# PROCESSIDENTIFIERING MED DATAMASKIN

Examensarbete i regleringsteknik

Erik Andersson

INNEHÅLL

Processidentifiering med datamaskin

Rättelse i procedur Laban (sid.18)

Erik Andersson

Ordningsföljden på beräkningarna av $\frac{\partial e(t)}{\partial \theta_i}$ och $\frac{\partial^2 e(t)}{\partial \theta_i \partial \theta_j}$ (EC och ECC) medför att fel uppstår i ECC och därmed i exakta andraderivatmatrisen vid identifiering av 1:a ordningens system .

Rad 30 - 38 (sid 18) skall omordnas till följande:

```
"  30      q:= 2xn+1
           if t=1 then
           begin for i:= 2xm step -1 until 2 do ECC[i]:= ECC[i-1];
                 ECC[1]:=eac-EC[1];
                 ECC[q]:=ebc-EC[n+1];
                 ECC[2xn+q]:=ecc-2xEC[q];
           end;
           for i:=m step -1 until 2 do EC[i]:=EC[i-1];
"  38      EC[1]:=ea+y; EC[n+1]:=eb-u; EC[q]:=ec-e;
"  39      comment slut ber. av första och andra deriv. av e;
```

Hela sid 18 återges korrekt på nästa sida. Denna ändring medför i exempel 2 A,6 exakt o 7 exakt att någon decimal kan vara felaktig. Samma gäller för de två första skattningarna i exempel 3, alltså 4 exakt och 5 exakt respektive 6 exakt, 7 exakt och 8 exakt.(sid.36) I övrigt ingen ändring.

```
procedure LABAN(n,N,t); value n,N,t; integer n,N,t;
begin integer i,j,k,m,K; m:=3×n; K:=N-n+1;
    begin integer p,q,r,s,v;
        u:=y:=e:=V:=0;
        for i:=1 step 1 until 8 do E[i]:=0;
        for i:=1 step 1 until 48 do ECC[i]:=VCC[i]:=0;
        for i:=1 step 1 until 24 do
        begin EC[i]:=V1[i]:=0;
            for j:=1 step 1 until 24 do V2[i,j]:=0
        end;
        comment start loop k;
        for k:=1 step 1 until N do
        begin real ea,eb,ec,eac,ebc,ecc;
            comment procedure Vap för beräkning av app. 2:a deriv.;
            procedure Vap(a,b,c,d,f,g);
            integer a,b,c,d,f,g;
            V2[a,b]:=V2[c,d]+EC[f]×EC[g];
            ea:=eb:=ec:=eac:=ebc:=ecc:=0;
            for i:=1 step 1 until n do
            begin p:=2×n+i;
                ea:=ea-C[p]×EC[i];
                eb:=eb-C[p]×EC[n+i];
                ec:=ec-C[p]×EC[p];
                if t=1 then
                begin eac:=eac-C[p]×ECC[i];
                    ebc:=ebc-C[p]×ECC[p];
                    ecc:=ecc-C[p]×ECC[2×n+p]
                end
            end;
            q:=2×n+1;
            if t=1 then
            begin for i:=2×m step -1 until 2 do ECC[i]:= ECC[i-1];
                ECC[1]:=eac-EC[1];
                ECC[q]:=ebc-EC[n+1];
                ECC[2×n+q]:=ecc-2×EC[q]
            end;
            for i:=m step -1 until 2 do EC[i]:=EC[i-1];
            EC[1]:=ea+y; EC[n+1]:=eb-u; EC[q]:=ec-e;
            comment slut ber. av första och andra deriv. av e;
            e:=-C[q]×E[1]+E[2]+C[1]×y-C[n+1]×u;
            for i:=2 step 1 until n-1 do
            E[i]:=-C[2×n+i]×E[1]+E[i+1]+C[i]×y-C[n+i]×u;
            E[n]:=-C[m]×E[1]+C[n]×y-C[2×n]×u;
            u:=ECS(2×k-1); y:=ECS(2×k);
            e:=e+y; E[1]:=e;
            comment slut ber. av felet;
            V:=V+e×e;
            for i:=1 step 1 until n do
            begin p:=n+i; q:=2×n+i;
                V1[i]:=V1[i]+e×EC[i];
                V1[p]:=V1[p]+e×EC[p];
                V1[q]:=V1[q]+e×EC[q]
            end grad av V;
```

# I

## INLEDNING OCH SYFTE

För utarbetande av kontrollstrategier för en industriell process måste systemekvationen vara känd. I detta examensarbete utvecklas ett datamaskinprogram i ALGOL, som identifierar en allmän linjär process med en ingång och en utgång. Ur en serie mätningar av in- och utdata med lämpligt samplingsintervall beräknar programmet dels systemparametrarna och dels spridning i matrial och parametrar. Systemparametrarna skattas statistiskt med hjälp av maximum-likelihood-metoden (ML-metoden).
En Newton-Raphson-algoritm utnyttjas för lösning av ML-ekvationen.
Grundläggande teorier finns i ref.(1) och (2). Se även ref.(3). En resumé av teorin har gjorts i avd. II.
Beträffande skattningens statistiska egenskaper se ref.(1) sekt.(4).

II

PROBLEMSTÄLLNING OCH LÖSNING

Här göres endast en kortfattad genomgång av teorin i ref.(1) och (2).

Antag systemekvationen är:

$$A(z^{-1})\cdot y(t)=B(z^{-1})\cdot u(t)+C(z^{-1})\cdot e(t) \qquad (1)$$

där u är insignal, y utsignal och e är störningar i processen. e antages oberoende $N(0,\lambda)$.

A,B och C är polynom i skiftoperatorn z, def. så att $z\cdot x(t) = x(t+1)$. (Samplingsintervall = 1).

Det förutsättes 1. A och C har sina nollställen inom enhetscirkeln.

2. A, B och C har inga gemensamma faktorer.

I detta arbete sättes:

$$A(z) = 1+a_1z+a_2z^2+ \ldots +a_nz^n$$
$$B(z) = b_1z+b_2z^2+ \ldots +b_nz^n$$
$$C(z) = 1+c_1z+c_2z^2+ \ldots +c_nz^n$$

Det är ingen inskränkning av det allmänna fallet att sätta $a_o=c_o=1$. Däremot medför $b_o=0$ att mycket snabba system ej kan identifieras korrekt. Detta kan dock kringås. Se III.9.

Huvuduppgift: Skatta koefficienterna i (1).

Vi gör en statistisk skattning av koeff. med hjälp av maximum-likelihood-metoden.

Då $e \in N(0,\lambda)$ är frekvensfunktionen

$$f(e(t)) = \frac{1}{\lambda\sqrt{2\pi}} \exp\left( - \frac{[e(t)-0]^2}{2\lambda^2} \right)$$

och ML-funktionen

$$ML = \prod_{t=1}^{N} f(e(t)).$$

Maximering av ML är ekvivalent med maximering av ln(ML), vilket i sin tur är ekvivalent med minimering av förlustfunktionen, def. som:

$$V = \tfrac{1}{2}\sum_{t=1}^{N} e^2(t) \qquad (2)$$

Bestäm alltså koeff. $a_i$, $b_i$ och $c_i$ för $V_{min}$.

Därefter fås $\lambda$ ur:

$$\lambda^2 = \frac{2}{N} \cdot V_{min} \tag{3}$$

(Medelvärdesriktig skattning får man ur $\lambda^2 = \frac{2}{N-3n} V_{min}$ där n är systemets ordning. 3n kan dock oftast försummas i förhållande till N, antalet mätpunkter.)

För lösning användes följande Newton-Raphson-algoritm:

$$\theta^{l+1} = \theta^l - (V_{\theta\theta}(\theta^l))^{-1} \cdot V_\theta(\theta^l) \tag{4}$$

där:

$\theta = (a_1, a_2, \ldots, a_n, b_1, \ldots, b_n, c_1, \ldots, c_n)$

$V_\theta$ = grad(V)

$V_{\theta\theta}$ = matris med andraderivator av V.

Här söks alltså nollställe till grad(V).

Derivering av (2) ger:

$$\frac{dV}{d\theta_i} = \sum_{t=1}^{N} e(t) \frac{de(t)}{d\theta_i} \tag{5}$$

$$\frac{d^2V}{d\theta_i d\theta_j} = \sum_{t=1}^{N} \frac{de(t)}{d\theta_i} \frac{de(t)}{d\theta_j} + \sum_{t=1}^{N} e(t) \frac{d^2 e(t)}{d\theta_i d\theta_j} \tag{6}$$

Derivering av (1) ger:

$$C(z^{-1}) \frac{de(t)}{da_j} = z^{-j} y(t)$$

$$C(z^{-1}) \frac{de(t)}{db_j} = -z^{-j} u(t) \tag{7}$$

$$C(z^{-1}) \frac{de(t)}{dc_j} = -z^{-j} e(t)$$

Sista formeln i (7) kan deriveras ytterligare:

$$C(z^{-1}) \frac{d^2 e(t)}{da_i dc_j} = z^{-i-j+1} \frac{de(t)}{da_1}$$

$$C(z^{-1}) \frac{d^2 e(t)}{db_i dc_j} = -z^{-i-j+1} \frac{de(t)}{db_1} \tag{8}$$

$$C(z^{-1}) \frac{d^2 e(t)}{dc_i dc_j} = -2z^{-i-j+1} \frac{de(t)}{dc_1}$$

Här har utnyttjats att

$$\frac{de(t)}{da_i} = z^{-i+1} \frac{de(t)}{da_1} = \frac{de(t-i+1)}{da_1} \quad . \tag{9}$$

Analogt för b och c.

Därmed är problemet i princip löst.

Startvärde

Den rekursiva formeln (4) kräver ett startvärde, $\theta^o$.

Bortser man från C-koeff. (antages konstant=0) fås minsta-kvadratskattning av a- och b-koeff., $a^o$ och $b^o$.

$\theta^o = (a^o, b^o, 0)$ tages sedan som startvärde.

Skattningens noggrannhet

Se ref.(2) appendix. Standardavvikelsen för koeff. ges av:

$$\sigma(\theta^{l+1}) = \left\{ \lambda^2 \, V_{\theta\theta}^{-1}(\theta^l) \right\}^{\frac{1}{2}} \quad . \tag{10}$$

# III

## BERÄKNINGSALGORITMEN

### III.1 ALLMÄNT

Vid programmering av lösningen direkt efter formlerna blir behovet av minnesutrymme mycket stort p.g.a. att det erfordras flera fält, som växer med antalet mätpunkter. Se ref.(3) där 9 fält är beroende av N. Då det i praktiken är frågan om långa mätserier, är detta en nackdel. (SMIL har ca 3300 tillgängliga celler i kärnminnet.) Minnesutrymmet kan göras oberoende av antalet mätningar om man inför tillståndsvariabler (III.3) samt använder yttre enheter för mätdata.

Programmet utformas som en procedur där man direkt i anropet anger systemordning n, antal mätpunkter N, index för minsta-kvadrat-skattning z, antal skattningar Z, index för exakt eller approximativ andraderivata t och eventuell reducering av det skattade steget alfa. Dessa index value-deklareras, vilket ger möjlighet att anropa med read. Beräkningar enligt formlerna 2, 5, 6, 7, 8 har förts samman i procedur LABAN. Formel (4) kräver en rutin för matrisinversion. Här används en procedur, som kräver lite extrautrymme. I B-polynomet är $b_0=0$. På grund härav måste vissa åtgärder vidtagas vid identifiering av snabba system. Se III.9 och ex.5.

Programmet skrives i SMIL-ALGOL (för siffermaskinen i Lund), vilket innebär vissa avvikelser från ALGOL-60. Se ref.(7). Obs. speciellt att in- och utdata placeras i och hämtas från yttre kärnminne (ECS) med de maskinkodade procedurerna LECS(read,adress) och ECS(adress). Programmet begränsas till maximal systemordning n=8 av utrymmesskäl.

BETECKNINGAR

iy,i,j,k,l löpande index

m,K,ea,eb,ec,
eac,ebc,ecc hjälpvariabler

u indata
y utdata
e fel
V förlustfunktion
korr korrektion till koeff.
lb hjälpvar. till standardav. för fel och koeff.
alfa reducering av korrektionen.

n systemets ordning
N antal mätpunkter
z startvärde för iterationsindex l.
Z slut- - " -
t index, som anger om exakta 2:a-deriv. eller den approximativa skall användas.

E(1:n) tillståndsvariabel för e

$C(1:3n)=(a_1, a_2, .., a_n, b_1, .., b_n, c_1, .., c_n)$ koeff.

$EC(1:3n)=(e'_{a_1}(t), .., e'_{a_n}(t), e'_{b_1}(t), ....., e'_{c_n}(t))$
derivator av e m.a.p. koeff.

$ECC(1:6n)=(e''_{a_1c_1}(t), .., e''_{a_1c_1}(t-2n+1), e''_{b_1c_1}(t), ...., e''_{c_1c_1}(t-2n+1))$
2:a-derivator av e m.a.p. koeff.

VCC(1:6n)= e(t)·ECC(1:6n) korrektionsterm till app. andraderivatan av V.

$$V2(1:6n, 1:6n) = \begin{Bmatrix} V''_{a_1a_1} & \cdots & V''_{a_1a_n} & V''_{a_1b_1} & \cdots & V''_{a_1b_n} & V''_{a_1c_1} & \cdots & V''_{a_1c_n} \\ & \ddots & & & & & & & \\ & & V''_{a_na_n} & & & V''_{a_nb_n} & & & V''_{a_nc_n} \\ & & & V''_{b_1b_1} & \cdots & V''_{b_1b_n} & V''_{b_1c_1} & \cdots & V''_{b_1c_n} \\ & & & & \ddots & & & & \\ & & & & & V''_{b_nb_n} & & & V''_{b_nc_n} \\ & & & & & & V''_{c_1c_1} & \cdots & V''_{c_1c_n} \\ & & & & & & & \ddots & \\ & & \text{sym.} & & & & & & V''_{c_nc_n} \end{Bmatrix}$$

$V1(1:3n)= (V'_{a_1}, .., V'_{a_n}, V'_{b_1}, .., V'_{b_n}, V'_{c_1}, .., V'_{c_n}) = \text{grad}V$

TILLSTÅNDSVARIABLER

Med hjälp av tillståndsvariabler för felet e(t) undvikes fält, som växer med antalet mätpunkter.

Betrakta ekv.(1).

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

Lös e(t) ur denna ekv. och gruppera kronologiskt.

$$e(t) = y(t) + a_1y(t-1) - b_1u(t-1) - c_1e(t-1) +$$
$$+ a_2y(t-2) - b_2u(t-2) - c_2e(t-2) + \ldots$$
$$\ldots + a_ny(t-n) - b_nu(t-n) - c_ne(t-n)$$

Sätt $E_1(t)=e(t)$ , $E_1(t-1)=e(t-1)$ och

$$E_2(t-1)= a_2y(t-2) - \ldots - c_ne(t-n).$$

Då fås:

$$E_1(t) = -c_1E_1(t-1) + E_2(t-1) + a_1y(t-1) - b_1u(t-1) + y(t)$$
$$E_2(t) = -c_2E_1(t-1) + E_3(t-1) + a_2y(t-1) - b_2u(t-1)$$
$$\vdots$$
$$E_n(t) = -c_nE_1(t-1) + \text{——} + a_ny(t-1) - b_nu(t-1)$$

$E_2 \ldots E_n$ har erhållits på liknande sätt.

Detta kan skrivas i matrisform.

$$\begin{bmatrix} E_1(t) \\ E_2(t) \\ \vdots \\ E_n(t) \end{bmatrix} = \begin{bmatrix} -c_1 & 1 & 0 & \ldots & 0 \\ -c_2 & 0 & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ -c_n & 0 & \ldots & \ldots & 0 \end{bmatrix} \begin{bmatrix} E_1(t-1) \\ E_2(t-1) \\ \vdots \\ E_n(t-1) \end{bmatrix} + \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} y(t-1) - \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} u(t-1) + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} y(t)$$

I algoritmen beräknas lämpligen först den del, som beror av tiden t-1. Därefter inläses in- och utdata för tiden t och tillståndsvariablerna beräknas färdigt. På detta sätt kan man succesivt bygga upp summorna i (2), (5) och (6) varvid man endast behöver lagra n st tillståndsvariabler, 3n st 1:a derivator och 6n st 2:a derivator.

Tillståndsvariablerna kan väljas på andra sätt. Se ref.(2).

III.4

Procedur PRIDE för hela beräkningsalgoritmen

I PRIDE utföres hela processidentifieringen.

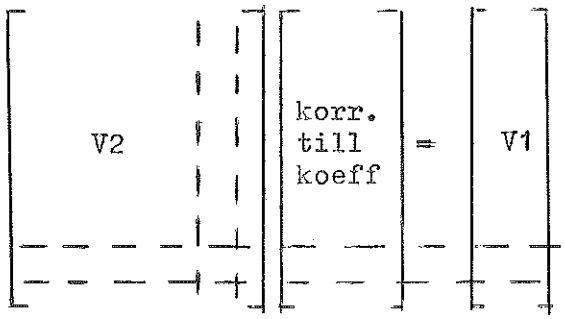
Beteckningen är PRIDE(n,N,z,Z,t,alfa) och ett anrop ger i huvudsak ett antal iterationer med formel (4).

Iterationerna betecknas med löpande l-värden. Vidare:

n = systemets ordning
N = antal mätpunkter
z = index för minsta-kvadrat-skattning
z=1 ger startvärde, z=2 MK-skattning för godtyckliga c-koeff.
Z = antal skattningar
t = index, 0 ger app.- och 1 exakta andraderivator
alfa= reduceringsfaktor för koeff.-korrektion

Startvärde erhålles för z=1 i anropet.

Efter anrop av LABAN nollställes då alla derivator där c-koeff. ingår ( $V'_{c_i}$, $V''_{a_i c_j}$, $V''_{b_i c_j}$, $V''_{c_i c_j}$), ty $c_i$ antages konstant = 0. Se fig.




Därvid förblir c-koeff. noll och skattningen av a- och b-koeff. blir en MK-skattning. Samma sak erhålles då z=2 och c-koeff. ändras ej.

Önskas andra startvärden måste dessa läsas in före anrop av PRIDE, och i anropet måste z=0.

Värdet på t går direkt över i anropet på LABAN.

III.5

Detaljgenomgång av PRIDE

Önskas MK-skattning (z=1 och 2) nollställes behövliga koeff.

Loop 1 börjar och genomlöpes Z gånger.

LABAN anropas. Därvid beräknas dubbla förlustfunktionen (2V), gradienten av V (V1) samt andraderivatmatrisen (V2) exakt eller approximativt beroende på t i anropet. Se III.6,7.

Om z≠0 så nollställes derivator där c-koeff. ingår (se föregående avsnitt). Härvid blir V2 automatiskt singulär. Detta undvikes genom att placera ettor i diagonalen i delmatrisen för cc-derivator.

Förlustfunktionen och standardavvikelsen $\lambda$ (enligt formel 3) för matrialet tryckes.

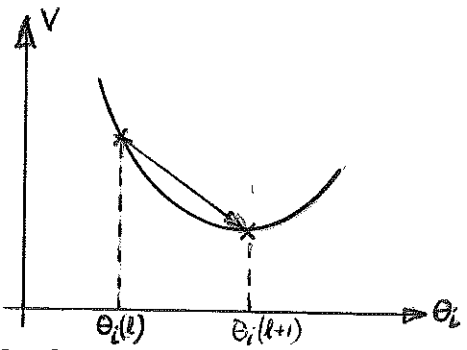
Gradienten av V tryckes.

V2 tryckes.

V2 inverteras. Är därvid V2 singulär utföres hopp till läge UT i huvudprogrammet. Se III.8.

$V2^{-1}$ tryckes. Denna finns på den ursprungliga matrisens plats.

Korrektionerna till koeff. ( $V2^{-1} \cdot V1$) beräknas. Koeff. kan normalt ändras med dessa korrektioner. För varje koeff. har en andragradskurva för V bestämts och algoritmen går till dess extremvärde.




I svåra fall där den verkliga formen på V(θ) avviker avsevärt från andragradsformen ( spec. om minimipunkten ligger nära konvergensranden, se körda ex.) erfordras en reducerings-faktor; här kallad alfa. Denna ingår i proceduranropet

och är alltså åtkommlig utifrån.

Koeff. korrigeras; med eller utan reducering.

Alfa och koeff. tryckes.

Efter varje iteration bestämmes $\sigma_i$ ur formel (10)

och tryckes.

Loop 1 och PRIDE är därmed slut.

Då index i proceduren är value-deklarerade, kan dessa anropas med 'read'. På detta sätt kan man under beräkningarnas gång välja lämpligt alfa och exakt eller app. andraderivata.

III.6

Procedur LABAN för grundläggande beräkningar

Beräkningarna av förlustfunktionen (V), grad V (V1) och 2:a-derivatorna av V (V2) har sammanförts i proceduren LABAN(n,N,t).

Vid de första iterationerna kan med fördel en approximativ andraderivata användas.

$$V2(i,j) = \frac{d^2V}{d\theta_i d\theta_j} \text{ app} = \sum_{t=1}^{N} \frac{de(t)}{d\theta_i} \frac{de(t)}{d\theta_j}$$

Tilläggselementet: $+ \sum_{t=1}^{N} e(t) \frac{d^2e(t)}{d\theta_i d\theta_j}$

beräknas endast då index t i proceduranropet är lika med ett. Detta adderas i efterhand till V2. Observera att tilläggselementet är skilt från noll endast då derivata m.a.p. c-koeff. ingår. Se formel (7) och (8). P.g.a. approcimativa andraderivatornas inbördes samband kan räknearbetet skäras ned betydligt.Se nästa avsnitt. Av formel (9) framgår att endast derivator av e m.a.p. $a_1$ ,$b_1$ och $c_1$ behöver beräknas. För formlerna (5) och (6) måste man då spara 3n förstaderivator och 6n andraderivator.

Beräkningsgången blir: Betrakta tiden t. Beräkna först de storheter, som beror av storheter vid tiden t-1. Läs sedan in in- och utsignaler för tiden t och beräkna övriga storheter. Gå till tiden t+1.

III.7

Detaljgenomgång av LABAN

K och m är hjälpvariabler; K användes vid uppbyggnad av V2. Först nollställes behövliga element. Därefter startar loop k, som genomlöpes för varje datapar i mätserien. ea, eb, ec, eac, ebc, och ecc är hjälpvariabler vid beräkning av $\frac{de}{d\theta_i}$ och $\frac{d^2e}{d\theta_i d\theta_j}$. Proceduren Vap multiplicerar derivator av e och summerar dem till V2.

Hjälpvariablerna nollställes. ea, eb och ec summeras upp enligt formel (7), vänstra ledet. eac, ebc och ecc summeras upp enligt formel (8), vänstra ledet. Vektorerna EC och ECC för $e'$ och $e''$ förskjuts ett samplingsintervall. Jfr. formel (9). Därefter beräknas $e'_{a_1}$, $e'_{b_1}$, $e'_{c_1}$, $e''_{a_1c_1}$, $e''_{b_1c_1}$ och $e''_{c_1c_1}$ färdigt enligt formlerna (7) och (8).

Av tillståndsvariablerna för felet (se III.3) beräknas först den del, som beror av tiden t-1. $E_1(t-1)$ sparas, då denna storhet ingår i övriga variabler. In- och utdata för tiden t läses in, och $E_1(t) = e(t)$ beräknas färdigt. Inläsning av data göres i detta program med hjälp av SMIL-s maskinprocedur ECS(adress i yttre kärnminne). Dubbla förlustfunktionen utökas med kvadraten på felet enligt formel (2). Gradienten av förlustfunktionen (V1) beräknas enligt formel (5).

Approximativa andraderivatorna beräknas med hjälp av proceduren Vap. Här är att märka:

$$\frac{d^2V}{da_i db_j} = \sum_{t=1}^{N} \frac{de(t)}{da_i} \frac{de(t)}{db_j} = \sum_{t=1}^{N} \frac{de(t-i+1)}{da_1} \frac{de(t-j+1)}{db_1}$$

om $j \leq i$ sätt $t'=t-j+1$ så $\frac{d^2V}{da_i db_j} = \sum_{t'=1}^{N-j+1} \frac{de(t'+j-i)}{da_1} \frac{de(t')}{db_1}$

om $i<j$ sätt $t'=t-i+1$ så $\frac{d^2V}{da_i db_j} = \sum_{t'=1}^{N-i+1} \frac{de(t')}{da_1} \frac{de(t'+i-j)}{db_1}$

Analogt fås:

$$\frac{d^2V}{da_{i-1} db_{j-1}} = \sum_{t'=1}^{N-j+2} \frac{de(t'+j-i)}{da_1} \frac{de(t')}{db_1} \qquad j \leq i$$

$$\frac{d^2V}{da_{i-1} db_{j-1}} = \sum_{t'=1}^{N-i+2} \frac{de(t')}{da_1} \frac{de(t'+i-j)}{db_1} \qquad i < j$$

Härur följer att element $\frac{d^2V}{da_{i-1} db_{j-i}}$ fås ur element $\frac{d^2V}{da_i db_j}$ med ytterligare en summering. Skilda formler måste betraktas beroende på om i eller j är störst; ty annars fås derivator av e , som ej är kända vid tiden t. Analoga formler fås för övriga delmatriser i V2. Det räcker sålunda att beräkna vissa delvektorer i V2 så länge $k \leq N-n+1 = K$. Slutelementet i delvektorerna får ej summeras dubbelt.

Då $k > K$ byggs hela V2 upp succesivt enligt formlerna ovan. Se under else. Här måste dubbelsummering i diagonalen undvikas.

Korrektionen till andraderivatorna beräknas och placeras i seperat vektor (VCC), då t=1 i proceduranropet. Korrektionen är skild från noll endast då derivata m.a.p. c-koeff ingår (se tidigare).

Betrakta element i delmatris för a- och c-koeff.

$$\frac{d^2V}{da_i dc_j}_{korr} = \sum_{t=1}^{N} e(t) \frac{d^2e(t)}{da_i dc_j} = \sum_{t=1}^{N} e(t) \frac{d^2e(t-i-j+2)}{da_1 dc_1}$$

Element med summan av index lika får alltså samma korrektion. Analogt i övriga delmatriser. Korrektionen kan därför hållas i vektorform i stället för matrisform. Av formeln framgår också att man måste spara 2n andraderivator av e för var och en av tre delmatriser.
Jfr. V2app. där skillnaden av index ingår.
Efter eventuell beräkning av korrektionerna är loop k slut. Slutligen ( om t=1 i anrop ) adderas korrektionerna till V2, som därvid innehåller de exakta andraderivatorna.

Därvid gäller:

VCC(1) ⟶ $V''_{a_i c_j}$ (i+j=2)
VCC(2) ⟶ $V''_{a_i c_j}$ (i+j=3)
.....
VCC(2n-1) ⟶ $V''_{a_i c_j}$ (i+j=2n)
VCC(2n) anv. ej
VCC(2n+1) ⟶ $V''_{b_i c_j}$ (i+j=2)
o.s.v.

V2 speglas så den blir kvadratisk.

Slut LABAN.

III.8

Procedur invers för matrisinversion.

Formel (4) kräver en rutin för matrisinversion till V2. I arbetets tidiga skede har använts en procedur, som hämtats från ref.(3), Det Gauss. Denna procedur kräver dock för stort minnesutrymme då systemordningen är hög. För systemprdning n=10 (matrisordning=30) kräves ca 2·3n·3n = 1800 celler. (SMIL har ca 3300 celler tillgängliga i kärnminnet.) Denna procedur har därför bytts mot proceduren invers, som är en kombination av två procedurer (matrixinvert och matrixperm) hämtade ur ACM-64, band 7. En sammanslagning var nödvändig då de allmänna anropen mellan procedurerna ej är tillåtna i SMIL-ALGOL. Se ref.( 7 ). Invers inverterar matrisen direkt i matrisfältet och antalet minnesceller blir ca 3n·3n = 900 för n=10; alltså ungefär samma som matrisen själv.

Vid anropet invers(a,n,eps,singular) inverteras matrisen a av ordning n med Gauss-Jordans metod och resultatet finns i fältet a. Den ursprungliga matrisen blir sålunda förstörd. Vid varje steg användes det absolut största elementet som pivotelement. Index för succesiva pivotelement placeras i vektorerna r och c, vilka sedan användes vid återpermutering. Om något pivotelement är mindre än eps, går proceduren till läge singular i huvudprogrammet.

Vid körning på SMIL måste fältgränser fixeras. Se ref.( 7 ). Här har valts maximal matrisordning 30·30, som motsvarar systemordning 10. Vid högre ordning måste dessa fältgränser ändras.

III.9

Identifiering av snabba system

Då $b_o$ i B-polynomet satts lika med noll, har insignalen vid en viss tidpunkt ingen inverkan på utsignalen vid samma tidpunkt. Mycket snabba system kan då ej identifieras korrekt. En förbättrad lösning kan eventuellt erhållas genom att minska samplingsintervallet.

Hur skall man få en skattning av $b_o$ med detta program?

Betrakta formel (1):

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

Denna omformas till:

$$A(z^{-1})y(t) = [z \cdot B(z^{-1})][z^{-1}u(t)] + C(z^{-1})e(t)$$

Förskjut alltså indata ett samplingsintervall bakåt i tiden (första insignalen och sista utsignalen får då kasseras). Beräkna som vanligt. Det skattade polynomet $B' = z \cdot B(z^{-1})$ blir då:

$$B'(z^{-1}) = b_1 + b_2 z^{-1} + b_3 z^{-2} + \ldots + b_n z^{-(n-1)}$$

$b_1$ är i detta fall en skattning av $b'_o$

$b_2$ - " - $b'_1$

.....

$b_n$ - " - $b'_{n-1}$

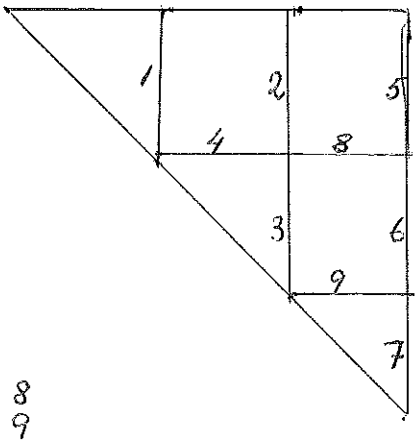
Ingen skattning av $b'_n$ erhålles. Vid så snabba system, som det här är frågan om , torde dock denna koeff. vara signifikant noll.

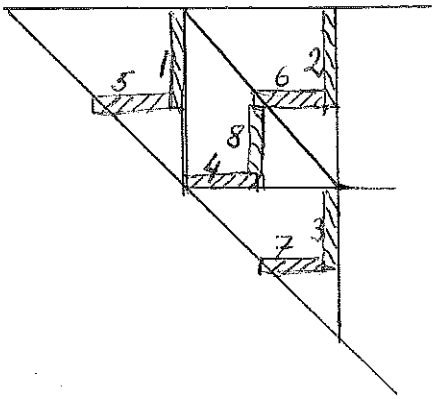
ALGOL-programmet

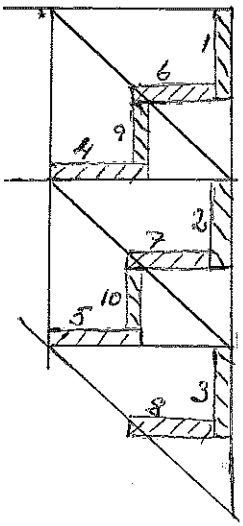
```
begin integer iy,x; real e,V,u,v,korr,lb;
    array E[1:8],C,EC,V1[1:24],VCC,ECC[1:48],V2[1:24,1:24];
procedure PRIDE(n,N,z,Z,t,alfa); value n,N,z,Z,t,alfa;
integer n,N,z,Z,t; real alfa;
begin integer i,j,k,l,m; m:=3×n;
    if z=1 then for i:=1 step 1 until m do C[i]:=0;
    if z=2 then for i:=1 step 1 until 2×n do C[i]:=0;
    comment start loop l;
    for l:=1 step 1 until Z do
    begin IAPAN(n,N,t);
        if z=1∨z=2 then
        begin for i:=2×n+1 step 1 until m do
            begin V1[i]:=0;
                for j:=1 step 1 until m do
                V2[j,i]:=V2[i,j]:=0;
            end;
            for i:=2×n+1 step 1 until m do V2[i,i]:=1
        end;
        punch(1); print(6,5,V/2); punch(1);
        lb:=V/N; print(sqrt(lb)); punch(1);
        for i:=1 step 1 until m do print(5,3,V1[i]); punch(1);
        for i:=1 step 1 until m do
        begin punch(1);
            for j:=1 step 1 until m do print(6,2,V2[i,j])
        end;
        invers(V2,m,10-8,UT);
        for i:=1 step 1 until m do
        begin punch(1);
            for j :=1 step 1 until m do print(2,6,V2[i,j])
        end; punch(1);
        for i:=1 step 1 until m do
        begin korr:=0;
            for j:=1 step 1 until m do
            korr:=korr+V2[i,j]×V1[j];
            C[i]:=C[i]-korr×alfa;
        end korrektion av koeff;
        print(alfa); if t=1 then punch(5); punch(1);
        for i:=1 step 1 until m do print(3,5,C[i]); punch(1);
        for i:=1 step 1 until m do print(4,4,sqrt(abs(lb×V2[i,i])));
        punch(1); punch(1);
    end loop l;
    punch(1); punch(1);
end PRIDE;;
```

```
procedure LABAN(n,N,t); value n,N,t; integer n,N,t;
begin integer i,j,k,m,K; m:=3×n; K:=N-n+1;
    begin integer p,q,r,s,v;
        u:=y:=e:=V:=0;
        for i:=1 step 1 until 8 do E[i]:=0;
        for i:=1 step 1 until 48 do ECC[i]:=VCC[i]:=0;
        for i:=1 step 1 until 24 do
        begin EC[i]:=V1[i]:=0;
            for j:=1 step 1 until 24 do V2[i,j]:=0
        end;
        comment start loop k;
        for k:=1 step 1 until N do
        begin real ea,eb,ec,eac,ebc,ecc;
            comment procedure Vap för beräkning av app. 2:a deriv.;
            procedure Vap(a,b,c,d,f,g);
            integer a,b,c,d,f,g;
            V2[a,b]:=V2[c,d]+EC[f]×EC[g];
            ea:=eb:=ec:=eac:=ebc:=ecc:=0;
            for i:=1 step 1 until n do
            begin p:=2×n+i;
                ea:=ea-C[p]×EC[i];
                eb:=eb-C[p]×EC[n+i];
                ec:=ec-C[p]×EC[p];
                if t=1 then
                begin eac:=eac-C[p]×ECC[i];
                    ebc:=ebc-C[p]×ECC[p];
                    ecc:=ecc-C[p]×ECC[2×n+p]
                end
            end;
            for i:=m step -1 until 2 do EC[i]:=EC[i-1];
            for i:=2×m step -1 until 2 do ECC[i]:= ECC[i-1];
            q:=2×n+1;
            EC[1]:=ea+v; EC[n+1]:=eb-u; EC[q]:=ec-e;
            if t=1 then
            begin ECC[1]:=eac-EC[2];
                ECC[q]:=ebc-EC[n+2];
                ECC[2×n+q]:=ecc-2×EC[q+1]
            end första och andra deriv. av e;
            e:=-C[q]×E[1]+E[2]+C[1]×v-C[n+1]×u;
            for i:=2 step 1 until n-1 do
            E[i]:=-C[2×n+i]×E[1]+E[i+1]+C[i]×v-C[n+i]×u;
            E[n]:=-C[m]×E[1]+C[n]×v-C[2×n]×u;
            u:=ECS(2×k-1); v:=ECS(2×k);
            e:=e+v; E[1]:=e;
            comment slut ber. av felet;
            V:=V+e×e;
            for i:=1 step 1 until n do
            begin p:=n+i; q:=2×n+i;
                V1[i]:=V1[i]+e×EC[i];
                V1[p]:=V1[p]+e×EC[p];
                V1[q]:=V1[q]+e×EC[q]
            end grad av V;
```

```
if k<K then
begin comment bygg upp delvektorer i app. V2;
    for i:=1 step 1 until n do
    begin p:=n+i; q:=2×n; r:=n+1-i;
        Vap(i,n,i,n,1,r);                          1.
        Vap(p,q,p,q,n+1,n+r);                      2.
        Vap(i,q,i,q,1,n+r);                        3.
        if i<n then Vap(n,p,n,p,n+1,r);            4
    end;
    for i:=1 step 1 until n do
    begin p:=2×n+1; q:=2×n+i; r:=m+1-i;
        Vap(i,m,i,m,1,r);                          5.
        Vap(n+i,m,n+i,m,n+1,r);                    6
        Vap(q,m,q,m,p,r);                          7
        if i<n then begin Vap(n,q,n,q,p,n+1-i);    8
                          Vap(p-1,q,p-1,q,p,p-i) end  9
    end;
end
else
begin comment bygg upp hela app  V2;
    s:=N-k+1; for i:=1 step 1 until s do
    begin p:=n+i; q:=2×n; r:=n+1-i;
        Vap(i,n,i,n,1,r);                          1
        Vap(p,q,p,q,n+1,n+r);                      2
        Vap(i,q,i,q,1,n+r);                        3
        Vap(n,p,n,p,n+1,r)                         4
    end;
    for i:=s step 1 until n-1 do
    begin p:=n+s; q:=n+i; r:=i+1-s;
        Vap(s,i,s+1,i+1,1,r);                      5
        Vap(p,q,p+1,q+1,n+1,n+r);                  6
        Vap(s,q,s+1,q+1,1,n+r);                    7
        if i>s then Vap(i,p,i+1,p+1,n+1,r)         8
    end;
    for i:=1 step 1 until s do
    begin p:=2×n; q:=2×n+i; r:=m+1-i;
        Vap(i,m,i,m,1,r);                          1
        Vap(n+i,m,n+i,m,n+1,r);                    2
        Vap(q,m,q,m,p+1,r);                        3
        Vap(n,q,n,q,p+1,n+1-i);                    4
        Vap(p,q,p,q,p+1,r-n)                       5
    end;
    for i:=s step 1 until n-1 do
    begin p:=n+s; q:=2×n+i; r:=q+1-s; v:=2×n+1;
        Vap(s,q,s+1,q+1,1,r);                      6
        Vap(p,q,p+1,q+1,n+1,r);                    7
        Vap(n+p,q,v+s,q+1,v,r);                    8
        if i>s then begin Vap(i,n+p,i+1,v+s,v,1+i-s);   9
                          Vap(n+i,n+p,n+i+1,v+s,v,r-n) end 10
    end;
end else och uppbyggnad av app. V2;
if t=1 then for i:=1 step 1 until 2×m do
VCC[i]:=VCC[i]+e×ECC[i]
end loop k;
```



```
        comment komplettera V2 med korrektion;
        if t=1 then
        for j:=1 step 1 until n do
        begin p:=2×n; q:=p+j;
            for i:=1 step 1 until n do
            begin r:=i+j-1;
                V2[i,q]:=V2[i,q]+VCC[r];
                V2[n+i,q]:=V2[n+i,q]+VCC[p+r]
            end;
            for i:=1 step 1 until j do
            begin r:=i+j-1;
                V2[p+i,q]:=V2[p+i,q]+VCC[p+p+r]
            end
        end V2 kompletterad med korr;
        for i:=1 step 1 until m do
        for j:=i+1 step 1 until m do V2[j,i]:=V2[i,j];
    end
end LABAN;;
```

```
procedure invers(a,n,eps,singular); value n,eps; array a;
integer n; real eps; label singular;
comment Inverterar matrisen a av ordning n med Gauss-Jordans metod.
Matrisen a blir förstörd då inversen bildas utan extra fält. Vid
varje steg användes det abs. största elementet som pivoelement.
Index för succesiva pivoelement placeras i vektorerna r och c,
vilka sedan användes för återpermutering. Om något pivoelement
är mindre än eps går proceduren till läge singular i huvudprogrammet;
begin integer i,j,k,l,pivi,pivj,p; real pivot; integer array r,c[1:30];
    for i:=1 step 1 until n do r[i]:=c[i]:=i;
    comment sök startvärde för pivot; pivi:=pivj:=1;
    for i:=1 step 1 until n do for j:=1 step 1 until n do
    if abs(a[i,j])>abs(a[pivi,pivj]) then
    begin pivi:=i; pivj:=j end;
    comment start lösning;
    for i:=1 step 1 until n do
    begin l:=r[i]; r[i]:=r[pivi]; r[pivi]:=l;
        l:=c[i];c[i]:=c[pivj]; c[pivj]:=l;
        if eps>abs(a[r[i],c[i]]) then go to singular;
        for j:=n step -1 until i+1,i-1 step -1 until 1 do
        a[r[i],c[j]]:=a[r[i],c[j]]/a[r[i],c[i]];
        a[r[i],c[i]]:=1/a[r[i],c[i]];
        pivot:=0;
        for k:=1 step 1 until i-1,i+1 step 1 until n do
        begin for j:=n step -1 until i+1,i-1 step -1 until 1 do
            begin  a[r[k],c[j]]:=a[r[k],c[j]]-a[r[i],c[j]]×a[r[k],c[i]];
                if k>i^j>i^abs(a[r[k],c[j]])>abs(pivot) then
                begin pivi:=k; pivj:=j; pivot:=a[r[k],c[j]] end test
            end j loop;
            a[r[k],c[i]]:=-a[r[i],c[i]]×a[r[k],c[i]];
        end k loop;
    end i loop och lösning;
    comment start återpermutation av rader för z=1 och av kolonner för z=2;
    begin  integer array tag, loc[1:30]; integer z,i,t; real w;
        for z:=1,2 do
        begin for i:=1 step 1 until n do tag[i]:=loc[i]:=i;
            for i:=1 step 1 until n do
            begin if z=1 then
                begin t:=r[i]; j:=loc[t]; k:=c[i] end
                else begin t:=c[i]; j:=loc[t]; k:=r[i] end;
                if j≠k then
                begin for p:=1 step 1 until n do
                    begin if z=1 then
                                begin w:=a[j,p]; a[j,p]:=a[k,p]; a[k,p]:=w end
                            else begin w:=a[p,j]; a[p,j]:=a[p,k]; a[p,k]:=w end
                    end p loop;
                    tag[j]:=tag[k]; tag[k]:=t;
                    loc[t]:=loc[tag[j]]; loc[tag[j]]:=j
                end j,k test
            end i loop
        end z loop
    end permutation
end invers;;
```

```
    x:=read;
    for iv:=1 step 1 until x do IECS(read,iv);
NYTT: PRIDE(read,read,read,read,read,read);
    go to NYTT;
UT: punch(3);
end
```

Kommentar till programmet

Skiss:
```
begin  deklarationer
       procedur PRIDE(n,N,z,Z,t,alfa)
       procedur LABAN(n,N,t)
       procedur invers(a,m,eps,singular)
       huvudprogram
end
```

Huvudprogrammet (sid 22) är av allmän typ. Det kan varieras från körning till körning. Dock måste följande beaktas:

1. Mätdata skall läsas in till ECS (yttre kärnminne) på följande sätt

| cell i ECS | 1 | 2 | 3 | 4 | ..... | 2N-1 | 2N |
|---|---|---|---|---|---|---|---|
| variabel | $u_1$ | $y_1$ | $u_2$ | $y_2$ | ..... | $u_N$ | $y_N$ |

   Detta göres med LECS(variabel,cellnr.).

2. Eventuell inläsning av speciella startvärden kan göras före anrop av PRIDE. Därvid måste z anropas med noll.
3. Läge UT måste finnas före programmets sista end. Dit sker hopp om V2 singulär.
4. Programmets sista end tillfogas sist ( programmets första begin finns på remsa 1 ).

Typexempel

A. Normal skattning

Använd huvudprogrammet sid 22.
Läs in data till ECS.
Läs index: (n,N, 1 , 1 , 0 , 1 )
som ger: MK-skatt.,1 loop, app. deriv.
Läs index: (n,N, 0 , 1, 0 ,alfa)
Det sista upprepas ( med lämpliga alfa-värden) tills skattningen konvergerar.
Läs index: (n,N, 0 , Z , 1 , 1 )
som ger: Z st skattningar med exakta derivator.
Normalt behövs endast 2 - 4 skattningar med exakta andraderivator.

B. MK-skattning för olika c-koeff. (jfr. ex. 3)

```
x:=read;        (2N)
for iy:=1 step 1 until x do LECS(read,iy);
x:=read;        (n)
koeff: for iy:=2×x step 1 until 3×x do C[iy]:=read;
PRIDE(read,read,2,1,0,1);
PRIDE(read,read,0,1,0,1);
go to koeff;
UT: punch(8)
end
```

Första anropet ger MK-skattningen och andra förlustfunktionen i denna punkt.

C. Speciella startvärden.

```
x:=read;
for iy:=1 step 1 until x do LECS(read,iy);
x:=read;     (3n)
koeff: for iy:=1 step 1 until x do
       begin C[iy]:=read; print(6,4,C[iy]) end;
PRIDE(read,read,read,read,read,read);
go to koeff;
UT:
end
```

Denna variant har använts i exempel 6 och 7, där man lätt erhållit instabil skattning av c-koeff.

D. Enkel regression

Här måste derivator i PRIDE nollställas. I appendix finns de varianter av proceduren och huvudprogrammet, som använts i exempel 5. Detta är ett typexempel på hur andra specialfall av den allmänna skattningen skall konstrueras.

IV

TESTKÖRNINGAR

Procedur LABAN har testats på 10 st datapar hämtade ur den genererade serien i ex. 2.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| u | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | -1 | 1 |
| y | 2.4 | 1.0 | 4.1 | 3.9 | 1.0 | -2.6 | 4.1 | -1.9 | -1.3 | -2.5 |

Startvärde: $\theta^1 = (-1.4,\ 0.8,\ 0.8,\ 0.4,\ -1.0,\ 0.1)$

Handräkning har gjorts direkt med formlerna (ej över tillståndsvariabler). Överensstämmande resultat har därvid erhållits av de kontrollerade variablerna e, e´, e´´, V, V´, $V''_{app}$, $V''_{korr}$.

Procedur invers har testats på följande matriser

1/ $$A = \begin{bmatrix} 2 & 4 \\ 3 & 1 \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} -0.1 & 0.4 \\ 0.3 & -0.2 \end{bmatrix}$$

2/ $$A = \begin{bmatrix} 1 & 2 & 5 \\ 2 & 4 & 9 \\ 3 & 7 & 11 \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} -19 & 13 & -2 \\ 5 & -4 & 1 \\ 2 & -1 & 0 \end{bmatrix}$$

3/ $$A = \begin{bmatrix} 1 & 1/2 & 1/3 \\ 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{bmatrix}$$

4/ $$A = \begin{bmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} 68 & -41 & -17 & 10 \\ -41 & 25 & 10 & -6 \\ -17 & 10 & 5 & -3 \\ 10 & -6 & -3 & 2 \end{bmatrix}$$

5/ $$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 3 & 1 & 2 \end{bmatrix}$$ $A^{-1}$ Existerar ej.

Exempel 3 är en Hilbert-matris; $a_{ij} = (i+j-1)^{-1}$.

Ex. 3 och 4 är hämtade ur Basic Theorems in Matrix Theory ( National Bureau of Standards ) sid 22.

Resultat av inverteringstest.

1/ $A^{-1} =$

| | |
|---|---|
| -0.1000000 | 0.4000000 |
| 0.3000000 | -0.2000000 |

2/ $A^{-1} =$

| | | |
|---|---|---|
| -19.0000003 | 13.0000002 | -2.0000000 |
| 5.0000001 | -4.0000001 | 1.0000000 |
| 2.0000000 | -1.0000000 | 0.0000000 |

3/ $A^{-1} =$

| | | |
|---|---|---|
| 9.0000013 | -36.0000064 | 30.0000060 |
| -36.0000064 | 192.0000329 | -180.0000305 |
| 30.0000060 | -180.0000305 | 180.0000282 |

4/ $A^{-1} =$

| | | | |
|---|---|---|---|
| 68.0000140 | -41.0000084 | -17.0000037 | 10.0000022 |
| -41.0000085 | 25.0000051 | 10.0000022 | -6.0000013 |
| -17.0000037 | 10.0000021 | 5.0000009 | -3.0000005 |
| 10.0000021 | -6.0000013 | -3.0000006 | 2.0000003 |

5/ Ger hopp till läge singular.

Det till synes dåliga resultatet i 3 och 4 beror på att matriserna är illa konditionerade.

Test av <u>första stegets minsta-kvadrat-skattning</u>.

Kontroll har gjorts med minsta-kvadrat-skattning från program i ref.(4). För att få jämförbart resultat har använts en dataserie om 100 punkter, där de 10 första punkterna har värdet noll; de övriga 90 är tagna ur serien, som använts i exempel 2 ($\lambda$=1.0, n=2).
Då skattningarna göres på helt olika sätt, är det överensstämmande resultatet en god indikation på att kodningen är riktigt utförd.

Verkliga koeff.: $a_1$=-1.5 $b_1$=1.0 $c_1$=-1.0
$a_2$= 0.7 $b_2$=0.5 $c_2$= 0.2

Skattade koeff.:

| | enl.PRIDE | enl. ref.(4) |
|---|---|---|
| n=1 | $a_1$=-0.877 | $a_1$=-0.88 |
| | $b_1$= 1.004 | $b_1$= 1.00 |
| | V =174,377 | V =174.377 |
| n=2 | $a_1$=-1.246 | $a_1$=-1.25 |
| | $a_2$= 0.460 | $a_2$= 0.46 |
| | $b_1$= 0.739 | $b_1$= 0.74 |
| | $b_2$= 0.854 | $b_2$= 0.85 |
| | V = 80.184 | V = 80.184 |
| n=3 | $a_1$=-0.939 | $a_1$=-0.94 |
| | $a_2$=-0.060 | $a_2$=-0.06 |
| | $a_3$= 0.302 | $a_3$= 0.30 |
| | $b_1$= 0.849 | $b_1$= 0.85 |
| | $b_2$= 1.009 | $b_2$= 1.01 |
| | $b_3$= 0.407 | $b_3$= 0.41 |
| | V = 65.700 | V = 65.700 |

V

KÖRDA EXEMPEL

Identifieringsalgoritmen har använts dels på kända genererade processer och dels på en verklig process.

De tänkta processerna i exempel 1 och 2 har genererats med formeln:

$$y(t)-1.5y(t-1)+0.7y(t-2) = u(t-1)+0.5u(t-2) + \\ +e(t)-e(t-1)+0.2e(t-2)$$

I exempel 3 identifieras följande system:

$$y(t) + a\,y(t-1) = u(t-1) + e(t) + c\,e(t-1)$$

där c antar värdena 0.99 och 0.0 och a antar värdet -0.9.

Som insignal har valts +1 och -1 så medelvärdet av u över mätintervallet blir noll. Såsom brus, e(t), har använts normalfördelade slumptal. Dessa har genererats från rektangelfördelade slumptal, vilka erhållits med tillgänglig maskinprocedur för SMIL.

Genereringarna har utförts av Kurt-Erik Eriksson och är beskrivna i ref.(4).

I exempel 4 identifieras ett okänt system.

Exempel 5 är ett problem med enkel regression.

Exempel 6 behandlar ett 1:a ordningens system jämförbart med exempel 3.

I exempel 7 undersökes ett oscillativt system.

Exempel 1

Tänkt process:

$$y(t)-1.5y(t-1)+0.7y(t-2) = u(t-1)+0.5u(t-2) + \\ + e(t)-e(t-1)+0.2e(t-2)$$

Antalet mätpunkter är 100 och e är slumptal N(0,0.75).

Processen skattas som ett 2:a ordningens system.

Resultat:

$$a_1 = -1.47 \pm 0.03$$
$$a_2 = 0.69 \pm 0.02$$
$$b_1 = 0.84 \pm 0.08$$
$$b_2 = 0.79 \pm 0.11$$
$$c_1 = -0.84 \pm 0.10$$
$$c_2 = 0.01 \pm 0.10$$

Skattningens förlopp: se tabell.

Vissa konvergensbesvär har erhållits då $c_2$-koeff. i början är negativ. Vid full korrektion av koeff. (alfa=1) skulle C-polynomet i vissa steg fått nollställen utom enhetsciekeln. Sämst skattat är $b_2$, vars värde ligger 2.5 standardavvikelser från det genererade.

Tabell 1 Exempel 1

(ex. anger exakt andraderivata)

| ℓ | alfa | $a_1$ | $a_2$ | $b_1$ | $b_2$ | $c_1$ | $c_2$ | V | λ |
|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 0 | 0 | 0 | 0 | 0 | 1573 | |
| 0 | 1 | -1.38 | 0.60 | 0.87 | 0.83 | 0 | 0 | 46 | 0.96 |
| 1 | 0.3 | -1.43 | 0.65 | 0.87 | 0.76 | -0.22 | -0.09 | 39 | 0.88 |
| 2 | 0.3 | -1.46 | 0.68 | 0.87 | 0.72 | -0.42 | -0.15 | 34 | 0.83 |
| 3 | 0.3 | -1.4723 | 0.6846 | 0.8644 | 0.7219 | -0.551 | -0.155 | 31.30 | 0.79 |
| 4 | 0.3 | -1.4739 | 0.6868 | 0.8573 | 0.7358 | -0.638 | -0.125 | 29.99 | 0.77 |
| 5 | 0.5 | -1.4734 | 0.6877 | 0.8490 | 0.7567 | -0.738 | -0.070 | 29.07 | 0.76 |
| 6 | 0.5 | -1.4724 | 0.6875 | 0.8443 | 0.7695 | -0.788 | -0.031 | 28.83 | 0.76 |
| 7 ex | 1 | -1.4699 | 0.6865 | 0.8355 | 0.7906 | -0.8367 | 0.0103 | 28.7532 | 0.758 |
| 8 ex | 1 | -1.4700 | 0.6865 | 0.8352 | 0.7905 | -0.8375 | 0.0118 | 28.7531 | 0.758 |
| 9 ex | 1 | -1.4700 | 0.6865 | 0.8352 | 0.7905 | -0.8375 | 0.0118 | | |
| | $\sigma_i$ = | 0.023 | 0.019 | 0.080 | 0.108 | 0.100 | 0.100 | | |

Exempel 2

Samma process som i exempel 1 men $e \in N(0,1)$.

Antal mätpunkter = 100.

A. Skattas som 1:a ordningens system.

| l | alfa | $a_1$ | $b_1$ | $c_1$ | V | M |
|---|---|---|---|---|---|---|
| | | 0 | 0 | 0 | 1325 | 100 |
| 0 | 1 | -0.87 | 1.05 | 0 | 184 | 100 |
| 1 | 0.5 | -0.84 | 1.13 | 0.23 | 164 | 80 |
| 2 | 0.5 | -0.84 | 1.03 | 0.30 | 160 | 60 |
| 3 | 0.5 | -0.84 | 0.93 | 0.34 | 157 | 60 |
| 4 | 0.5 | -0.8468 | 0.87 | 0.37 | 156.31 | 60 |
| 5 | 1 | -0.8500 | 0.78 | 0.403 | 155.62 | 60 |
| 6ex | 1 | -0.8504 | 0.750 | 0.412 | 155.56 | 60 |
| 7ex | 1 | -0.8502 | 0.743 | 0.415 | | |
| | $\sigma_i$ = | 0.06 | 0.2 | 0.1 | | |

$\lambda = 1.829$

Koeff: $a_1 = -0.85 \pm 0.06$, $b_1 = 0.7 \pm 0.2$, $c_1 = 0.4 \pm 0.1$

B. Skattas som 2:a ordningens system.

Se tabell nästa sida.

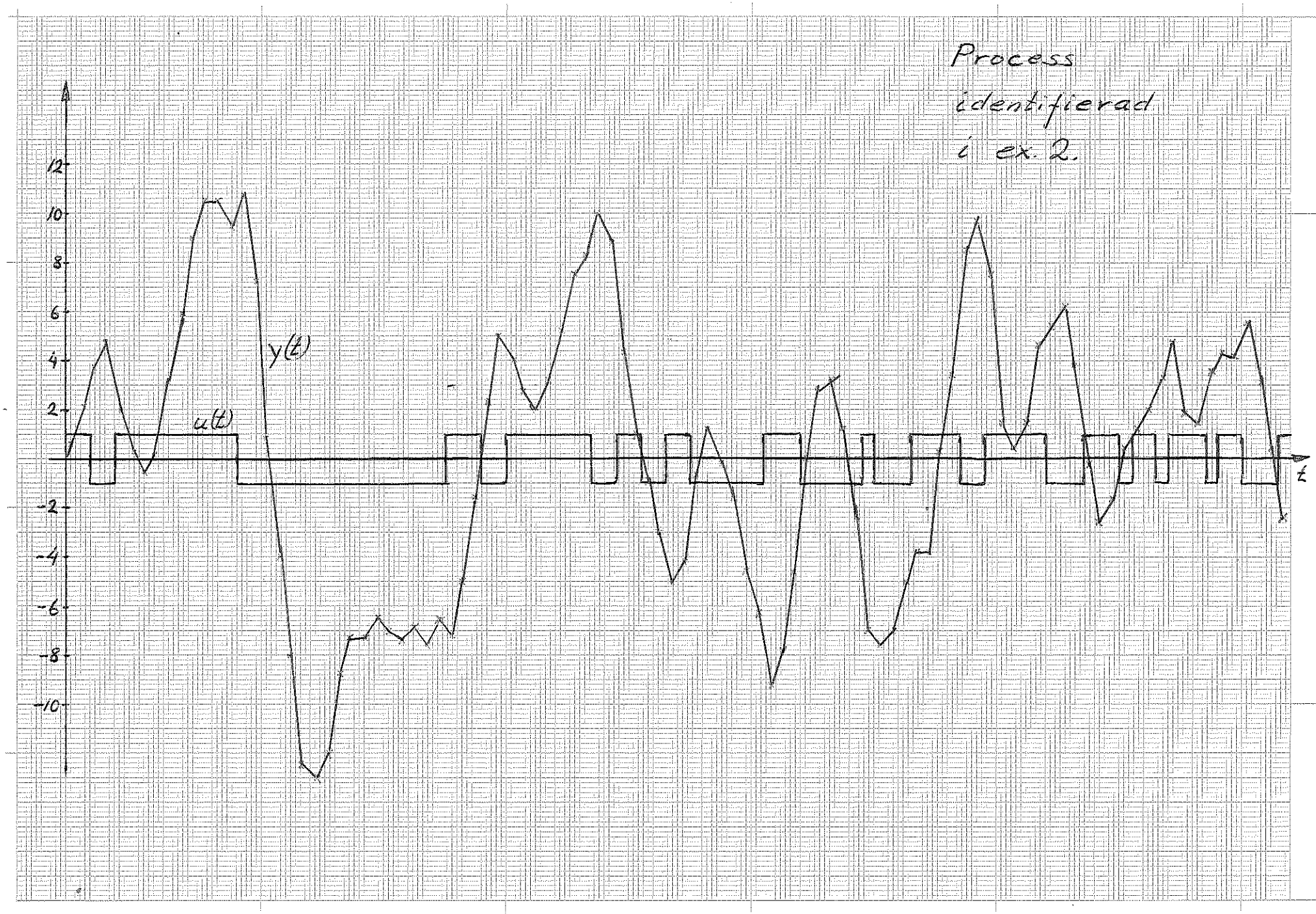
Koeff. blir:

$a_1 = -1.54 \pm 0.04$ $\quad a_2 = 0.73 \pm 0.03$

$b_1 = 0.99 \pm 0.12$ $\quad b_2 = 0.45 \pm 0.15$

$c_1 = -0.94 \pm 0.11$ $\quad c_2 = 0.04 \pm 0.10$

Med undantag av $c_2$ ligger de skattade värdena på mindre än en standardavvikelse från de genererade värdena.



Tabell Exempel 2B

| ℓ | alfa | $a_1$ | $a_2$ | $b_1$ | $b_2$ | $c_1$ | $c_2$ | V | M |
|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 0 | 0 | 0 | 0 | 0 | 1325 | 200 |
| 0 | 1 | -1.25 | 0.46 | 0.82 | 0.82 | 0 | 0 | 86 | 300 |
| 1 | 0.5 | -1.43 | 0.62 | 0.89 | 0.60 | -0.38 | -0.13 | 70 | 1300 |
| 2 | 0.5 | -1.52 | 0.71 | 0.94 | 0.47 | -0.66 | -0.15 | 59 | 3500 |
| 3 | 0.5 | -1.5444 | 0.7246 | 0.9723 | 0.4530 | -0.7999 | -0.0928 | 55.83 | 7000 |
| 4 | 0.5 | -1.5435 | 0.7254 | 0.9788 | 0.4552 | -0.8668 | -0.0319 | 54.97 | 7500 |
| 5 | 1 | -1.5436 | 0.7268 | 0.9869 | 0.4525 | -0.9399 | 0.0383 | 54.6621 | 7800 |
| 6 ex | 1 | -1.54405 | 0.72707 | 0.98732 | 0.45048 | -0.93929 | 0.038255 | 54.6618 | 7800 |
| 7 ex | 1 | -1.54405 | 0.72707 | 0.98732 | 0.45048 | -0.93928 | 0.038256 | | |
| | $\sigma_i$ = | 0.04 | 0.03 | 0.12 | 0.15 | 0.11 | 0.10 | | |

$\lambda$ = 1.084

Tabell Exempel 2C

| ℓ | alfa | $a_1$ | $a_2$ | $a_3$ | $b_1$ | $b_2$ | $b_3$ | $c_1$ | $c_2$ | $c_3$ | V | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1325 | 500 |
| 0 | 1 | -0.95 | -0.05 | 0.30 | 0.93 | 1.00 | 0.39 | 0 | 0 | 0 | 71 | 1900 |
| 1 | 0.4 | -1.42 | 0.56 | 0.06 | 0.92 | 0.59 | 0.03 | -0.55 | 0.05 | -0.09 | 62 | 220000 |
| 2 | 0.2 | -1.93 | 1.32 | -0.28 | 0.94 | 0.12 | -0.24 | -1.12 | 0.33 | -0.13 | 58 | 540000 |
| 3 | 0.2 | -1.93 | 1.32 | -0.28 | 0.95 | 0.11 | -0.22 | -1.15 | 0.35 | -0.11 | 56 | 810000 |
| 4 | 0.2 | -1.922 | 1.308 | -0.276 | 0.958 | 0.110 | -0.202 | -1.184 | 0.361 | -0.105 | 56.06 | $0.6 \cdot 10^6$ |
| 5 | 0.5 | -1.909 | 1.294 | -0.269 | 0.973 | 0.099 | -0.165 | -1.244 | 0.395 | -0.087 | 54.89 | $0.6 \cdot 10^6$ |
| 6 ex | 1 | -1.956 | 1.371 | -0.305 | 0.989 | 0.030 | -0.152 | -1.363 | 0.487 | -0.069 | 54.54 | $2.6 \cdot 10^6$ |
| 7 ex | 1 | -1.873 | 1.241 | -0.244 | 0.990 | 0.109 | -0.117 | -1.277 | 0.416 | -0.075 | | |

C. Skattas som 3:e ordningens system.

Skattningens förlopp: se tabell.

Koeff blir:

$a_1 = -1.9 \pm 0.9$ $\quad a_2 = 1.2 \pm 1.4$ $\quad a_3 = -0.2 \pm 0.7$

$b_1 = 1.0 \pm 0.1$ $\quad b_2 = 0.1 \pm 0.9$ $\quad b_3 = -0.1 \pm 0.4$

$c_1 = -1.3 \pm 0.9$ $\quad c_2 = 0.4 \pm 0.8$ $\quad c_3 = -0.1 \pm 0.1$

M i tabellerna är ett approximativt konditionstal ( se ref.(5))) för andraderivatmatrisen. Den dåliga konvergensen och den stora spridningen på koeff. i skattning C är en följd av de stora konditionstalen.

En anledning till de höga konditionstalen kan vara att de rader i matrisen,som tillkommit för 3:e ordningens system, är linearkombinationer av övriga rader. Detta skulle innebära att systemet i detta fall är av 2:a ordningen.

Vi undersöker förlustfunktionen.

$$V_1 = 155.6 \quad V_2 = 54.7 \quad V_3 = 54.5$$

Då felen är normalfördelade fås: (N=100)

$$V_1/\sigma^2 \in \chi^2(97) \; ; \quad V_2/\sigma^2 \in \chi^2(94) \; ; \quad V_3/\sigma^2 \in \chi^2(91) \; ;$$

Vi vill statistiskt testa om sänkningen av förlustfunktionen är signifikant då vi ökar systemordningen.

$V_i$ är ej säkert oberoende men däremot $(V_i - V_{i+1})$ och $V_i$.

Vi bildar uttrycket:

$$\frac{V_i - V_{i+1}}{n_i - n_{i+1}} \frac{n_i}{V_i}$$ där $n_j$ är antal frihetsgrader.

Detta uttryck antages ha fördelningen $F(n_i - n_{i+1}, n_i)$.

Betrakta 5%-nivån.

$$F_{0.05}(3,97) = 2.70 \qquad F_{0.05}(3,94) = 2.71$$

Men

$$\frac{V_1 - V_2}{3}\,\frac{97}{V_1} = 21.0 \qquad \frac{V_2 - V_3}{3}\,\frac{94}{V_2} = 0.12.$$

Då ordningstalet ökas från 1 till 2 fås en signifikant sänkning av förlustfunktionen. Ökning av ordningstalet till 3 ger ingen signifikant sänkning på denna nivå. Härav slutes att systemordningen är 2 med de koeff. som erhållits under B.

Exempel 3

Detta exempel visar hur svårigheter kan uppstå då minimipunkten på förlustfunktionen ligger nära stabilitetsgränsen.

Betrakta följande system:

$$y(t) + a \cdot y(t-1) = u(t-1) + e(t) + c \cdot e(t-1) \quad ; \quad e \in N(0,1)$$

Sätt a=-0.9 och c=0.0 och identifiera från mätserie med 100 mätpunkter. Resultat:

| l | alfa | a | b | c | V |
|---|---|---|---|---|---|
| | | 0 | 0 | 0 | 1069.6 |
| 0 | 1 | -0.9052 | 0.9502 | 0 | 52.3388 |
| 1 | 0.5 | -0.9034 | 0.9490 | 0.0389 | 52.0995 |
| 2 | 1 | -0.9009 | 0.9504 | 0.0853 | 51.9720 |
| 3 | 1 | -0.9002 | 0.9534 | 0.0941 | 51.9668 |
| 4 ex | 1 | -0.9000 | 0.9539 | 0.0960 | 51.9665 |
| 5 ex | 1 | -0.9000 | 0.9541 | 0.0964 | |

$\lambda = 1.0195$

$\sigma_a = 0.025$ $\quad \sigma_b = 0.108$ $\quad \sigma_c = 0.100$

$a = -0.90 \pm 0.03$ $\quad b = 0.95 \pm 0.11$ $\quad c = 0.10 \pm 0.10$

Här finns inga svårigheter med konvergens.

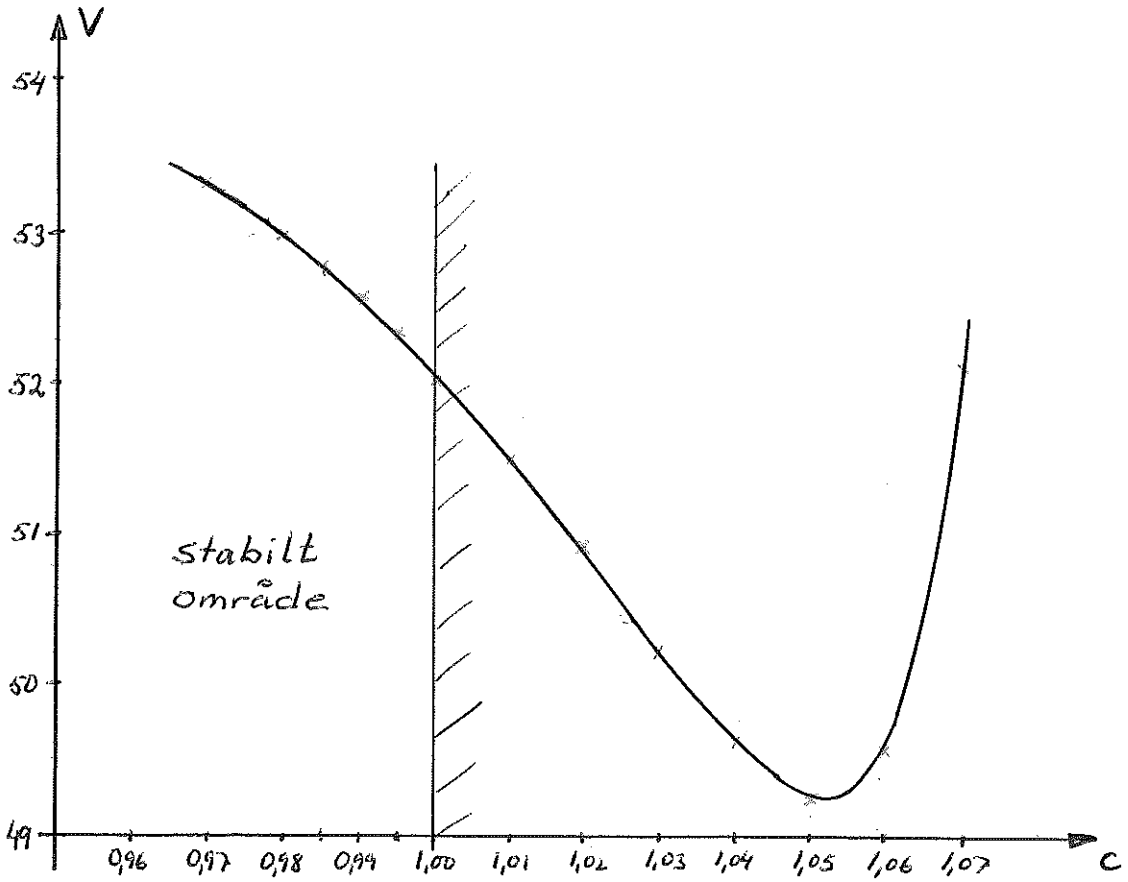
Generera i stället samma system med c=0.99. Antal mätpunkter samma = 100. Resultat:

| l | alfa | a | b | c | V |
|---|---|---|---|---|---|
| | | 0 | 0 | 0 | 2052.0 |
| 0 | 1 | -0.920 | 0.991 | 0 | 111.3 |
| 1 | 0.5 | -0.906 | 0.983 | 0.257 | 88.9 |
| 2 | 0.5 | -0.895 | 0.992 | 0.506 | 73.6 |
| 3 | 0.5 | -0.889 | 0.993 | 0.728 | 62.9 |
| 4 | 0.5 | -0.887 | 0.978 | 0.898 | 55.9 |
| 5 | 0.5 | -0.890 | 0.957 | 0.985 | 52.89 |
| 6ex | 0.2 | -0.891 | 0.968 | 0.992 | 52.56 |
| 7ex | 0.2 | -0.891 | 0.979 | 0.999 | 52.20 |
| 8ex | 0.2 | -0.891 | 0.989 | 1.007 | 51.75 |

Detta c-värde ger instabilt system.

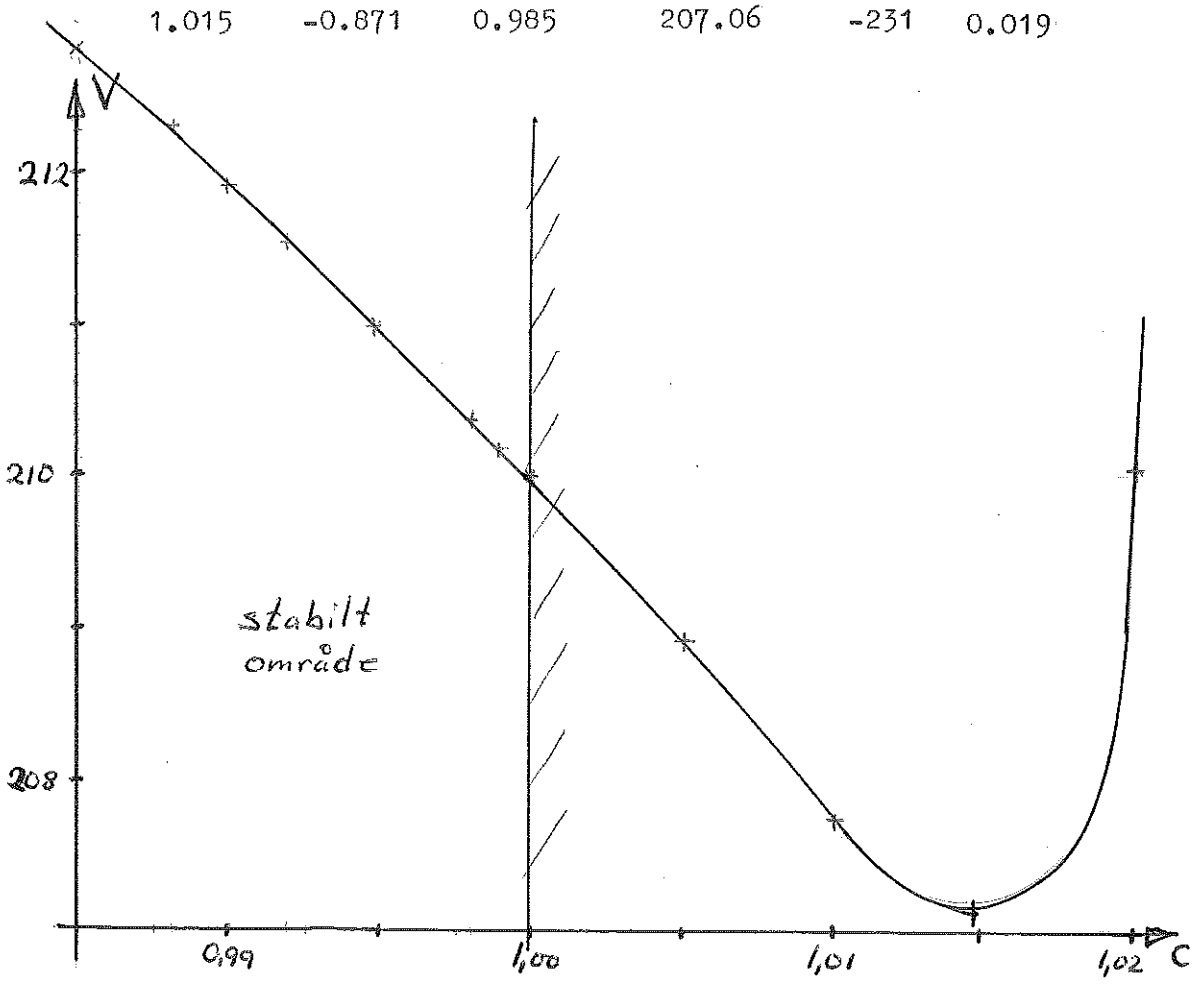
Värdet på c ökar även då det överstigit det genererade värdet 0.99. Finns det någon minimipunkt inom stabilitetsområdet? Pröva genom att läsa in några lämpliga c-värden och bestäm motsvarande minsta-kvadrat-skattning av a och b.

| Inläst | Minsta-kvadrat-sk. | | | | |
|---|---|---|---|---|---|
| c | a | b | V | $\frac{dV}{dc}$ | $\sigma_c$ |
| 0.97 | -0.90 | 0.98 | 53.32 | -35.3 | 0.094 |
| 0.98 | -0.90 | 0.99 | 52.95 | -38.5 | 0.064 |
| 0.985 | -0.90 | 0.99 | 52.75 | -40.9 | 0.063 |
| 0.99 | -0.90 | 0.99 | 52.54 | -43.8 | 0.074 |
| 0.995 | -0.90 | 0.99 | 52.31 | -47.2 | 0.157 |
| 1.00 | -0.90 | 1.00 | 52.07 | -50.9 | 0.095 |
| 1.01 | -0.89 | 1.00 | 51.52 | -58.9 | 0.141 |
| 1.02 | -0.90 | 1.00 | 50.89 | -65.4 | 0.077 |
| 1.03 | -0.90 | 0.98 | 50.23 | -66.5 | 0.017 |
| 1.04 | -0.90 | 0.96 | 49.60 | -55.3 | 0.007 |
| 1.05 | -0.90 | 0.92 | 49.21 | -16.2 | 0.003 |
| 1.06 | -0.90 | 0.85 | 49.52 | 99.8 | 0.002 |
| 1.07 | -0.91 | 0.73 | 52.04 | 483.4 | 0.001 |



Av denna undersökning att döma finns ingen minimipunkt inom stabilitetsområdet. Anledningen till detta kan vara den korta mätserien. Undersök i stället en serie om 400 punkter. Minsta-kvadrat-skattning för olika c-värden ger:

| Inläst | Minsta-kvadrat-sk. | | | | |
|---|---|---|---|---|---|
| c | a | b | V | $\frac{dV}{dc}$ | $\sigma_c$ |
| 0.985 | -0.866 | 0.983 | 212.79 | -167 | 0.021 |
| 0.988 | -0.866 | 0.984 | 212.28 | -173 | 0.022 |
| 0.990 | -0.866 | 0.984 | 211.93 | -177 | 0.023 |
| 0.992 | -0.866 | 0.984 | 211.57 | -181 | 0.028 |
| 0.995 | -0.866 | 0.985 | 211.02 | -190 | 0.026 |
| 0.998 | -0.866 | 0.985 | 210.43 | -199 | 0.018 |
| 0.999 | -0.866 | 0.985 | 210.23 | -204 | 0.008 |
| 1.000 | -0.866 | 0.984 | 210.02 | -208 | 0.007 |
| 1.005 | -0.866 | 0.984 | 208.93 | -229 | 0.003 |
| 1.010 | -0.866 | 0.983 | 207.74 | -243 | 0.001 |
| 1.020 | -0.866 | 0.982 | 210.05 | 48222 | 0.000 |
| 1.015 | -0.871 | 0.985 | 207.06 | -231 | 0.019 |



Fortfarande ligger minimipunkten utanför stabilitets-området. c-värdet för minimum har dock minskat från ca 1.055 till ca 1.015. Vi gör samma beräkningar för en serie om 800 punkter.

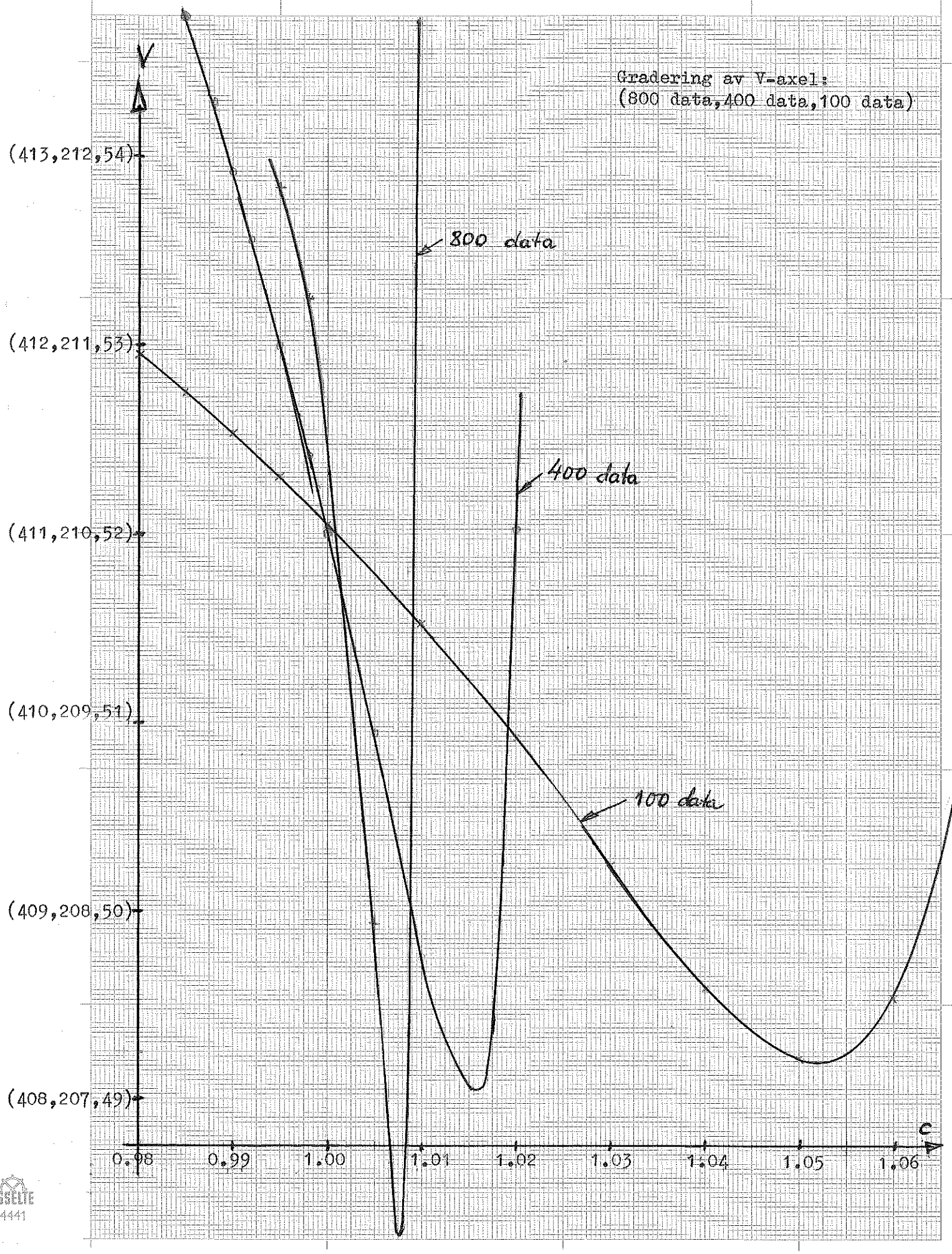
| Inläst | Minsta-kvadrat-sk. | | | | |
|---|---|---|---|---|---|
| c | a | b | V | $\frac{dV}{dc}$ | $\sigma_c$ |
| 0.995 | -0.892 | 0.989 | 412.83 | -144 | 0.005 |
| 0.998 | -0.892 | 0.989 | 412.25 | -263 | 0.004 |
| 1.005 | -0.892 | 0.991 | 408.92 | -627 | 0.004 |
| 1.010 | -0.894 | 0.995 | 496.32 | 966207 | 0.002 |
| 1.007 | -0.892 | 0.992 | 407.65 | -2314 | 0.004 |

De tre serierna är samlade i ett diagram. Se nästa sida. Av diagrammet framgår att minimipunkten närmar sig det stabila området då mätserien växer. I detta fall erfordras betydligt längre mätserie för stabil skattning.

På teoretisk väg kan man bestämma förlustfunktionens variation med skattat c-värde för ett 1:a ordningens system. Detta är gjort i appendix.

Därur fås att förlustfunktionen har minimum för c=0.99 och $V \to \infty$ då $c \to 1$. Härledningen gäller för oändligt långa serier.

Diagram till Exempel 3.



Exempel 4

Som praktiskt exempel har valts ett system hämtat från ett examensarbete i elektrisk mätteknik av H.G. Karlsson. Insignalen är blodtrycket i ett ben och utsignalen är benets volymsvariation. Se diagram.

Systemet skattas för n=2 och n=3.

Antal mätpunkter: 314.

Skattningens förlopp: se tabell.

Resultat:

n=2.

$a_1 = -1.80 \pm 0.03$ $\quad a_2 = 0.83 \pm 0.03$

$b_1 = 0.10 \pm 0.03$ $\quad b_2 = -0.07 \pm 0.03$

$c_1 = -0.62 \pm 0.07$ $\quad c_2 = 0.08 \pm 0.06$

För n=3 fås instabilt system redan efter andra skattningen. Detta har ej kunnat undvikas trots att flera små alfa-värden har provats. Konditionstalet anger betydande numeriska svårigheter, vilket torde bero på lineär-beroende, och vi sluta oss direkt till att systemet är av 2:a ordningen:

$$y(t)-1.80\ y(t-1)+0.83\ y(t-2) = 0.10\ u(t-1)-0.07\ u(t-2) + e(t)-0.62\ e(t-1)+0.08\ e(t-2)$$

där e(t) är normalfördelade (0;3.4), och standardavvikelser på koeff. enligt ovan.

Tabell Exempel 4

| | ℓ | alfa | $a_1$ | $a_2$ | $b_1$ | $b_2$ | $c_1$ | $c_2$ | V | λ | M |
|---|---|---|---|---|---|---|---|---|---|---|---|
| n = 2 | | | 0 | 0 | 0 | 0 | 0 | 0 | 401529 | | 1000 |
| | 0 | 1 | -1.61 | 0.71 | -0.028 | 0.102 | 0 | 0 | 2118.5 | 3.67 | 5000 |
| | 1 | 1 | -1.809 | 0.857 | 0.043 | -0.011 | -0.478 | -0.007 | 1841.9 | 3.42 | 7000 |
| | 2 | 0.5 | -1.795 | 0.840 | 0.064 | -0.034 | -0.521 | 0.050 | 1808.9 | 3.39 | 10000 |
| | 3 | 0.5 | -1.7914 | 0.835 | 0.075 | -0.046 | -0.552 | 0.070 | 1800.6 | 3.387 | 10000 |
| | 4 | 1 | -1.7915 | 0.832 | 0.087 | -0.060 | -0.595 | 0.084 | 1795.4 | 3.3817 | 10000 |
| | 5 | 1 | -1.7944 | 0.8335 | 0.0911 | -0.0657 | -0.6081 | 0.0798 | 1794.6 | 3.3809 | 12000 |
| | 6 ex | 1 | -1.79545 | 0.83331 | 0.09527 | -0.07086 | -0.61623 | 0.07816 | 1794.4 | 3.3808 | 12000 |
| | 7 ex | 1 | -1.79535 | 0.83327 | 0.09512 | -0.07067 | -0.61588 | 0.07831 | | | |
| | | $\sigma_i$ = | 0.026 | 0.023 | 0.024 | 0.028 | 0.067 | 0.056 | | | |

| | ℓ | alfa | $a_1$ | $a_2$ | $a_3$ | $b_1$ | $b_2$ | $b_3$ | $c_1$ | $c_2$ | $c_3$ | V | λ | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| n = 3 | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 401529 | | 12000 |
| | 0 | 1 | -1.45 | 0.32 | 0.20 | -0.08 | 0.29 | -0.16 | 0 | 0 | 0 | 1795 | 3.4 | 200000 |
| | 1 | 0.1 | -1.52 | 0.44 | 0.15 | -0.08 | 0.29 | -0.17 | -0.09 | -0.01 | -0.02 | - | | |

Diagram till exempel 4.

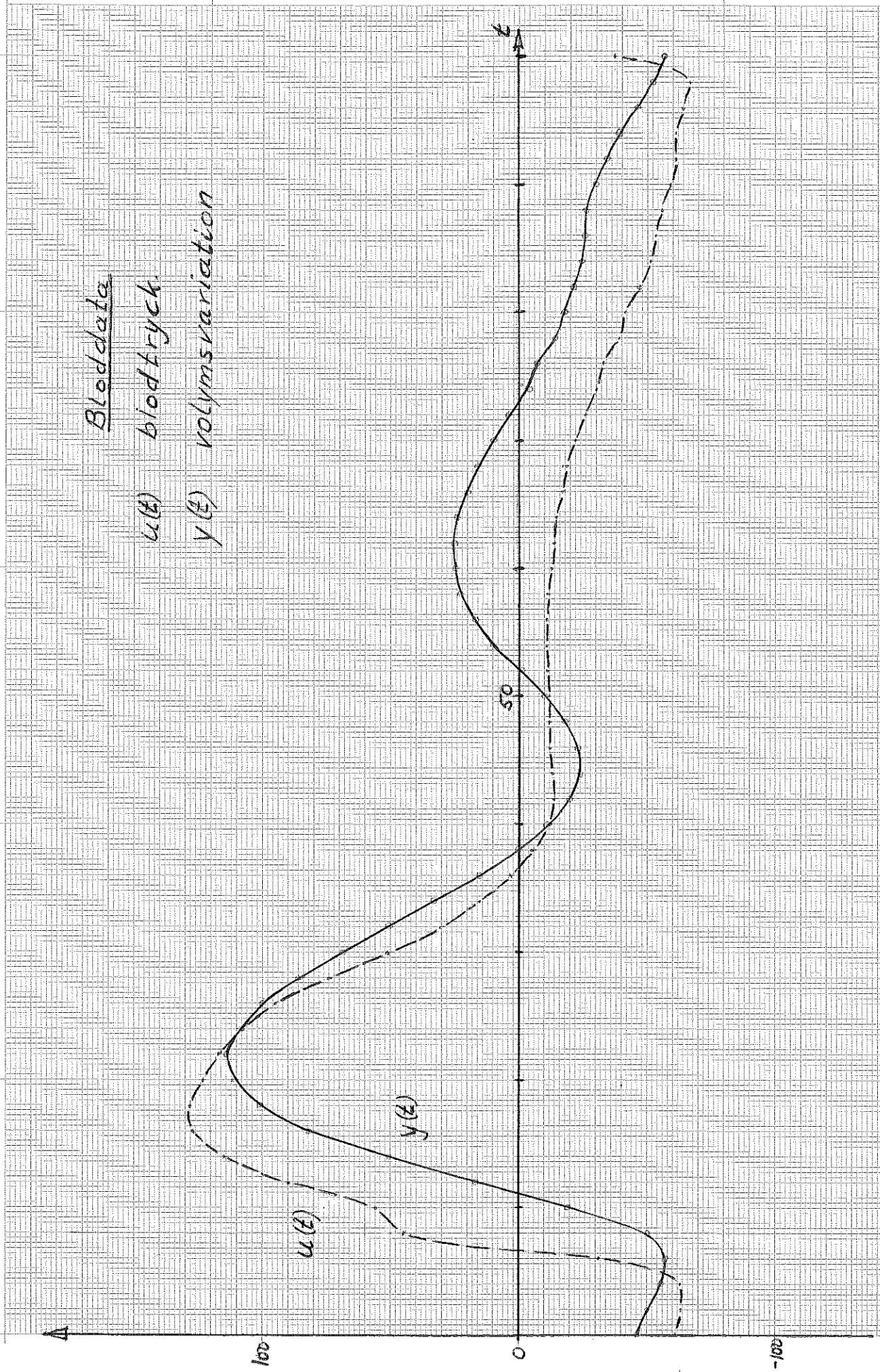


Diagram till exempel 4

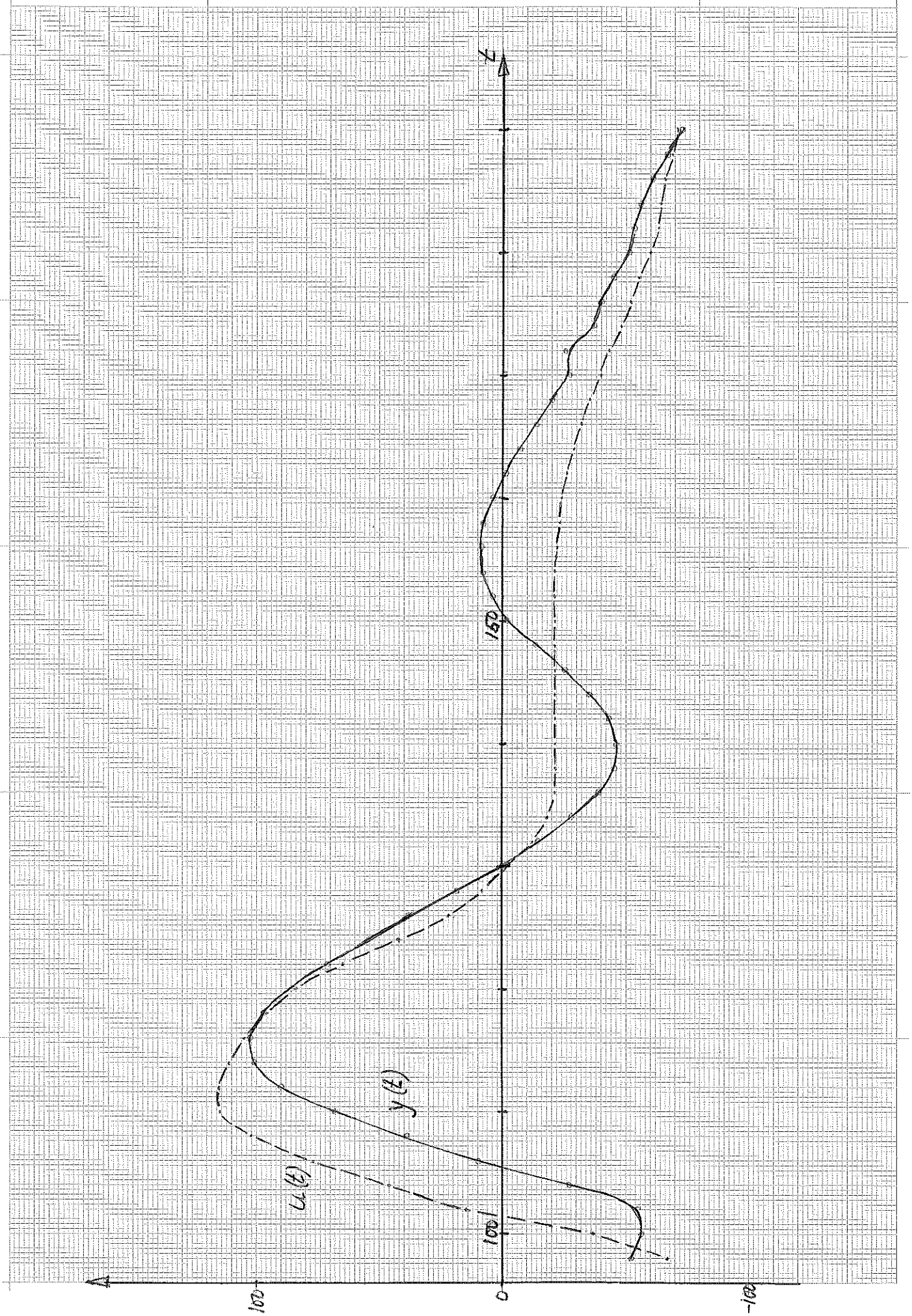


Diagram till exempel 4

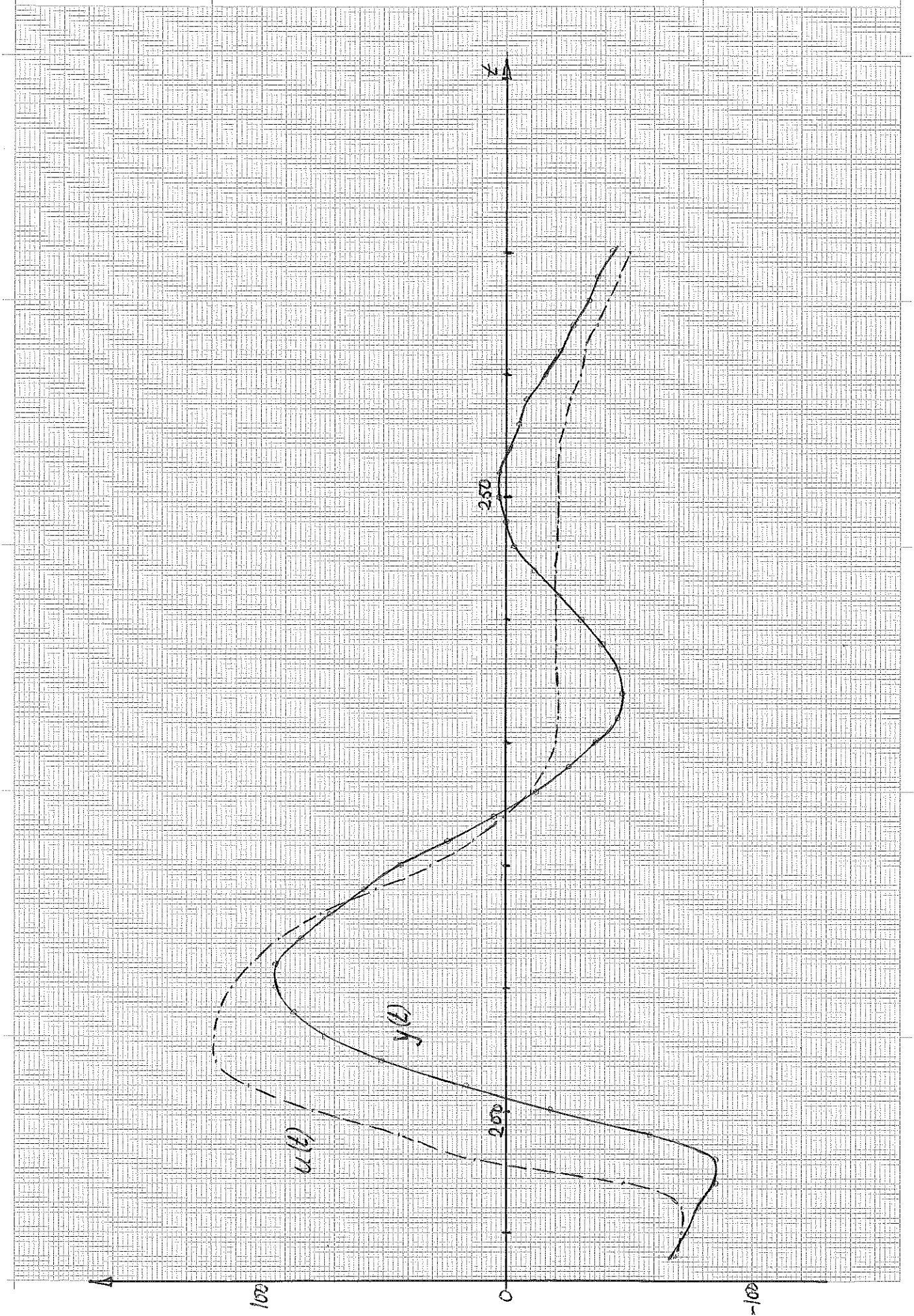
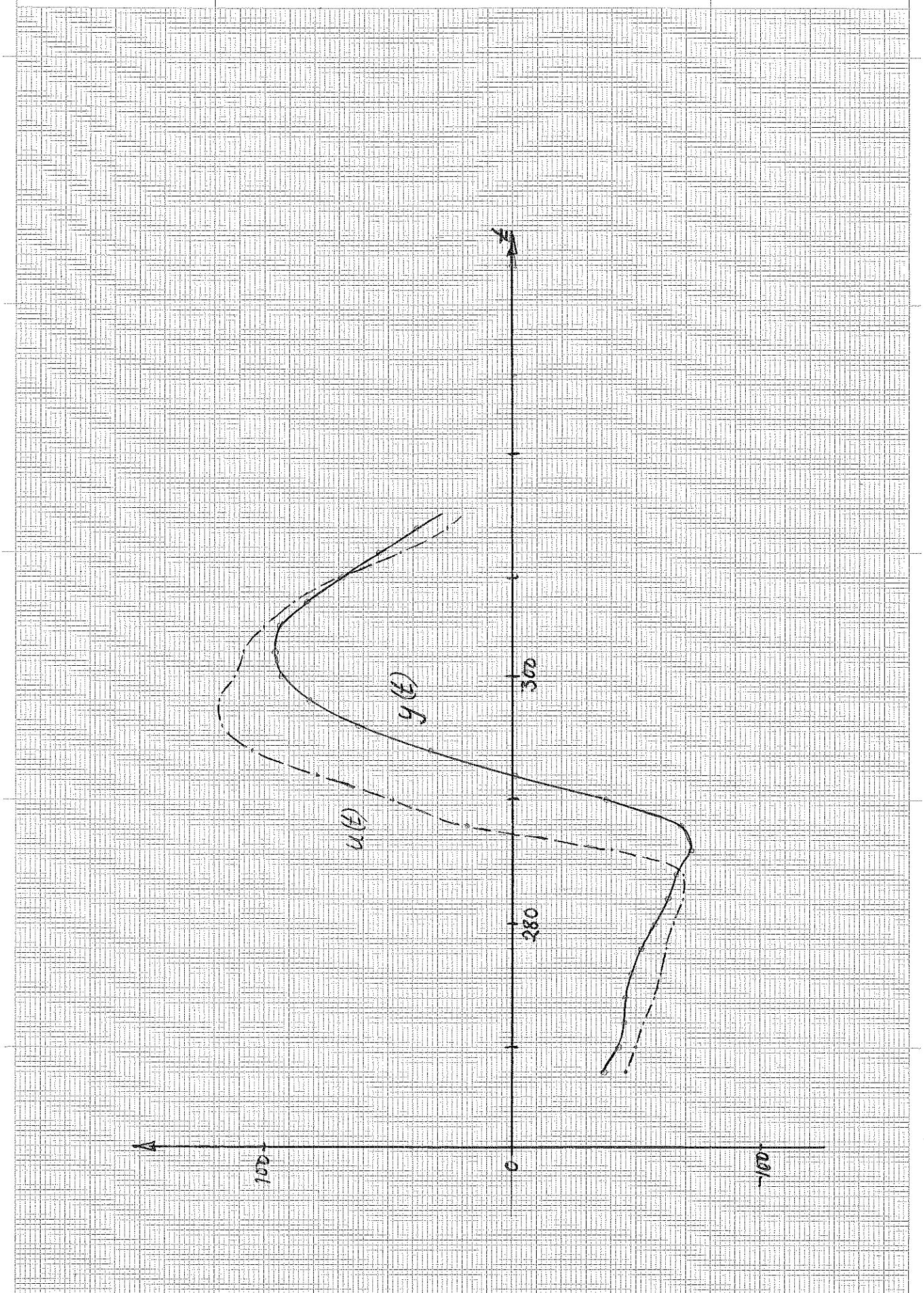


Diagram till exempel 4



Exempel 5

Med enkla korrigeringar av programmet kan vissa special-fall av det allmänna identifieringsproblemet lösas. I många fall är det tillräckligt att hålla vissa koeff. noll. Detta erhålles på samma sätt som första stegets minsta-kvadrat-skattning, där alla derivator med avseende på c-koeff. nollställes ( se III.4 ).

I detta exempel visas en enkel regressionsanalys, där ytterligare åtgärder måste vidtagas.

Vi genererar processen:

$$y(t) = 5 \cdot u(t) + e(t)$$

där $e(t) \in N(0,2)$ och $u(t) \in N(0,1)$.

Lösning:

För att kunna skatta $b_o$ måste indata förskjutas ett steg bakåt i tiden ( se III.9 ). Därvid blir $b_1$ en skattning av $b_o$. För att få en skattning av $b_1$ måste vi anropa n=1. Koeff. $a_1$ och $c_1$ skall dock vara noll. Detta fås genom nollställning av derivator enl. ovan.

Vi använder 100 mätpunkter.

Resultat: $V = 216.0$

$$\lambda = 2.10$$

$$b_1' = b_o = 5.0 \pm 0.5$$

I appendix finns den variant på PRIDE som använts samt huvudprogrammet.

Exempel 6

Genererad process:

$$y(t)-0.5\ y(t-1) = u(t-1) + e(t)+0.99\ e(t-1)$$

med $u(t) \in N(0,1)$

$e(t) \in N(0,1)$

antal mätpunkter = 500.

Detta exempel har körts i samband med motsvarande körning i ref.(4). Jämför även exempel 3.

Skattningens förlopp:

| alfa | a | b | c | V | $\lambda$ | $\frac{dV}{dc}$ |
|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 1420 | | |
| 1 | -0.69 | 1.01 | 0 | 472 | 1.37 | -343 |
| 0.5 | -0.60 | 1.01 | 0.27 | 394 | 1.26 | -262 |
| 0.5 | -0.54 | 1.01 | 0.52 | 341 | 1.17 | -213 |
| 0.5 | -0.50 | 1.01 | 0.72 | 305 | 1.10 | -179 |
| 0.5 | -0.49 | 1.02 | 0.85 | 283 | 1.06 | -161 |
| 0.2 | -0.49 | 1.02 | 0.88 | 278 | 1.05 | -160 |
| 0.2 | -0.49 | 1.02 | 0.91 | 274 | 1.05 | -160 |
| 0.2 | -0.49 | 1.01 | 0.93 | 270 | 1.04 | -161 |
| 0.2 | -0.49 | 1.01 | 0.95 | 267 | 1.03 | -162 |
| 0.2 | -0.49 | 1.01 | 0.96 | 265 | 1.03 | -160 |
| 0.4 | -0.496 | 1.008 | 0.989 | 260 | 1.02 | -117 |
| 0.4 | -0.502 | 1.006 | 0.995 | 258.8 | 1.017 | -167 |
| 0.2 | -0.504 | 1.005 | 0.998 | 258.0 | 1.016 | -206 |
| 0.2 | -0.505 | 1.005 | 0.999 | 257.4 | 1.015 | -254 |
| 0.1 | -0.506 | 1.005 | 1.001 | | | |
| $\sigma_i$ = | 0.03 | 0.02 | 0.008 | | | |

Jämför resultatet i exempel 3. Även här är mätserien för kort, för att vi skall erhålla en stabil skattning.

Se även appendix.

Exempel 7

Oscillativt system.

$$\dot{X} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} X + \begin{bmatrix} 0 \\ 1 \end{bmatrix} U$$

$$y = \begin{bmatrix} 1 \\ 0 \end{bmatrix} X + \lambda e$$

Detta system har egenvärdena $\pm 1$.

Lösning:

$$X(t+h) = \begin{bmatrix} \cos h & \sin h \\ -\sin h & \cos h \end{bmatrix} X(t) + \begin{bmatrix} 1-\cos h \\ \sin h \end{bmatrix} U(t)$$

Inför förskjutningsoperatorn z.

$$\begin{bmatrix} z-\cos h & -\sin h \\ \sin h & z-\cos h \end{bmatrix} X(t) = \begin{bmatrix} 1-\cos h \\ \sin h \end{bmatrix} U(t)$$

$$\begin{cases} x_1(t) = \dfrac{(1-\cos h)(z+1)}{z^2-2\cos h \cdot z+1} \; U(t) \\ y(t) = x_1(t) + \lambda \cdot e(t) \end{cases}$$

Med $h=0.1$ ; $\lambda = 0.5$ ; $e \in N(0,1)$ genereras 400 mätpunkter.

In- och utsignal: se diagram.

De genererade koeff. är:

$a_1 = 1.990$ $\quad b_1 = 0.005$ $\quad c_1 = a_1$

$a_2 = 1.000$ $\quad b_2 = 0.005$ $\quad c_2 = a_2$

Skattningens gång: se tabell.

En mycket dålig MK-skattning erhålles i första steget.

Ingen minimipunkt för förlustfunktionen har erhållits.

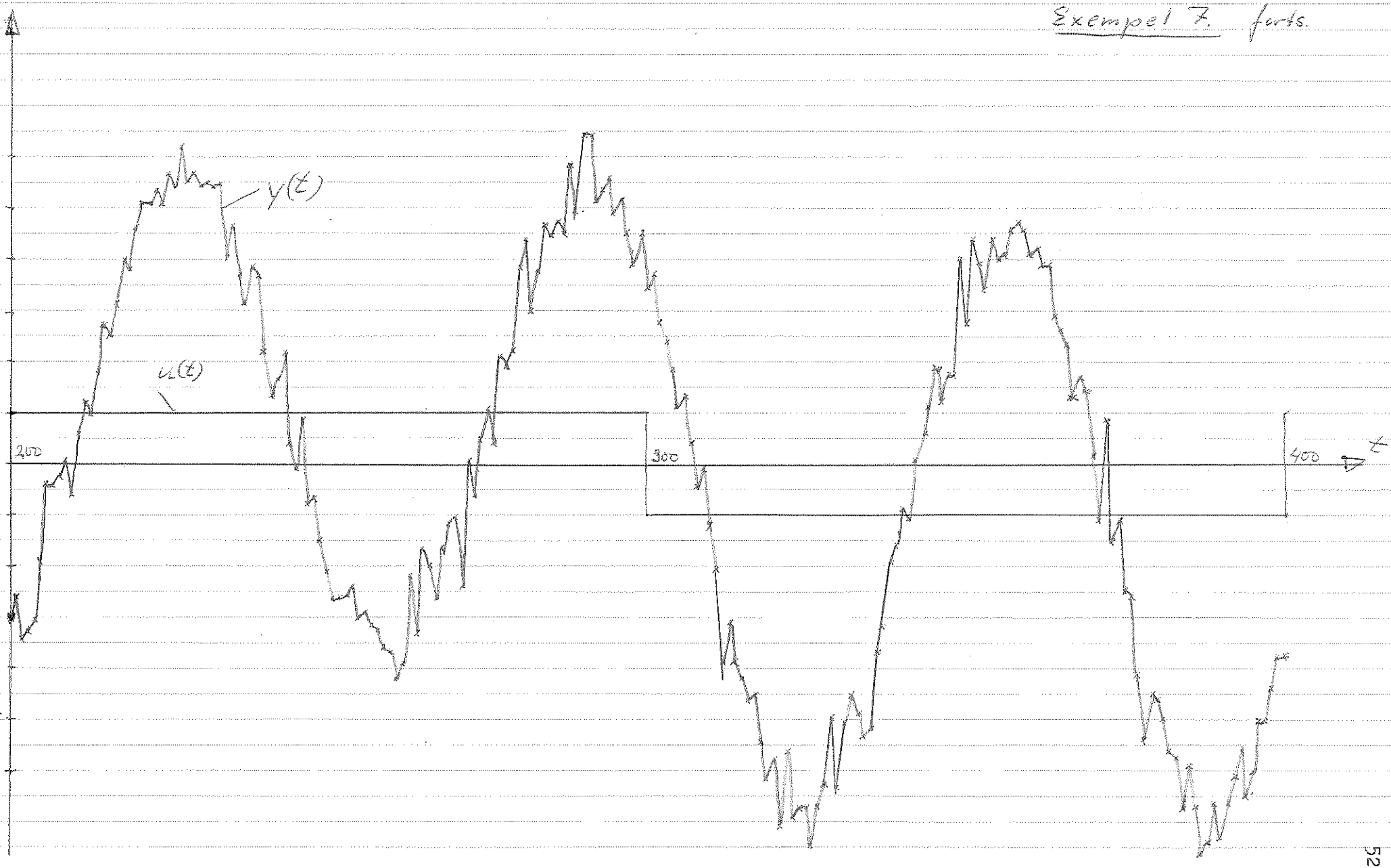
Skattningen strävar hela tiden ut ur konvergensområdet.

Jfr. de små alfa-värdena.

Tabell Exempel 7

| alfa | $a_1$ | $a_2$ | $b_1$ | $b_2$ | $c_1$ | $c_2$ | V | λ | M |
|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 2022 | | |
| 1 | -0.67 | -0.29 | 0.087 | 0.029 | 0 | 0 | 107 | 0.73 | 12000 |
| 0.5 | -1.78 | 0.76 | 0.088 | -0.060 | -1.09 | 0.38 | 120 | | |
| 0.2 | -1.92 | 0.92 | 0.061 | -0.044 | -1.27 | 0.43 | 98.0 | | |
| 0.2 | -1.999 | 1.006 | 0.048 | -0.039 | -1.42 | 0.47 | 81.9 | | |
| 0.1 | -2.006 | 1.015 | 0.030 | -0.029 | -1.47 | 0.49 | 80.6 | | |
| 0.1 | -2.004 | 1.012 | 0.022 | -0.017 | -1.51 | 0.52 | 77.0 | | |
| 0.4 | -2.000 | 1.009 | 0.010 | -0.003 | -1.58 | 0.59 | 71.1 | | |
| 0.4 | -1.997 | 1.006 | 0.007 | 0.002 | -1.64 | 0.65 | 66.9 | | |
| 0.4 | -1.994 | 1.004 | 0.009 | 0.000 | -1.75 | 0.76 | 62.1 | | |
| 0.4 | -1.993 | 1.002 | 0.012 | -0.003 | -1.85 | 0.85 | 61.2 | | 500000 |
| 0.4 | -1.991 | 1.001 | 0.015 | -0.005 | -1.86 | 0.87 | 59.2 | | |
| 0.4 | -1.9904 | 1.0004 | 0.0145 | -0.0047 | -1.879 | 0.882 | 58.1 | | |
| 0.4 | -1.9900 | 1.0000 | 0.0141 | -0.0043 | -1.902 | 0.907 | 56.9 | | |
| 0.1 | -1.9895 | 0.9995 | 0.0131 | -0.0033 | -1.9634 | 0.9712 | 55.27 | | |
| 0.5 | -1.9896 | 0.9996 | 0.0081 | 0.0020 | -1.9771 | 0.9856 | 54.02 | | |
| 0.2 | -1.9896 | 0.9996 | 0.0084 | 0.0017 | -1.9812 | 0.9897 | 54.04 | | |
| 0.05 | -1.98960 | 0.99959 | 0.00868 | 0.00139 | -1.98320 | 0.99170 | 53.86 | | $5 \cdot 10^6$ |
| 0.05 | -1.98960 | 0.99958 | 0.00880 | 0.00126 | -1.98515 | 0.99366 | 53.60 | | |
| 0.05 | -1.98961 | 0.99958 | 0.00889 | 0.00116 | -1.98713 | 0.99559 | 53.33 | | |
| 0.05 | -1.98962 | 0.99958 | 0.00897 | 0.00107 | -1.98898 | 0.99745 | 53.21 | | |
| 0.05 | -1.98962 | 0.99958 | 0.00904 | 0.00099 | -1.99076 | 0.99925 | 53.05 | | |
| 0.01 | -1.98963 | 0.99958 | 0.00905 | 0.00098 | -1.99110 | 0.99959 | 53.00 | | $8 \cdot 10^6$ |
| 0.01 | -1.98964 | 0.99959 | 0.00905 | 0.00097 | -1.99150 | 0.99990 | 52.99 | 0.51 | |
| 0.005 | -1.98964 | 0.99959 | 0.00905 | 0.00097 | -1.99175 | 0.99995 | 53.02 | | |
| $\sigma_{\theta i}$ | 0.0003 | 0.0003 | 0.0026 | 0.0026 | 0.0130 | 0.0135 | | | |
| $\frac{\partial V}{\partial_{\theta i}}$ | -8600 | -9800 | 380 | 564 | -148 | -193 | | | |

Exempel 7.


y(t)
$u_L(t)$
200
300
400
t
6
5
4
3
2
1
0
−1
−2
−3
−4
−5
−6

REFERENSER

(1) K.J. Åström, T. Bohlin:
Numerical identification of linear dynamic systems from normal operation records.

(2) K.J. Åström, T. Bohlin, S. Wensmark:
Automatic construction of linear stochastic dynamic models för stationary industrial processes with random disturbanses using operation records.

(3) B. Svärd:
Undersökning av dynamiska egenskaper hos en industriell process. (Examensarbete i regleringsteknik)

(4) K.E. Eriksson:
Numerisk bestämning av processdynamik.
(Examensarbete i regleringsteknik)

(5) C.E. Fröberg:
Lärobok i numerisk analys.

(6) T. Ekman, C.E. Fröberg:
Lärobok i algol.

(7) T. Ekman:
Större PM ang. SMIL - ALGOL.

# APPENDIX

## Utskrifter i PRIDE

Förlustfunktionen V
Materialets spridning
1:a-derivator av förlustfunktionen V1

Matris med 2:a-derivator av förlustfunktionen V2

Matris med inversen till V2,

alfa
Nya koeff.
Koefficienternas spridning

## Specialprogram för enkel regression

På följande sidor gives varianten på PRIDE och huvudprogrammet till enkel regressionsanalys. Observera att förskjutningen av indata göres vid inläsningen till yttre kärnminne.

Enkel regression

PRIDE

```
begin integer iy,x; real e,V,u,y,korr,lb;
   array E[1:8],C,EC,V1[1:24],VCC,ECC[1:48],V2[1:24,1:24];
procedure PRIDE(n,N,z,Z,t,alfa); value n,N,z,Z,t,alfa;
integer n,N,z,Z,t; real alfa;
begin integer i,j,k,l,m; m:=3×n;
   if z=1 then for i:=1 step 1 until m do C[i]:=0;
   comment start loop l;
   for l:=1 step 1 until Z do                                          } Ändring
   begin LABAN(n,N,t);                                                 } jfr. med
      if z=1 then                                                      } allmänt
      begin for i:=1 step 1 until n, 2×n+1 step 1 until m do           } program
         begin V1[i]:=0;
            for j:=1 step 1 until m do V2[i,j]:=V2[j,i]:=0;
            V2[i,i]:=1;
         end
      end;
      punch(1); print(6,5,V/2); punch(1);
      lb:=V/N; print(sqrt(lb)); punch(1);
      for i:=1 step 1 until m do print(4,4,V1[i]); punch(1);
      for i:=1 step 1 until m do
      begin punch(1);
         for j:=1 step 1 until m do print(6,2,V2[i,j])
      end;
      invers(V2,m,10-8,UT);
      for i:=1 step 1 until m do
      begin punch(1);
         for j :=1 step 1 until m do print(2,6,V2[i,j])
      end; punch(1);
      for i:=1 step 1 until m do
      begin korr:=0;
         for j:=1 step 1 until m do
         korr:=korr+V2[i,j]×V1[j];
         C[i]:=C[i]-korr×alfa;
      end korrektion av koeff;
      print(alfa); if t=1 then punch(5); punch(1);
      for i:=1 step 1 until m do print(2,4,C[i]); punch(1);
      for i:=1 step 1 until m do print(2,3,sqrt(abs(lb×V2[i,i])));
      punch(1); punch(1);
   end loop l;
   punch(1); punch(1);
end PRIDE;;
```

Enkel regression

Huvudprogram

```
    for iy:=1 step 1 until 99 do
    begin LECS(read,2×iy); LECS(read,2×iy-1 ) end;
    PRIDE(1,99,1,1,0,1);
    PRIDE(1,99,0,1,1,1);
UT: punch(3)
end
```

Teoretisk undersökning av förlustfunktionen

Vi undersöker förlustfunktionens teoretiska utseende för ett 1:a ordningens system.

Antag vi genererat en serie med $c=c_o$ och skattat värdet c.

Då gäller: $y(t) = e(t) + c_o e(t-1)$.

Skattat fel:

$$E(t) = \frac{1}{1+c\ z^{-1}}\ y(t) = \frac{1+c_o z^{-1}}{1+c\ z^{-1}}\ e(t) = H(z)\ e(t)$$

Förlustfunktionen erhåller bidraget:

$$V = \frac{1}{2\pi i} \oint H(z)\ H(z^{-1})\ \frac{dz}{z}$$

där integralen tages runt enhetscirkeln.

Fall 1. $c<1$.

$$V_1 = \frac{c_o}{c} + \frac{(-c+c_o)(1-c_o c)}{(1-c^2)\ (-c)} = \frac{1-2c_o c+c_o^2}{1-c^2}$$

$$\frac{dV}{dc}1 = \frac{(c-c_o)(1-c_o c)}{(1-c^2)^2}$$

Fall 2. $c>1$.

$$V_2 = \frac{(1-c_o)^2}{c(c^2-1)} + \frac{1+c_o^2}{c(c+1)}$$

$$\frac{dV}{dc}2 = \frac{-(1-c_o)^2(3c^2-1)}{(c(c^2-1))^2} - \frac{(1+c_o)^2(2c+1)}{(c(c+1))^2}$$

Märk:

A. $\frac{dV}{dc}1 = 0$ för $c=c_o$ då $c_o<1$

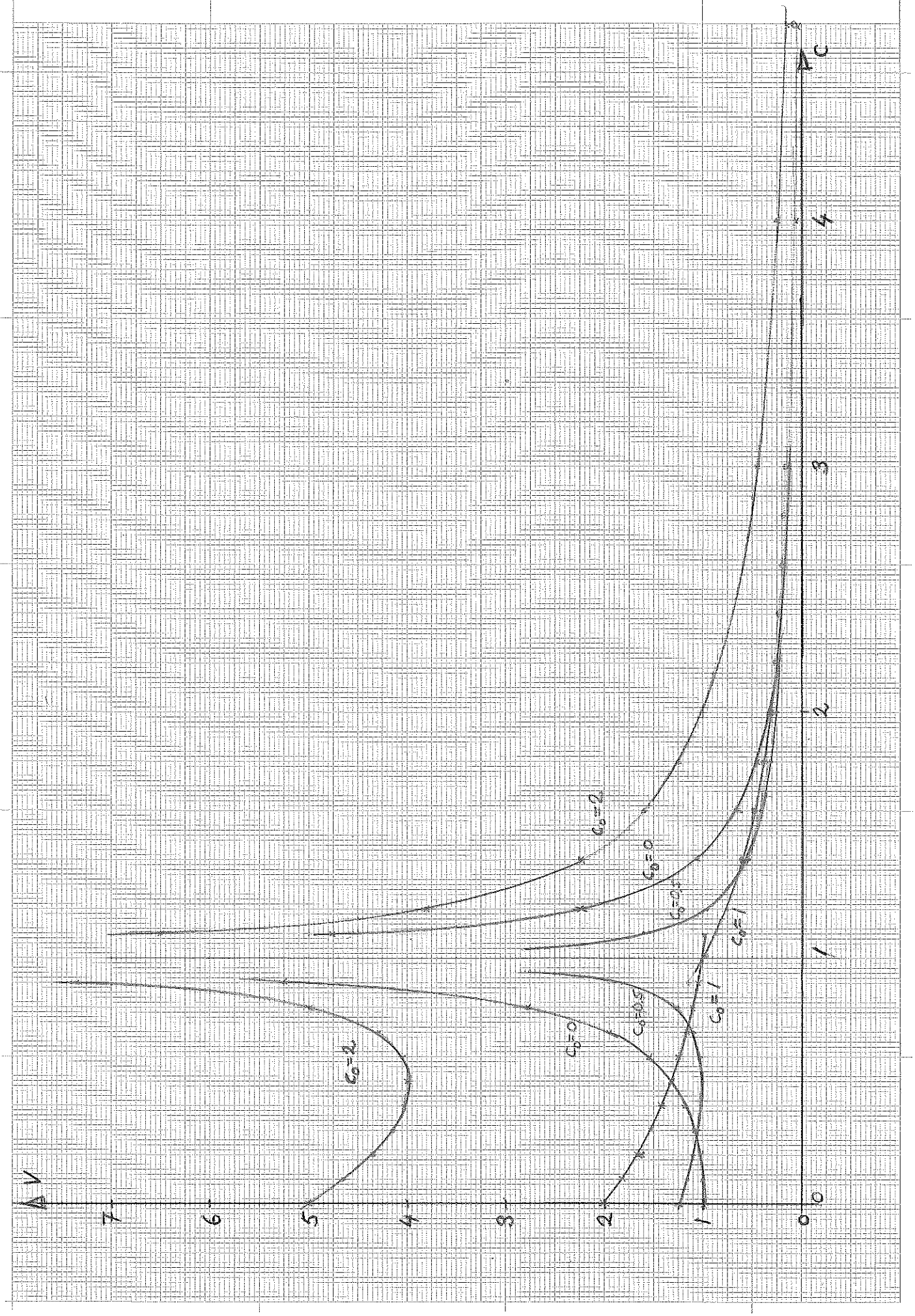
$c=1/c_o$ då $c_o>1$.

B. $\frac{dV}{dc}1 < 0$ då $c_o=1$.

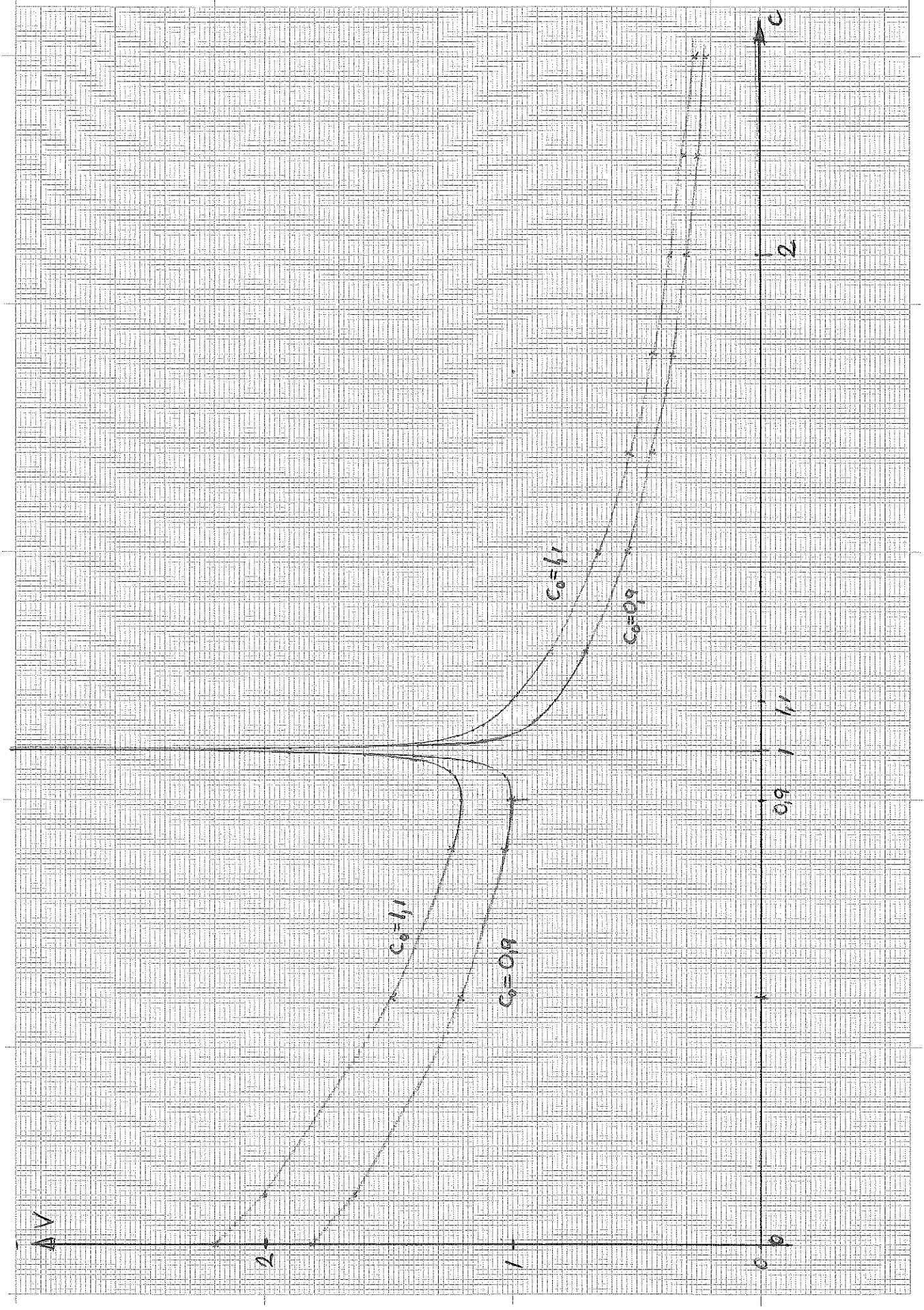
C. $\frac{dV}{dc}2 < 0$ alltid.

Se diagram.

Förlustfunktionens teoretiska utseende.

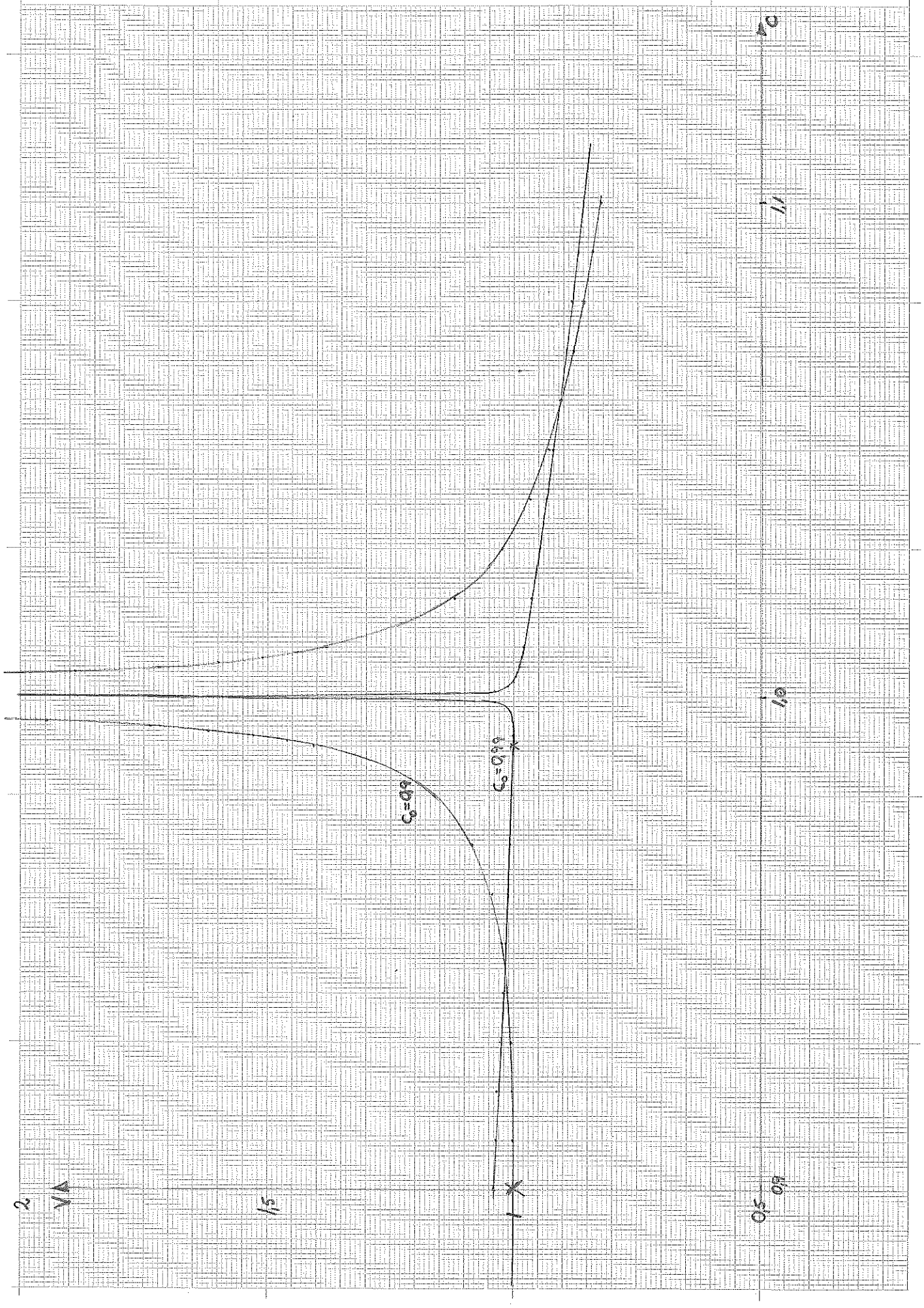


Förlustfunktionens teoretiska utseende.



Förlustfunktionens teoretiska utseende.

Förlustfunktionens teoretiska utseende. Detalj.



# Processidentifiering med datamaskin

## Rättelse i procedur Laban (sid 18)

Erik Andersson

Ordningsföljden på beräkningarna av $\frac{\partial e(t)}{\partial \theta_i}$ och $\frac{\partial^2 e(t)}{\partial \theta_i \partial \theta_j}$ (EC och ECC) medför att fel uppstår i ECC och därmed i exakta andraderivatmatrisen vid identifiering av lia ordningens system

Rad 30-38 (sid 18) skall omordnas till följande:

```
30      q:=2xn+1
        if t=1 then
        begin for i:=2xm step -1 until 2 do ECC[i]:=ECC[i-1];
            ECC[1]:=eac-EC[1];
            ECC[q]:=ebc-EC[n+1];
            ECC[2xn+q]:=ecc-2xEC[q];
        end;
        for i:=m step -1 until 2 do EC[i]:=EC[i-1];
38      EC[1]:=ea+y; EC[n+1]:=eb-u; EC[q]:=ec-e;
39      comment slut ber. av första och andra der. av e;
```

Hela sid 18 återges korrekt på nästa sida.

Denna ändring medför i exempel 2A, 6 exakt o 7 exakt att någon decimal kan vara felaktig.

Samma gäller för de två första skattningarna i exempel 3, alltså 4 exakt och 5 exakt respektive 6 exakt, 7 exakt och 8 exakt. (sid 36).

I övrigt ingen ändring.