\quad y(t) = e(t) + ce(t-1)$
$\mathcal{M}$ $\quad y(t) + a_1 y(t-1) + \cdots + a_n y(t-n) = e(t)$
$\mathcal{I}$ $\quad$ Minstakvadratmetoden
Beräkna prediktionsfelets varians då $n = 1, 2, 3$ för fallen $c = 0.5$ och $c = 1$. Prediktionsfelets varians blir asymptotiskt (när antalet data är oändligt)

$$\text{då} \quad c = 0.5 \quad \begin{cases} V_1 = 1.05 \\ V_2 = 1.0119 \\ V_3 = 1.0029 \end{cases} \qquad \text{då} \quad c = 1 \quad \begin{cases} V_1 = 1.5 \\ V_2 = 1.33 \\ V_3 = 1.25 \end{cases}$$

För att få dessa resultat krävs ganska långa dataserier. Om du kör på persondator med kraftigt begränsat minnesutrymme får du nöja dig med färre data, vilket ger lite avvikande resultat från det asymptotiska ovan.

```
e=rand(20000,1);
th=mktheta(1,[],[1 0.5]);
y1=idsim(e,th);
th=mktheta(1,[],[1 1]);
y2=idsim(e,th);
arx(y1,1);present(ans);
arx(y1,2);present(ans);
arx(y1,3);present(ans);
arx(y2,1);present(ans);
arx(y2,2);present(ans);
arx(y2,3);present(ans);
```

3. Uppgift 10.

$\mathcal{S}$ $\quad y(t) + a_0 y(t-1) = b_0 u(t-1) + e(t)$ $\qquad S = \frac{b_0}{1+a_0}$
$\mathcal{M}$ $\quad y(t) + ay(t-1) = bu(t-1) + e(t)$ $\qquad \hat{S} = \frac{\hat{b}}{1+\hat{a}}$
$\mathcal{I}$ $\quad$ Minstakvadratmetoden
$\mathcal{X}_1$ $\quad u(t)$ vitt brus $\qquad Eu(t) = 0; \quad Eu^2(t) = 1$
$\mathcal{X}_2$ $\quad u(t) = 1$ steg, $\qquad Eu^2(t) = 1$
Uppgiften är att skatta stationära förstärkningen S enligt ovan. På räkneövningarna har vi visat att $\mathcal{X}_2$ ger den noggrannaste skattningen av $\hat{S}$, (dock ej av $\hat{a}$ och $\hat{b}$). Visa detta genom att skatta $\hat{S}$ för några olika brus-sekvenser $\{e(t)\}$. En skattning görs på följande sätt:

```
e=rand(1000,1);
u1=rand(1000,1);
u2=ones(1000,1);
th=mktheta([1 -0.9],[0 1],1);
y1=idsim([u1,e],th);
y2=idsim([u2,e],th);
th1=arx([y1,u1],[1,1,1]);present(th1);
```

```
th2=arx([y2,u2],[1,1,1]);present(th2);
[a1,b1]=polyform(th1);
s1=sum(b1)/sum(a1);
[a2,b2]=polyform(th2);
s2=sum(b2)/sum(a2);
```

Vi kommer se att, i långa loppet, s2 är en noggrannare skattning av den sanna stationära förstärkningen $S = 10$.

4. Uppgift 18.

Lägg till bias i mätningarna på systemet i uppgift 10.

$$\mathcal{S}\begin{cases}\tilde{y}(t) + a_0\tilde{y}(t-1) = b_0\tilde{u}(t-1) + e(t)\\ y(t) = \tilde{y}(t) + m_y \\ u(t) = \tilde{u}(t) + m_u\end{cases}$$

$\mathcal{M}_1 \quad y(t) + ay(t-1) = bu(t-1) + e(t)$
$\mathcal{M}_2 \quad y(t) + ay(t-1) = bu(t-1) + c + e(t)$
$\mathcal{I} \quad$ Minstakvadratmetoden

Verifiera att sambandet $m_y = \frac{b_0}{1+a_0}m_u$ måste gälla för att $\mathcal{M}_1$ skall kunna ge konsistenta skattningar. Börja med att låta sambandet vara uppfyllt, $m_u = 1$ och $m_y = 10$.

```
u=u1+1;
y=y1+10;
th=arx([y,u],[1,1,1]);
present(th);
```

Ändra nu så att sambandet inte längre är uppfyllt, $m_u = 1$ som innan och $m_y = 5$.

```
y=y1+5;
th=arx([y,u],[1,1,1]);
present(th);
```

Om vi skattar biasen blir skattningarna konsistenta oberoende om sambandet ovan gäller eller ej. Skatta biasen genom att införa ytterligare en insignal som är 1. För multi-insignalsystem måste vi använda den allmänna funktionen pem.

```
u=[u,ones(1000,1)];
th=pem([y,u],[1,[1,1],0,0,[0,0],[1,0]]);
present(th);
```