TFRT - 5002

ALGOL - PROGRAM FÖR PROCESSIDENTIFIERING

BO SVÄRD

Rapport RE - 2 mars 1965

ALGOL - program för processidentifiering

Del av examensarbete i regleringsteknik

Bo Svärd

Undersökning av dynamiska egenskaper hos en industriell process

Examensarbete i regleringsteknik av Bo Svärd.
Godkänt den 29 mars 1966.

För att styra industriella processer är det nödvändigt att känna till såväl processdynamiken som störningarnas egenskaper. Detta examensarbete behandlar en metodik för att erhålla dessa data direkt ur mätningar på en industriell process. Arbetet omfattar två delar, en teoretisk och en praktisk del.

1. Algolprogrammering av identifieringsalgoritm på SMIL samt testkörning av programmet med artificiellt genererade data.

Identifieringsproblemet blir ett statistiskt parameteruppskattningsproblem vars lösning är tidigare känd. Parametrarna uppskattas med maximum - likelihoodmetoden. Likelihoodfunktionen maximeras med en gradientmetod. Den styrlag som minimerar utsignalens spridning erhålles ur modellens parametrar genom enkla räkningar. Den modell som framtages kan alltså direkt användas för dimensionering av en regulator för processtyrning.

2. Genomförande av mätningar samt behandling av mätdata från pappersmaskin nr 4 Gruvöns Bruk, Billeruds AB. Denna del har utarbetats tillsammans med Billeruds AB och IBM Nordiska Laboratorier. Syftet var att ta reda på hur det färdiga papprets fukthalt och ytvikt beror av ångtrycken i de olika torkcylindergrupperna i torkpartiet, samt att undersöka hur trycken skall styras för att ge så små fluktuationer som möjligt i det färdiga papprets fukthalt. En central fråga var om en digital reglering kunde ge bättre resultat än den befintliga analoga regulatorn.

Denna praktiska del omfattade planering av experimenten och deras utförande. Den databehandling som fordras för att ur de erhållna mätdata bestämma processdynamiken och störningarnas egenskaper har genomförts med en metodik som beskrives i (1). Arbetet har utförts av fil.mag. S. Wensmark, IBM Nordiska Laboratorier. En rapport (2) över denna del utarbetas i samarbete med S. Wensmark.

Resultaten av den praktiska undersökningen indikerade att man med en digital fukthaltsreglering med rimlig samplingstid kan uppnå bättre resultat än med den nuvarande analoga utrustningen. Undersökningen visade nämligen att tidsfördröjningen var större och processdynamiken långsammare än vad man tidigare förmodat.

(1) B. Svärd. ALGOL - program för processidentifiering. Rapport 6601 Institutionen för regleringsteknik LTH.

(2) B. Svärd, S. Wensmark. Undersökning av torkpartiets dynamik på pappersmaskin nr 4. Gruvön. Rapport IBM Nordiska Laboratorier.

DISPOSITION

Identifieringsprogram skrivet i ALGOL och uttestat på SMIL.

I. Teori som ligger till grund för identifieringsprogrammet

I.1. Inledning

Detta program är skrivet för numerisk identifikation av en process, när man har tillgång till in- och utsignaldata. Identifieringens ändamål är att ge styrstrategier för processregleringen. Styrstrategierna erhålles genom att vi använder linjär stokastisk styrteori. Systemet beskrives av ett bestämt antal parametrar, och identifieringsproblemet kan betraktas som ett statistiskt parameteruppskattningsproblem. I denna procedur uppskattas parametrarna genom maximum-likelihood-metoden. Den optimala ett-steg-framåt-prediktorn för utsignalen erhålles direkt från identifieringskoefficienterna. Endast stationära enkla (single input - single output) system behandlas.

Först kommer en beskrivning av den teori, som ligger till grund för programmet (jfr ref: (1)), därefter genomgås själva programmet.

I.2. Definition av problemet

I.2.1. Systemmodell som skall identifieras.

Betrakta ett diskret tids enkelt dynamiskt system vars insignal / utsignal relation ges av ekvationen

$$A(z^{-1})\, y(t) = B(z^{-1})\, u(t) + C(z^{-1})\, e(t) \qquad (2.1)$$

där { u(t) } är insignalen, { y(t) } utsignalen och { e(t) } en följd av oberoende normalfördelade $(0,\lambda)$ slumpvariabler. z betecknar skiftoperatorn.

$$z\,x(t) = x(t+1) \qquad (2.2)$$

$A(z)$, $B(z)$ och $C(z)$ är polynomen

$$A(z) = 1 + a_1 z + \ldots + a_n z^n$$

$$B(z) = b_o + b_1 z + \ldots + b_n z^n$$

$$C(z) = 1 + c_1 z + \ldots + c_n z^n \qquad (2.3)$$

Radvektorerna a,b och c med komponenterna a_i, b_i och c_i resp. introduceras även.

Följande antagande göres:

- Funktionerna $A(z^{-1})$ och $C(z^{-1})$ har alla nollställen innanför enhetscirkeln.
- Det finns inga gemensamma faktorer till alla tre polynomen $A(z)$, $B(z)$ och $C(z)$.

I.2.2. Problemet

Observationer av insignalen $\{ u(t),\ t = 1,2,\ldots,N \}$ och utsignalen $\{ y(t),\ t = 1,2,\ldots,N \}$ är givna. Sök en uppskattning av parametrarna i modellen (2.1).

I.2.3. Minimumvariansprediktionen och styralgoritmen.

Ett exempel: Antag $b_o = 0$ och $b_1 \neq 0$. Betrakta situationen vid tiden t. Datana $y(t)$, $y(t-1)$, ..., $u(t-1)$, $u(t-2)$, ... är givna, och problemet är att bestämma $u(t)$ som funktion av datana, så att variansen av $y(t)$ är minimum. Problemet är löst om $y(t+1)$ kan uttryckas som en funktion av datana ..., $y(t-1)$, $y(t)$, ..., $u(t-1)$, $u(t)$.

Lös ekvation (2.1) i termer av $y(t + 1)$:

$$y(t+1) = e(t+1) + A^{-1}(z^{-1})\, B(z^{-1})\, u(t+1) + A^{-1}(z^{-1}) \{ C(z^{-1}) - A(z^{-1}) \}\, e(t+1) \qquad (2.4)$$

Eliminera $e(t + 1)$ i sista termen genom att använda (2.1)

$$y(t+1) = e(t+1) - C^{-1}(z^{-1})\, B(z^{-1})\, zu(t) + C^{-1}(z^{-1}) \{ C(z^{-1}) - A(z^{-1}) \}\, zy(t) \qquad (2.5)$$

Enligt de antagande som gjorts finner vi att serieutvecklingen i potenser av z^{-1} av operatorerna $C^{-1}(z^{-1})\, B(z^{-1})$ och $C^{-1}(z^{-1}) \{ C(z^{-1}) - A(z^{-1}) \}$ saknar konstanta termer. Högra ledet av (2.5) beror bara på datana $y(t)$, $y(t - 1)$, ..., $u(t)$, $u(t - 1)$,... och $e(t + 1)$. Eftersom $e(t + 1)$ är oberoende av de andra termerna i högra ledet, har det önskade uttrycket erhållits. De sista två termerna i ekvation (2.5) kan sålunda tolkas som den minimala variansprediktionen av $y(t + 1)$ baserad på datana $y(t)$, $y(t - 1)$,..., $u(t)$, $u(t - 1)$,... Prediktionsfelet är $e(t + 1)$. Eftersom $e(t)$ är normalfördelad $(0, \lambda)$ är λ prediktionsfelets standardavvikelse.

När den nu minimala variansprediktorn har erhållits, kan minimum variansstyrlagen framtagas. Lägg märke till att

$$E \{ y^2(t+1) \} \geq \lambda^2 \qquad (2.6)$$

där likheten gäller om

$$u(t) = - B^{-1}(z^{-1}) \{ C(z^{-1}) - A(z^{-1}) \}\, y(t) \qquad (2.7)$$

Ekvation (2.7) är sålunda den minimum variansstyrlagen. Eftersom $b_o = 0$ och $b_1 \neq 0$ innehåller högra membrum termer $y(t)$, $y(t-1), \ldots$, och styrlagen är sålunda fysikaliskt realiserbar.

Under de speciella antagandena $b_1 \neq 0$, $b_o = 0$ erhålles minimum variansstyrlagen lätt ur modellen (2.1). För bestämning av styrlagar i allmännare fall se ref. (2) . När identifieringsproblemet väl är löst har vi även en lösning till minimum variansstyrproblemet, och påståendet i inledningen är bevisat.

I.3. Lösningen till problemet.

Problemet som är uppställt i del I.2. är ett statistiskt parameteruppskattningsproblem. Detta kommer att lösas genom maximum - likelihoodmetoden.

I.3.1. Likelihood - funktionen.

Sannolikhetsfunktionen för en stokastisk variabel ξ, som är normalfördelad med medelvärdet noll och standardavvikelsen λ { $\xi \in N(0,\lambda)$ } är

$$p_\xi(e(t)) = \frac{1}{\lambda\sqrt{2\pi}} e^{-\frac{1}{2}\{\frac{e(t)}{\lambda}\}^2} \qquad (1)$$

Likelihood - funktionen (L - funktionen) $L(\lambda)$ blir

$$L(\lambda) = p_\xi(e(1), \lambda) \cdot p_\xi(e(2), \lambda) \ldots p_\xi(e(N), \lambda) \qquad (2)$$

insättes (1) i (2) så erhålles

$$L(\lambda) = \frac{1}{\lambda^N (2\pi\lambda)^{N/2}} e^{-\frac{1}{2\lambda^2}\{e^2(1) + e^2(2) + \ldots + e^2(N)\}} \qquad (3)$$

Logaritmen av likelihoodfunktionen blir

$$\log L(\lambda) = -\frac{1}{2\lambda^2} \sum_{t=1} e^2(t) - N \log \lambda - \frac{1}{2} \cdot N \log 2\pi \qquad (4)$$

"Felen" e beräknas ur insignalen $\{ u(t) \}$ ~~och~~ utsignalen $\{ y(t) \}$ genom

$$C(z^{-1})\, e(t) = A(z^{-1})\, y(t) - B(z^{-1})\, u(t) \qquad (2.8)$$

som erhålles ur (2.1).

Sannolikhetsfunktionen är alltså en funktion av parametrarna a, b, c, λ och n begynnelsevillkor av (2.8).

I.3.2. Maximering av likelihoodfunktionen.

Eftersom logaritmen för likelihoodfunktionen når maximum samtidigt som likelihoodfunktionen själv, kan log L maximeras. Det går att maximera log L med avseende på parametrarna a, b och c separat. För att göra detta införes förlustfunktionen $V(\theta)$ definierad av

$$V(\theta) = \frac{1}{2} \sum_{t=1}^{N} e^2(t) \qquad (2.10)$$

där θ = kolonnvektorn (a, b, c). Maximering av log L är ekvivalent med minimering av V. När $\hat{\theta}$ har erhållits, så att $V(\hat{\theta})$ är minimum så erhålles ML estimatet av λ ur

$$\hat{\lambda}^2 = \frac{2}{N} V(\hat{\theta})$$

och alla parametrarna är uppskattade.

1.3.3. Numerisk algoritm.

För maximering av sannolikhetsfunktionen, dvs minimering av förlustfunktionen $V(\theta)$, användes följande Newton-Raphson algoritm

$$\theta^{k+1} = \theta^{k} - \{ V_{\theta\theta}(\theta^{k}) \}^{-1} V_{\theta}(\theta^{k}) \tag{2.12}$$

där V_θ är gradienten av $V(\theta)$ och $V_{\theta\theta}$ är matrisen av andraordningens partiella derivator till $V(\theta)$.

De partiella derivatorna av förlustfunktionen erhålles genom differentiering.

$$\frac{\partial V}{\partial \theta_i} = \sum_{t=1}^{N} e(t) \cdot \frac{\partial e(t)}{\partial \theta_i} \tag{2.13}$$

$$\frac{\partial^2 V}{\partial \theta_i \partial \theta_j} = \sum_{t=1}^{N} \frac{\partial e(t)}{\partial \theta_i} \cdot \frac{\partial e(t)}{\partial \theta_j} + \sum_{t=1}^{N} e(t) \cdot \frac{\partial^2 e(t)}{\partial \theta_i \partial \theta_j} \tag{2.14}$$

Derivatorna av $e(t)$ erhålles genom differentiering av differensekvationen (2.8)

$$C(z^{-1}) \frac{\partial e(t)}{\partial a_i} = z^{-i} y(t)$$

$$C(z^{-1}) \frac{\partial e(t)}{\partial b_i} = - z^{-i} u(t)$$

$$C(z^{-1}) \frac{\partial e(t)}{\partial c_i} = - z^{-i} e(t) \tag{2.15}$$

$$C(z^{-1}) \frac{\partial^2 e(t)}{\partial a_i \partial c_j} = - z^{-i-j+1} \frac{\partial e(t)}{\partial a_1}$$

$$C(z^{-1}) \frac{\partial^2 e(t)}{\partial b_i \partial c_j} = - z^{-i-j} \frac{\partial e(t)}{\partial b_o}$$

$$C(z^{-1}) \frac{\partial^2 e(t)}{\partial c_i \partial c_j} = -2z^{-i-j+1} \frac{\partial e(t)}{\partial c_1} \qquad (2.16)$$

Bevis av några av formlerna. Utgå från (2.8)

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

$$(1 + c_1 z^{-1} + \dots + c_n z^{-n})\, e(t) = (1 + a_1 z^{-1} + \dots + a_n z^{-n})$$

$$\cdot\, y(t) - (b_o + b_1 z^{-1} + \dots + b_n z^{-n})$$

$$u(t) \qquad (5)$$

Nu bildas ekvationerna (2.15) lätt.

Observera

$$C(z^{-1}) \frac{\partial e(t)}{\partial a_i} = z^{-i}\, y(t) = y(t-i)$$

speciellt gäller

$$C(z^{-1}) \frac{\partial e(t)}{\partial a_1} = y(t-1)$$

$$C(z^{-1}) \frac{\partial e(t)}{\partial a_{i+k}} = z^{-i}\, y(t-k)$$

Vi kan alltså skriva

$$C(z^{-1}) \frac{\partial e(t)}{\partial a_i} = C(z^{-1}) \frac{\partial d(t-i+1)}{\partial a_1} \quad \{ = y(t-i) \} \qquad (6)$$

medför

$$\frac{\partial e(t)}{\partial a_i} = \frac{\partial e(t-i+1)}{\partial a_1} \qquad i \leq t+1 \qquad (2.17)$$

Bilda nu

$$C(z^{-1})\ \frac{\partial^2 e(t)}{\partial a_i \partial c_j} = \frac{\partial}{\partial a_i}\ C(z^{-1})\ \frac{\partial e(t)}{\partial c_j} = \frac{\partial}{\partial a_i}\ \{ -z^{-j}\ e(t)\ \} =$$

$$= -z^{-j}\ \frac{\partial e(t)}{\partial a_i} = -z^{-j}\ \frac{\partial e(t - i + 1)}{\partial a_1} = -z^{-i-j+1}\ \frac{\partial e(t)}{\partial a_1}$$

På samma vis blir

$$C(z^{-1}) \cdot \frac{\partial^2 e(t)}{\partial b_i \partial c_j} = -z^{-i-j}\ \frac{\partial e(t)}{\partial b_j}$$

Härledning av sista ekvationen (2.16)

$$\frac{\partial}{\partial c_j}\ \{ C(z^{-1})\ \frac{\partial e(t)}{\partial c_i}\ \} = \frac{\partial}{\partial c_j}\ \{ -z^{-i}\ e(t)\ \}$$

$$z^{-j}\ \frac{\partial e(t)}{\partial c_i} + C(z^{-1}) \cdot \frac{\partial^2 e(t)}{\partial c_i \partial c_j} = -z^{-i}\ \frac{\partial e(t)}{\partial c_j}$$

$$C(z^{-1})\ \frac{\partial^2 e(t)}{\partial c_i \partial c_j} = -z^{-i}\ \frac{\partial e(t)}{\partial c_j} - z^{-j}\ \frac{\partial e(t)}{\partial c_i} =$$

$$= -z^{-i} \cdot \frac{\partial e(t - j + 1)}{\partial c_1} - z^{-j}\ \frac{\partial e(t - i + 1)}{\partial c_1} =$$

$$= -\ 2\ z^{-i-j+1}\ \frac{\partial e(t)}{\partial c_1}$$

Begynnelsevärdena för differensekvationerna (2.15) och (2.16) är noll. Man kan lägga märke till att de andra ordningens partiella derivator av $e(t)$, som inte innehåller koefficienterna c_i är alla identiskt noll.

Ekvationerna (2.8), (2.10), (2.15) och (2.16) ger omedelbart ett rekursionsschema för beräkning av förlustfunktionen $V(\theta)$ och dess partiella derivator. I (2.15) och (2.16) kan derivatorna med avseende på olika parametrar i samma grupp (a, b eller c) erhållas genom skift. För (2.15) ger (2.17) hur man skall skifta. Detta leder till avsevärd förenkling av beräkningsgången eftersom (2.15) och (2.16) endast behöver lösas för ett värde på i och j. T.ex. $i = j = 1$.

Genom att använda (2.17) är det möjligt att förenkla beräkningen av högra membrums första term i (2.14). Vi har t.ex.

$$\sum_{t=1}^{N} \frac{\partial e(t)}{\partial a_i} \cdot \frac{\partial e(t)}{\partial b_j} = \sum_{t=1}^{N} \frac{\partial e(t-i+1)}{\partial a_1} \cdot \frac{\partial e(t-j+1)}{\partial b_1}$$

$$= \sum_{t=1}^{N-j+1} \frac{\partial e(t+j-i)}{\partial a_1} \cdot \frac{\partial e(t)}{\partial b_1} \qquad \text{där } j \leq i \tag{2.18}$$

Liknande formler gäller även för de andra derivatorna.

Algoritmen för maximering av likelihoodfunktionen blir alltså

1. Sätt $\theta^k = \theta^o$ (=begynnelsevärdet på θ)
2. Lös $V_\theta(\theta^k)$ och $V_{\theta\theta}(\theta^k)$ genom användning av (2.13), (2.14), (2.15) och (2.16).
3. Beräkna θ^{k+1} ur (2.12) och börja om från 2.

I.3.4 Begynnelsevärde.

Algoritmen (2.12) kräver ett begynnelsevärde. Om parametern c är given, så är $V(\theta)$ en kvadratisk funktion av a och b och andra ordningens partiella derivator av e(t) är alla noll. Iterationen (2.12) konvergerar då i ett steg från varje begynnelsevärde på parametrarna a och b. Om vi speciellt sätter $c_i = 0$ för $i > 0$ erhålles i ett steg med de approximativa partiella andra-derivatorna minsta kvadratuppskattningen a^o och b^o av a och b. Begynnelsevärdet för itereringen (2.12) tages då

$$\theta^o \;=\; \text{kolonnvektorn } (a^o, b^o, 0)$$

I.3.5 Noggrannhet i bestämningarna.

Ur $V_{\theta\theta}^{-1}(\theta^k)$ bestämmes hur noggrant parametrarna är uppskattade. Man kan ur diagonalelementen för denna matris beräkna parameterspridningen $\{ \sigma(\theta^{k+1}) \}$. Följande relation gäller

$$\sigma\,(\theta^{k+1}) \;=\; \{ V_{\theta\theta}^{-1}\,(\theta^k) \}^{1/2}$$

där högersidan består av det aktuella diagonalelementet.

II. Programmering av identifieringsalgortimen.

II.1. Inledning.

Identifieringsalgoritmen programmeras i ALGOL. Programmet har testkörts på Siffermaskinen i Lund (SMIL) och är alltså vad gäller in- och utmatningsrutiner anpassat för denna maskin. De obetydliga avvikelserna från det allmänna ALGOL 60, som i övrigt måste göras på SMIL är angivna i ref. (4). Programmet maximerar likelihoodfunktionen och är ganska rättframt konstruerat. Genom att det har försetts med kommentarer i riklig mängd, hoppas vi att överskådligheten blivit god. Målsättningen har i första hand varit att få fram ett generellt och lättlöst program, därefter har snabbheten i beräkningarna beaktats. Hela beräkningsproceduren sker i maskinens inre kärnminne (KM) och programmet är stansat på 8-kanals hålremsa. Inläsning och utskrift av data sker med 5-kanals hålremsa.

II.2. Minnesutrymme och parametrar.

Med en andra ordningens systemmodell och 135 par av in- och utdata upptas ca. 3000 celler i KM.

II.2.1. Indata.

n :	Systemets ordning ($n \leq 2$).
N :	Antalet par av in- och utdata -1.
a,b och c :	Vektorer som innehåller koefficienterna för utsignalen Y insignalen U och "felen" E resp. som komponenter.
V :	Vektor med de loggade insignalerna som komponenter.
Y :	Dito utsignalerna.

II.2.2. Utdata

V :	Förlustfunktionen $V(\theta^k)$
V1 :	Gradientvektorn av förlustfunktionen $V_\theta(\theta^k)$
V2 :	Matrisen som innehåller förlustfunktionens partiella andraderivator $V_{\theta\theta}(\theta^k)$. Både den approximativa och den exakta andraderivatan utstansas.
$V2^{-1}$:	Inversen av den exakta andraderivatematrisen.
s0:	Determinanten av V2.
VV :	$\{ V_{\theta\theta}(\theta^k) \}^{-1} \cdot V_\theta(\theta^k)$
a,b och c:	Se indata.

II.2.3. Variabler.

n : Systemets ordning.

N : Antalet data -1.

a(0:n) : Polynomet A(z):s koefficienter

$$a(0) = 1$$
$$a(1) = a_1$$
$$\vdots$$
$$a(n) = a_n$$

b(0:n) : Polynomet B(z):s koefficienter

$$b(0) = b_o$$
$$\vdots$$
$$b(n) = b_n$$

c (0:n) : Polynomet C(z):s koefficienter

$c(0) = 0$ {av programmeringstekniska skäl ($c_o = 1$ i polynomet) }

$c(1) = c_1$

$\vdots$

$c(n) = c_n$

E (-n:N) : Felen e(t)

$E(-n) = 0$

$E(-n+1) = 0$

$\vdots$

$E(-1) = 0$

$E(0) = e(t - N) = e(0)$

$E(1) = e(t - N + 1) = e(1)$

$\vdots$

$E(N) = e(t) = e(N)$

U (-n:N) : Insignalerna u(t)

$U(-n) = 0$

$U(-n+1) = 0$

$\vdots$

$U(-1) = 0$

$U(0) = u(t - N) \quad = u(0)$

$U(1) = u(t - N + 1) \quad = u(1)$

$\vdots$

$U(N) = u(t) \quad = u(N)$

Y (-n:N) : Utsignalerna y(t)

$Y(-n) = 0$

$Y(-n+1) = 0$

$\vdots$

$Y(-1) = 0$

$Y(0) = y(t - N) = y(0)$

$Y(1) = y(t - N + 1) = y(1)$

$\vdots$

$Y(N) = y(t) \quad = y(N)$

V : Förlustfunktionen $V = \frac{1}{2} \sum_{o}^{N} e^2(t)$

$$EA1(-n:N) = \frac{\partial e(t)}{\partial a_1}$$

$EA1(-n) = 0$

$EA1(-n+1) = 0$

$\vdots$

$EA1(-1) = 0$

$$EA1(0) = \frac{\partial e(0)}{\partial a_1} = 0$$

$$EA1(1) = \frac{\partial e(1)}{\partial a_1} = y(0)$$

$$EA1(2) = \frac{\partial e(2)}{\partial a_1} = -c_1 \frac{\partial e(1)}{\partial a_1} + y(1) = -c_1\ EA1(1) + y(1)$$

$$EA1(3) = \frac{\partial e(3)}{\partial a_1} = -c_1 \frac{\partial e(2)}{\partial a_1} - c_2 \frac{\partial e(1)}{\partial a_1} + y(2) = c_1 EA1(2) -$$

$$-c_2\ EA1(1) + y(2)$$

$\vdots$

$$EA1(N) = \frac{\partial e(N)}{\partial a_1} = -c_1 \frac{\partial e(N-1)}{\partial a_1} - c_2 \frac{\partial e(N-2)}{\partial a_1} - \ldots - c_n \frac{\partial e(N-n)}{\partial a_1} +$$

$$+ y(N-1) = -c_1\, EA1(N-1) - c_2\, EA1(N-2) - \ldots - c_n\, EA1(N-n) +$$

$$+ y(N-1)$$

EB/(-n:N) : $\frac{\partial e(t)}{\partial b_o}$ Komponenterna erhålles på samma sätt som EA1:s.

EC1(-n:N) : $\frac{\partial e(t)}{\partial c_1}$ Komponenterna erhålles på samma sätt som EA1:s.

V1(1:3 x n + 1) : Förstaderivatan av V.

$$V1\ (1) = \frac{\partial V}{\partial a_1} = \sum_{t=1}^{N} e(t) \cdot \frac{\partial e(t)}{\partial a_1}$$

$$V1\ (2) = \frac{\partial V}{\partial a_2} = \sum_{t=1}^{N} e(t) \cdot \frac{\partial e(t)}{\partial a_2} = \sum_{t=1}^{N} e(t) \cdot \frac{\partial e(t-1)}{\partial a_1}$$

$\vdots$

$$V1\ (n) = \frac{\partial V}{\partial a_n} = \sum_{t=1}^{N} e(t) \cdot \frac{\partial e(t)}{\partial a_n} = \sum_{t=1}^{N} e(t) \cdot \frac{\partial e(t-n+1)}{\partial a_1}$$

$$V1\ (n+1) = \frac{\partial V}{\partial b_o} = \sum_{t=0}^{N} e(t) \cdot \frac{\partial e(t)}{\partial b_o}$$

$$V1\ (n+2) = \frac{\partial V}{\partial b_1} = \sum_{t=0}^{N} e(t) \cdot \frac{\partial e(t)}{\partial b_1} = \sum_{t=0}^{N} e(t) \cdot \frac{\partial e(t-1)}{\partial b_o}$$

$\vdots$

$$V1(2 \times n + 1) = \frac{\partial V}{\partial b_n} = \sum_{t=0}^{N} e(t) \cdot \frac{\partial e(t)}{\partial b_n} = \sum_{t=0}^{N} e(t) \cdot \frac{\partial e(t-n+1)}{\partial b_o}$$

$$V1(2 \times n + 2) = \frac{\partial V}{\partial c_1} = \sum_{t=1}^{N} e(t) \cdot \frac{\partial e(t)}{\partial c_1}$$

$$V1(2 \times n + 3) = \frac{\partial V}{\partial c_2} = \sum_{t=1}^{N} e(t) \cdot \frac{\partial e(t)}{\partial c_2} = \sum_{t=1}^{N} e(t) \cdot \frac{\partial e(t-1)}{\partial c_1}$$

$$\vdots$$

$$V1(3 \times n + 1) = \frac{\partial V}{\partial c_n} = \sum_{t=1}^{N} e(t) \cdot \frac{\partial e(t)}{\partial c_n} = \sum_{t=1}^{N} e(t) \cdot \frac{\partial e(t-n+1)}{\partial c_n}$$

V2(1:3 x n + 1, 1:2 x (3 x n + 1)): Andra derivatan av V. Matrisen är symmetrisk som framgår av schemat nedan, som visar komponenternas placering. Matrisen innehåller dubbelt så många kolonner som rader. Detta användes vid inversionen.

$$
\nabla^2 = \begin{array}{cccccccccc}
\frac{\partial^2 V}{\partial a_1^2} & \frac{\partial^2 V}{\partial a_1 \partial a_2} & \cdots & \frac{\partial^2 V}{\partial a_1 \partial a_n} & \frac{\partial^2 V}{\partial a_1 \partial b_o} & \cdots & \frac{\partial^2 V}{\partial a_1 \partial b_n} & \frac{\partial^2 V}{\partial a_1 \partial c_1} & \cdots & \frac{\partial^2 V}{\partial a_1 \partial c_n} \\
\frac{\partial^2 V}{\partial a_1 \partial a_2} & \frac{\partial^2 V}{\partial a_2^2} & \cdots & \frac{\partial^2 V}{\partial a_2 \partial a_n} & \frac{\partial^2 V}{\partial a_2 \partial b_o} & \cdots & \frac{\partial^2 V}{\partial a_2 \partial b_n} & \frac{\partial^2 V}{\partial a_2 \partial c_1} & \cdots & \frac{\partial^2 V}{\partial a_2 \partial c_n} \\
\vdots & & & & & & & & & \\
\frac{\partial^2 V}{\partial a_1 \partial a_n} & \frac{\partial^2 V}{\partial a_2 \partial a_n} & \cdots & \frac{\partial^2 V}{\partial a_n^2} & \frac{\partial^2 V}{\partial a_n \partial b_o} & \cdots & \frac{\partial^2 V}{\partial a_n \partial b_n} & \frac{\partial^2 V}{\partial a_n \partial c_1} & \cdots & \frac{\partial^2 V}{\partial a_n \partial c_n} \\
\frac{\partial^2 V}{\partial a_1 \partial b_o} & \frac{\partial^2 V}{\partial a_2 \partial b_o} & \cdots & \frac{\partial^2 V}{\partial a_n \partial b_o} & \frac{\partial^2 V}{\partial b_o^2} & \cdots & \frac{\partial^2 V}{\partial b_o \partial b_n} & \frac{\partial^2 V}{\partial b_o \partial c_1} & \cdots & \frac{\partial^2 V}{\partial b_o \partial c_n} \\
\vdots & & & & & & & & & \\
\frac{\partial^2 V}{\partial a_1 \partial b_n} & \frac{\partial^2 V}{\partial a_2 \partial b_n} & \cdots & \frac{\partial^2 V}{\partial a_n \partial b_n} & \frac{\partial^2 V}{\partial b_o \partial b_n} & \cdots & \frac{\partial^2 V}{\partial b_n^2} & \frac{\partial^2 V}{\partial b_n \partial c_1} & \cdots & \frac{\partial^2 V}{\partial b_n \partial c_n} \\
\frac{\partial^2 V}{\partial a_1 \partial c_1} & \frac{\partial^2 V}{\partial a_2 \partial c_1} & \cdots & \frac{\partial^2 V}{\partial a_n \partial c_1} & \frac{\partial^2 V}{\partial b_o \partial c_1} & \cdots & \frac{\partial^2 V}{\partial b_n \partial c_1} & \frac{\partial^2 V}{\partial c_1^2} & \cdots & \frac{\partial^2 V}{\partial c_1 \partial c_n} \\
\vdots & & & & & & & & & \\
\frac{\partial^2 V}{\partial a_1 \partial c_n} & \frac{\partial^2 V}{\partial a_2 \partial c_n} & \cdots & \frac{\partial^2 V}{\partial a_n \partial c_n} & \frac{\partial^2 V}{\partial b_o \partial c_n} & \cdots & \frac{\partial^2 V}{\partial b_n \partial c_n} & \frac{\partial^2 V}{\partial c_1 \partial c_n} & \cdots & \frac{\partial^2 V}{\partial c_n^2}
\end{array}
$$

För två godtyckliga parametrar θ_i och θ_j är approximativa andraderivateelementet

$$V2\ \{s,t\} = \sum_{t=0}^{N} \frac{\partial e(t)}{\partial \theta_i} \cdot \frac{\partial e(t)}{\partial \theta_j}$$

Det exakta andraderivateelementet blir

$$V2\ \{s,t\} = \sum_{t=0}^{N} \frac{\partial e(t)}{\partial \theta_i} \cdot \frac{\partial e(t)}{\partial \theta_j} + \sum_{t=0}^{N} e(t) \cdot \frac{\partial^2 e(t)}{\partial \theta_i \partial \theta_j}$$

där sista termen beräknas enligt nedan.

$$EAC\{-n:N\} = \frac{\partial^2 e(t)}{\partial a_1 \partial c_1}$$

$$EAC\{-n\} = 0$$

$$EAC\{-n+1\} = 0$$

$$\vdots$$

$$EAC\{-1\} = 0$$

$$EAC\{0\} = \frac{\partial^2 e(0)}{\partial a_1 \partial c_1} = 0$$

$$EAC\{1\} = \frac{\partial^2 e(1)}{\partial a_1 \partial c_1} = 0$$

$$EAC\{2\} = \frac{\partial^2 e(2)}{\partial a_1 \partial c_1} = - EA1\{1\}$$

$$EAC\{3\} = \frac{\partial^2 e(3)}{\partial a_1 \partial c_1} = - EA1\{2\} - c_1\, EAC\{2\}$$

$$EAC\{4\} = \frac{\partial^2 e(4)}{\partial a_1 \partial c_1} = - EA1\{3\} - c_1\, EAC\{3\} - c_2\, EAC\{2\}$$

$$\vdots$$

$$EAC\{N\} = \frac{\partial^2 e(N)}{\partial a_1 \partial c_1} = - EA1\{N-1\} - c_1\, EAC\{N-1\} - c_2\, EAC\{N-2\} - \ldots - c_n\, EAC\{N-n\}$$

$EBC\{-n:N\} = \dfrac{\partial^2 e(t)}{\partial b_o \partial c_1}$ Komponenterna erhålles som för EAC

$ECC\{-n:N\} = \dfrac{\partial^2 e(t)}{\partial c_1^2}$ Komponenterna erhålles som för EAC

I högra delen av V2 insättes enhetsmatrisen.

s0 : Determinanten av V2.

Efter matrisinversionen finns inversen av V2 i matrisens högra del

$VV\{1:3 \times n + 1\} = \{V2\}^{-1} \cdot V1$

II.3 Genomgång av programmet.

n läses in.

N läses in.

Plats reserveras i KM för vektorerna a, b, c, E, U och Y. E(-n), E(-n+1), ... , E(0) p.s.s. för Y och U användes för att initialsvårigheter skall kunna bemästras i repetitionssatsen vid bildande av vektorn E.

1, b(0), 0, a(1), b(1), c(1), ..., a(n), b(n), c(n) inläses i denna ordning.

U(0), Y(0), U(1), Y(1), ..., U(N), Y(N) läses in i den här ordningen.

L2 : Läge som programmet går till vid upprepning av iterationen.

Vektorn E:s komponenter bildas.

Block som beräknar förlustfunktionen $V = \frac{1}{2} \sum_{t=0}^{N} E^2(t)$

Vektorerna EA1, EB0 och EC1 lägges ut i KM.

Proceduren EV användes för att beräkna EA1, EB0 och EC1, samt även senare i programmet då EAC, EBC och ECC bildas.

Gradientvektorn V1 placeras i KM.

Proceduren Vv ger gradientvektorn V1.

Matrisen V2 som innehåller de partiella andraderivatorna deklareras. Matrisen har dubbelt så många kolonner som rader. Detta kommer att behövas vid matrisinversionen. Eftersom matrisen V2 är symmetrisk arbetar vi bara med den övre triangulärmatrisen.

Proceduren Vvv beräknar först den approximativa andraderivatematrisen.

Efter deklarering av EAC, EBC och ECC beräknas dessa korrektionstermer genom att proceduren EV åter användes, och därefter får proceduren Vvv ge den exakta andraderivatematrisen.

Hela V2 bildas genom att elementen i undre triangulärmatrisen sätts lika med elementen i den övre.

Funktionsproceduren Det Gauss införes, och anropas vid inversionen av V2.

Enhetsmatrisen införes i V2:s högra kolonndel.

Det Gauss anropas och funktionsproceduren inverterar V2 samt skriver ut determinanten av V2 omedelbart följd av asterisk. Är V2 singulär sker uthopp ur funktionsproceduren till läget L1 och asterisk erhålles i vänstermarginalen på utskriften.

Om determinanten av V2 $\neq 0$ beräknas $\{V2\}^{-1}$ och utskrives .

Vektorn VV beräknas och utskrives.

Parametrarna erhåller sina nya värden, hopp sker till L2 och nästa iteration påbörjas.

II.4 Inversionsprogrammet.

Funktionsproceduren Det Gauss använder Gauss eliminationsmetod för att triangulera koefficientmatrisen. Först sker en skalning av matrisen, genom att dess radvektorer skalas med heltal 2^p där p väljes så att den euklidiska normen $|x|$ av varje rad är

$$\frac{1}{2} \leq |x| \leq 1$$

{ Den euklidiska normen av x är

$$|x| = \left(\sum_i |x_i|^2 \right)^{1/2} \quad \}$$

Skalningen göres i basen två eftersom vi då kan använda flytande räkning utan att få några avrundningsfel.

Härefter reduceras koefficientmatrisen till triangulär form genom Gauss' elimination, där i varje steg det numeriskt största elementet i den återstående matrisen väljes som pivotelement. Om något av dessa pivotelement har ett absolut värde $\leq 10^{-8}$ går Det Gauss till läget L1. Matrisen är då nästan singulär. Proceduren är hämtad ur ref. (8).

II.5 Testning och diskussion.

Uppbyggnaden av programmet har skett stegvis genom att beräkningar adderats efter hand. Varje beräkningssteg har testats efter det att det införts i huvudprogrammet. För matrisinversionsproceduren Det Gauss utfördes en test även innan proceduren fogades in i identifieringsprogrammet.

Samtliga operationer utfördes av maskinen i KM. Där finns också alla vektor- och matrisfält placerade. Programmet är beroende av antalet data. På SMIL, som har ett inre kärnminnesutrymme av ca 3250 tillgängliga celler, kan ung 150 par av in- och utdata behandlas med en systemmodell av andra ordningen. Med första ordningens system erhålles ytterligare plats för data men vi kan inte köra långa serier.

För att lösa detta problem får ett program konstrueras som är oberoende av antalet data. Av ekvationerna ses att vi enbart behöver arbeta med vissa aktuella data. Genom att ta fram en lämplig tillståndsvektor, som ênbart kräver nödvändiga data, blir programmet beroende av systemets ordning, men ej av antalet data.

Med en maskinkodsprocedur inpassad i ALGOL-programmet kan in- och utdatavektorerna överföras från inläsningsremsan direkt till SMIL:s yttre kärnminne (ECS). Härigenom kan plats vinnas i KM för operationer. För denna operation användes t.ex. den vid SMIL befintliga proceduren LECS. När vi behöver in- och utdata inläses dessa sedan till KM från ECS genom funktions-proceduren ECS. Detaljerade regler för användning av procedurer i maskinkod i SMIL - ALGOL - program ges i ref. (9). Användning av SMIL:s yttre enheter medför att utrymme frigöres för operationer i inre kärnminnet. Emellertid kräver överföringen från de yttre enheterna till inre kärnminnet viss tid, varigenom programmen blir långsammare än om endast KM utnyttjats.

Bruk av SMIL:s yttre enheter bör kunna ge ett program som gör KM oberoende av datamängden. Häri ligger programmets omedelbaraste utvecklingsmöjligheter.

Programtesten utfördes först med 10 godtyckliga data för ett andra ordningens system med utskrift av varje beräkningsled. Resultat i bil. 1. Vid minimering av förlustfunktionen visar sig systemet vara instabilt, ty $C(z^{-1})$:s rötter kommer utanför enhetscirkeln. Med 100 datapar alstrade av systemet

$$y(t) - 1.5y(t-1) + 0.7y(t-2) = 1.0u(t-1) + 0.5u(t-2) + e(t) - 1.0e(t-1) + 0.2e(t-2)$$

kördes identifieringsprogrammet då parameterindata utgjordes av de exakta systemkoefficienterna enligt ovan. Resultat i bil. 2.

Där finns även ett program, som bestämmer utsignalen enligt en-stegs-prediktion och deterministisk modell. Insignal, utsignal, en-stegs-prediktionsmodell, deterministisk modell samt kovariansfunktion är uppritade i diagram för exemplet ovan.

(1) 1/2-66

```
comment Numerisk identifiering av en process, som kan beskrivas som ett
        enkelt lineärt system.
          n: Systemets ordning.
          N: Antalet data -1.
      a,b,c: Vektorer, som innehåller koefficienterna för utsignalen Y,
             insignalen U och avvikelserna E resp. som komponenter.
          V: Förlustfunktionen.
EA1,EB0,EC1: Vektorer, som såsom komponenter innehåller de/da, de/db,
             de/dc resp.
         V1: Gradienten av förlustfunktionen V.
         V2: En matris, som innehåller förlustfunktionen V:s partiella and-
             raderivator.
EAC,EBC,ECC: Vektorer med d/da(de/dc), d/db(de/dc) och d/dc(de/dc) resp.
             som komponenter.
         s0: Determinanten av V2.
         VV: Inversen av V2×V1.
     Indata: n,N, a[0], b[0], c[0],... a[n], b[n], c[n],U[0], Y[0],... U[N],
             Y[N].
         L1: Läge, som programmet går till om V2 är singulär.
         L2: Läge, som programmet går till vid ny iteration.;

begin integer n,N;
  n:=read;
  N:=read;
  begin integer i,k;
  real array a,b,c[0:2],E,U,Y[-2:130];
    for k:=-n step 1 until 0 do
      E[k]:=U[k]:=Y[k]:=0;
    for k:=0 step 1 until n do
      begin
        a[k]:=read; b[k]:=read; c[k]:=read
      end;
    for i:=0 step 1 until N do
      begin
        U[i]:=read; Y[i]:=read;
        print(3,0,i+1); punch(0); punch(0);
        print(0,6,U[i]); punch(0); punch(0);
        print(Y[i]); punch(1)
      end inläsning och utskrift av in- och utdata;
    punch(15);
L2: for i:=0 step 1 until N do
      begin
        E[i]:=0;
        for k:=i step -1 until i-n do
          E[i]:=E[i]-c[i-k]×E[k]+a[i-k]×Y[k]-b[i-k]×U[k];
      end vektorn E;
    begin real V;
      V:=0;
      for i:=0 step 1 until N do
        V:=V+E[i]×E[i];
      V:=V/2;
      print(0,8,V);punch(1)
    end förlustfunktionen V;
    punch(1);;
```

(2)


```
begin real array EA1,EBO,EC1[-2:13];
   procedure EV(Ev,z,m,mO);
   value m,mO;
   integer m,mO;
   real array Ev,z;
     begin
       for k:=1 step 1 until n do
         Ev[i]:=Ev[i]-c[k]×Ev[i-k];
       Ev[i]:=Ev[i]+(if mO=1 then z[m] else if mO=0 then -z[m]
       else -2×z[m])
     end;
   for i :=-n step 1 until N do
     EA1[i]:=EBO[i]:=EC1[i]:=0;
   for i:=0 step 1 until N do
     begin
       EV(EA1,Y,i-1,1);
       EV(EBO,U,i,0);
       EV(EC1,E,i-1,0)
     end de/da, de/db, de/dc;
  begin integer j,s,t;
  real array V1[1:7];
    procedure Vv(Ev,p,q);
    value p,q;
    integer p,q;
    real array Ev;
      for k:=1 step 1 until q do
        for i:=0 step 1 until N-k+1 do
          begin
            s:=i+k-1; t:=k+p;
            V1[t]:=V1[t]+E[s]×Ev[i]
          end;
   j:=3×n+1;
   for k:=1 step 1 until j do
     V1[k]:=0;
   Vv(EA1,0,n);
   Vv(EBO,n,n+1);
   Vv(EC1,2×n+1,n);
   for k:=1 step 1 until j do
     begin
       print(V1[k]);punch(1)
     end förstaderivatan V1;
  punch(1);;
```

```
begin integer m,o,v;
real array V2[1:7,1:1'];
  procedure Vvv(Eu,Ev,j0,j,o,p,q,r);
  value j0,j,o,p,q,r;
  integer j0,j,o,p,q,r;
  real array Eu,Ev;
    for i:=1 step 1 until q do
      begin
        if j‡1 then j:=i else j:=1;
        for k:=j step 1 until r do
          for m:=0 step 1 until if j0=0 then N-k+1 else N-k-i+2 do
            begin
              s:=i+o; t:=k+p;
              if j0=0 then v:=m+k-i else v:=m+k+i-2;
              V2[s,t]:=V2[s,t]+Eu[v]×Ev[m]
            end
      end;
    for i:=1 step 1 until j do
      for k:=i step 1 until j do
        V2[i,k]:=0;
    Vvv(EA1,EA1,0,0,0,0,n,n);
    Vvv(EBO,EBO,0,0,n,n,n+1,n+1);
    Vvv(EC1,EC1,0,0,2×n+1,2×n+1,n,n);
    Vvv(EA1,EBO,0,1,0,n,n,n+1);
    Vvv(EA1,EC1,0,1,0,2×n+1,n,n);
    Vvv(EBO,EC1,0,1,n,2×n+1,n+1,n);
    for i:=1 step 1 until j do
      begin
        for k:=i step 1 until j do
          begin
            print(0,',V2[i,k]); punch(0)
          end;
        punch(1)
      end approximativa andraderivatan;
   punch(1);
   begin real array EAC,EBC,ECC[-2:13k];
     for i:=-n step 1 until N do
       EAC[i]:=EBC[i]:=ECC[i]:=0;
     for i:=1 step 1 until N do
       begin
         EV(EAC,EA1,i-1,0);
         EV(EBC,EBO,i-1,0);
         EV(ECC,EC1,i-1,2)
       end d/da(de/dc),d/db(de/dc),d/dc(d/dc);
     Vvv(E,EAC,1,1,0,2×n+1,n,n);
     Vvv(E,EBC,1,1,n,2×n+1,n+1,n);
     Vvv(E,ECC,1,0,2×n+1,2×n+1,n,n);
     for i:=1 step 1 until j do
       begin
         for k:=i step 1 until j do
           begin
             print(0,',V2[i,k]);punch(0)
           end;
         punch(1)
       end
  end exakta andraderivatan;
  punch(1);punch(15);;
```

```
for i:=1 step 1until j do
  for k:=i step 1 until j do
    V2[k,i]:=V2[i,k];
begin real s0;
  real procedure Det Gauss(n,m,A,exit);
  value n,m;
  integer  n,m;
  array A;
  label exit;
  comment Det Gauss löser m lineära ekvationssystem med n obekanta och
          beräknar determinanten av koefficientmatrisen. Här användes
          proceduren Det Gauss för matrisinversion.
       n: Matrisens ordning.
       m: Vid matrisinversion är m=n.
       A: Har gränserna [1:n,1:n+m] och måste från början ha koefficient-
          matrisen i A[1:n,1:n] och enhetsmatrisen i A[1:n,n+1:2×n]. Den
          inverterade matrisen lagras  när programmet genomlöpts i
          A[1:n,n+1:2×n].
    exit: Läge, som Det Gauss går till, när koefficientmatrisen är sin-
          gulär;
    begin integer i,j,k,i0,j0;
    real factor, max, detfac, twofac;
    integer array permute[1:7];
      m:=n+m;
      detfac:=1;
      for i:= 1 step 1 until n do
        begin
          max:=0;
          for j:=1 step 1 until n do
            max:=max+A[i,j]↑2;
          if max> 1 ∨ max<.25 then
            begin
              twofac:=2↑(-entier(ln(max)/1.3863+1));
              for j:=1 step 1 until m do
                A[i,j]:=A[i,j]×twofac;
              detfac:=detfac/twofac
            end
        end equilibration;
      for k:=1 step 1 until n do
        begin
          max:=0;
          for i:=k step 1 until n do
            for j:=k step 1 until n do
              begin
                factor:=abs(A[i,j]);
                if max ≤ factor then
                  begin
                    max:=factor;
                    i0:=i;
                    j0:=j
                  end
              end searching for pivotal element;;
```

```
          if max ≤ ₁₀-8 then go to exit;
          max:=A[i0,j0];
          detfac:=detfac×max;
          if i0≠k then
            begin
              detfac:=-detfac;
              for j:=k step 1 until m do
                begin
                  factor:=A[k,j];
                  A[k,j]:=A[i0,j];
                  A[i0,j]:=factor
                end
            end interchange of rows;
          permute[k]:=j0;
          if j0≠k then
            begin
              detfac:=-detfac;
              for i:=1 step 1 until n do
                begin
                  factor:=A[i,k];
                  A[i,k]:=A[i,j0];
                  A[i,j0]:=factor
                end
            end interchange of columns;
          for i:=k+1 step 1 until n do
            begin
              factor:=A[i,k]/max;
              if factor ≠0 then
                for j:=k+1 step 1 until m do
                  A[i,j]:=A[i,j]-A[k,j]×factor
            end reduction
        end for k;
      for k:=n+1 step 1 until m do
        for i:=n step -1 until 1 do
          begin
            factor:=A[i,k];
            for j:=i+1 step 1 until n do
              factor:=factor-A[i,j]×A[j,k];
            A[i,k]:=factor/A[i,i]
          end solving;
      if m≠n then
        begin
          for i:=n-1 step -1 until 1 do
            begin
              i0:=permute[i];
              if i0≠i then
                for k:=n+1 step 1 until m do
                  begin
                    factor:=A[i,k];
                    A[i,k]:=A[i0,k];
                    A[i0,k]:=factor
                  end
            end
        end interchange of solutions;
     Det Gauss:=detfac
   end Det Gauss;;
```

```
                  for i:=1 step 1 until j do
                    for k:=j+1 step 1 until 2×j do
                      if k=j+1 then V2[i,k]:=1 else V2[i,k]:=0;
                  s0:= Det Gauss(j,j,V2,L1);
                  for i:=1 step 1 until j do
                    begin
                      for k:=j+1 step 1 until 2×j do
                        begin
                          print(V2[i,k]); punch(0)
                        end;
                      punch(1)
                    end inversen av andraderivatematrisen;
                  punch(1);
                  print(s0);
              L1: punch(8);
                  punch(1); punch(1)
                end determinanten av andraderivatematrisen;
                begin real array VV[1:7];
                  for i:=1 step 1 until j do
                    VV[i]:=0;
                  for i:=1 step 1 until j do
                    for k:=1 step 1 until j do
                      VV[i]:=VV[i]+V2[i,j+k]×V1[k];
                  for i:=1 step 1 until j do
                    begin
                      print (0,8,VV[i]);punch(1)
                    end;
                  punch(1);
                  for i:=1 step 1 until n do
                    begin
                      a[i]:=a[i] -VV[i];
                      print(a[i]);punch(3);
                      c[i]:=c[i]-VV[2×n+1+i];
                      print(c[i]);punch(1)
                    end;
                  punch(1);
                  for i:=0 step 1 until n do
                    begin
                      b[i]:=b[i]-VV[n+1+i];
                      print(b[i]);punch(1)
                    end
                end VV
            end V2
            end V1
         end EA1,EB0,EC1;
         punch(15);
       go to L2
       end a,b,c,E,U,Y
     end n,N
```

II.7 Referenser.

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

(2) Åström, K.-J. Notes on the regulation problem.

(3) Åström, K.-J., Bohlin, T., Wensmark, S. Automatic construction of linear stochastic dynamic models for stationary industrial processes with random disturbances using operating records.

(4) Ragazzini, J.R., Franklin, G.F. Sampled-Data Control Systems.

(5) Cramér, H. Mathematical Methods of Statistics.

(6) Fröberg, C.-E. Lärobok i numerisk analys.

(7) Ekman, T., Fröberg, C.-E. Lärobok i algol.

(8) Nordisk tidskrift för informationsbehandling Bind. 5. Häfte nr 1. 1965.

(9) Ekman, T. Större PM ang. SMIL - ALGOL

Programtest for 10 st godtyckliga in- och utdata.

Parametervärden på indatana:

$a = (1, 1, 0)$

$b = (1, 0, 2)$

$c = (0, 2, 1)$

$$C(z^{-1}) = 1 + 2z^{-1} + z^{-2}$$

$$z^2 = -2z - 1$$

$$z = -1 \pm \sqrt{1-1}$$

$$z = -1 \qquad |z| = 1$$

$C(z^{-1})$:s nollställen ligger på enhetscirkeln.

Efter första iterationen.

$a = (1, 0.55, -0.13)$

$b = (0.98, -0.32, 0.63)$

$c = (0, 1.60, 0.50)$

$$C(z^{-1}) = 1 + 1.6z^{-1} + 0.5z^{-2}$$

$$z^2 = -1.6z - 0.5$$

$$z = -0.8 \pm \sqrt{0.64 - 0.5}$$

$$z = -0.8 \pm 0.37$$

$$z_1 = -1.17$$

$$z_2 = -0.43$$

$C(z^{-1})$ har ett nollställe innanför och ett utanför enhetscirkeln.

Efter andra iterationen.

$a = (1, 0.42, -0.27)$

$b = (0.66, 0.09, 1.27)$

$c = (0, 0.99, -0.25)$

$C(z^{-1}) = 1 + 0.99z^{-1} - 0.25z^{-2}$

$z^2 = -0.99z + 0.25$

$$z = -\frac{0.99}{2} \pm \sqrt{\frac{0.99^2 + 1.0}{4}} = -0.49 \pm \sqrt{0.7}$$

$z_1 = -1.19$

$z_2 = 0.21$

$C(z^{-1})$ har ett nollställe innanför och ett utanför enhetscirkeln.

Systemet är instabilt.

bil. 1.

Indata för testexempel med godtyckliga data.

	insignal	utsignal
1	5.000000 -1	1.000000 +0
2	-5.000000 -1	-1.000000 +0
3	-1.000000 +0	5.000000 -1
4	-5.000000 -1	-5.000000 -1
5	5.000000 -1	-1.000000 +0
6	1.000000 +0	-5.000000 -1
7	5.000000 -1	5.000000 -1
8	1.000000 +0	1.000000 +0
9	1.000000 +0	5.000000 -1
10	-1.000000 +0	1.000000 +0

25/7

Resultat av testexempel med godtyckliga indata.

1. Iterationen.

V 4.75000004 +0

	-9.27500007 +1
	7.66250006 +1
	-2.28750002 +1
V1	1.98750001 +1
	-1.66250001 +1
	1.08937501 +2
	-8.16875007 +1

V2 approx.

1.3920 +3	-1.1570 +3	3.3300 +2	-2.8250 +2	2.5500 +2	-1.6413 +3	1.2280 +3
9.7175 +2	-2.7125 +2	2.3050 +2	-2.1075 +2	1.3836 +3	-1.0416 +3	
8.5250 +1	-7.0000 +1	6.1000 +1	-3.8450 +2	2.8388 +2		
6.0250 +1	-5.2500 +1	3.2175 +2	-2.3825 +2			
4.8000 +1	-2.9475 +2	2.1938 +2				
1.9956 +3	-1.5053 +3					
1.1400 +3						

V2 exakt

1.3920 +3	-1.1570 +3	3.3300 +2	-2.8250 +2	2.5500 +2	-2.3585 +3	1.6833 +3
9.7175 +2	-2.7125 +2	2.3050 +2	-2.1075 +2	1.8389 +3	-1.3115 +3	
8.5250 +1	-7.0000 +1	6.1000 +1	-6.8188 +2	4.9275 +2		
6.0250 +1	-5.2500 +1	5.3063 +2	-3.7850 +2			
4.8000 +1	-4.3500 +2	3.0763 +2				
3.2136 +3	-2.2378 +3					
1.5504 +3						

1. Iterationen forts.

	a_1	a_2	b_0	b_1	b_2	c_1	c_2
	6.3639 -1	4.7731 -1	-3.1548 -1	4.3079 -1	-4.2822 -1	-2.9783 -2	-3.9744 -2
	4.7731 -1	3.7094 -1	-2.2220 -1	3.5774 -1	-2.6660 -1	-4.2471 -2	-5.4875 -2
$V2^{-1}$	-3.1548 -1	-2.2220 -1	2.9675 -1	6.3343 -2	3.6802 -1	3.9192 -2	5.9244 -2
	4.3079 -1	3.5774 -1	6.3343 -2	9.8359 -1	2.8437 -1	1.4919 -2	2.0010 -2
	-4.2822 -1	-2.6660 -1	3.6802 -1	2.8437 -1	8.9239 -1	-9.9117 -2	-1.2827 -1
	-2.9783 -2	-4.2471 -2	3.9192 -2	1.4919 -2	-9.9117 -2	-4.8773 -2	-6.3136 -2
	-3.9744 -2	-5.4875 -2	5.9244 -2	2.0010 -2	-1.2827 -1	-6.3136 -2	-8.2244 -2

s0 1.3987 +9*

VV
4.48649820 -1
6.33844506 -1
1.67307071 -2
8.18508335 -1
1.36740335 +0
4.00057789 -1
4.97010693 -1

a
5.51350180 -1
-1.33844506 -1

c
1.59994221 +0
5.02989307 -1

b
9.83269293 -1
-3.18508335 -1
6.32596649 -1

2. Iterationen.

V   3.42475491 +0

V1

-3.78444235 +1
3.02988458 +1
-8.88817985 +0
8.09100013 +0
-7.91419697 +0
4.11940080 +1
-3.03043366 +1

$V2_{approx}$

5.0305 +2	-4.1822 +2	1.2846 +2	-9.8835 +1	8.5988 +1	-6.1333 +2	4.5934 +2
3.5224 +2	-1.0533 +2	8.0825 +1	-7.1340 +1	5.1664 +2	-3.8949 +2	
3.6159 +1	-2.5659 +1	2.1271 +1	-1.5752 +2	1.1685 +2		
2.1114 +1	-1.6975 +1	1.1640 +2	-8.6818 +1			
1.6101 +1	-1.0116 +2	7.5589 +1				
7.7541 +2	-5.8476 +2					
4.4317 +2						

$V2_{exakt}$

5.0305 +2	-4.1822 +2	1.2846 +2	-9.8835 +1	8.5988 +1	-8.0004 +2	5.8029 +2
3.5224 +2	-1.0533 +2	8.0825 +1	-7.1340 +1	6.3758 +2	-4.6324 +2	
3.6159 +1	-2.5659 +1	2.1271 +1	-2.3396 +2	1.7100 +2		
2.1114 +1	-1.6975 +1	1.7055 +2	-1.2318 +2			
1.6101 +1	-1.3753 +2	9.9339 +1				
1.0862 +3	-7.7471 +2					
5.5001 +2						

2. Iterationen forts.

	a_1	a_2	b_o	b_1	b_2	c_1	c_2
	4.4809 -1	3.3155 -1	-2.1576 -1	3.7397 -1	-2.7672 -1	-8.7464 -2	-1.1590 -1
	3.3155 -1	2.8081 -1	-1.3391 -1	2.8613 -1	-1.2038 -1	-1.0002 -1	-1.2672 -1
$V2^{-1}$	-2.1576 -1	-1.3391 -1	2.5041 -1	2.9812 -2	1.5928 -1	5.5695 -2	9.3355 -2
	3.7397 -1	2.8613 -1	2.9812 -2	6.9733 -1	6.8342 -2	1.0284 -1	1.2585 -1
	-2.7672 -1	-1.2038 -1	1.5928 -1	6.8342 -2	7.8858 -1	-4.2519 -2	-4.5966 -2
	-8.7464 -2	-1.0002 -1	5.5695 -2	1.0284 -1	-4.2519 -2	-2.0815 -1	-2.7175 -1
	-1.1590 -1	-1.2672 -1	9.3355 -2	1.2585 -1	-4.5966 -2	-2.7175 -1	-3.5794 -1

a0	1.5783 +7*	
	1.30609371 -1	
	1.38822716 -1	
VV	3.27977624 -1	
	-2.24430760 -1	
	-6.37400772 -1	
	6.13778002 -1	
	7.51442917 -1	
a	4.20740809 -1	o 9.86164209 -1
	-2.72667222 -1	-2.48453610 -1
	6.55291669 -1	
b	-9.40775751 -2	
	1.26999742 +0	

25/3

3. Iterationen.

bil. 1.

V	2.22535127 +0

	-2.14173876 +1
	1.70882344 +1
V1	-5.66397607 +0
	5.17137626 +0
	-3.49491531 +0
	2.40277647 +1
	-1.78678347 +1

	1.9040 +2	-1.5425 +2	5.3857 +1	-3.8588 +1	2.8707 +1	-2.2026 +2	1.6914 +2
	1.2864 +2	-4.2858 +1	3.0739 +1	-2.4113 +1	1.8066 +2	-1.3950 +2	
$V2_{approx}$	1.8354 +1	-1.0416 +1	7.7124 +0	-6.3948 +1	4.8742 +1		
	9.7004 +0	-4.9979 +0	4.3409 +1	-3.3718 +1			
	6.3079 +0	-3.2923 +1	2.4482 +1				
	2.6173 +2	-2.0144 +2					
	1.5613 +2						

	1.9040 +2	-1.5425 +2	5.3857 +1	-3.8588 +1	2.8707 +1	-3.0228 +2	2.2549 +2
	1.2864 +2	-4.2858 +1	3.0739 +1	-2.4113 +1	2.3701 +2	-1.7722 +2	
$V2_{exakt}$	1.8354 +1	-1.0416 +1	7.7124 +0	-9.4718 +1	7.0698 +1		
	9.7004 +0	-4.9979 +0	6.5365 +1	-4.9604 +1			
	6.3079 +0	-4.8809 +1	3.5729 +1				
	4.1141 +2	-3.0083 +2					
	2.2025 +2						

3. Iterationen forts. bil. 1.

	a_1		a_2		b_o		b_1		b_2		c_1		c_2	
	-1,0787	+0	-1,3198	+0	-4,0693	-1	-8,9916	-1	-2,3181	+0	-3,7817	+0	-4,8189	+0
	-1,3198	+0	-1,4601	+0	-3,2190	-1	-1,1949	+0	-2,3753	+0	-4,1687	+0	-5,2981	+0
$V2^{-1}$	-4,0693	-1	-3,2190	-1	2,5956	-1	-1,9424	-1	-2,8782	-1	-4,9081	-1	-5,9317	-1
	-8,9916	-1	-1,1949	+0	-1,9424	-1	-4,2426	-1	-1,9473	+0	-3,0925	+0	-3,9822	+0
	-2,3181	+0	-2,3753	+0	-2,8782	-1	-1,9473	+0	-2,7761	+0	-5,7206	+0	-7,2476	+0
	-3,7817	+0	-4,1687	+0	-4,9081	-1	-3,0925	+0	-5,7206	+0	-9,8267	+0	-1,2516	+1
	-4,8189	+0	-5,2981	+0	-5,9317	-1	-3,9822	+0	-7,2476	+0	-1,2516	+1	-1,5951	+1

s0 4.9191 +4*

VV

1.54272640 +0
1.76279157 +0
5.51539093 -1
1.39897680 +0
2.36553836 +0
4.05537164 +0
5.05135035 +0

a

-1.12198559 +0
-2.03545880 +0

c

-3.06920743 +0
-5.29980396 +0

b

1.03752576 -1
-1.49305438 +0
-1.09554094 +0

4. Iterationen. bil. 1.

V	2.84799982+10						
V1	1.75481067+10						
	4.07916040 +9						
	-3.13101769+10						
	-7.27936691 +9						
	-1.69222359 +9						
	-1.04085328+11						
	-2.18045689+10						
$V2_{approx}$	5.4062 +9	1.2567 +9	-9.6460 +9	-2.2426 +9	-5.2134 +8	-3.2066+10	-6.7175 +9
	2.9213 +8	-2.2423 +9	-5.2131 +8	-1.2119 +8	-7.4540 +9	-1.5615 +9	
	1.7211+10	4.0014 +9	9.3019 +8	5.7214+10	1.1986+10		
	9.3029 +8	2.1626 +8	1.3302+10	2.7866 +9			
	5.0274 +7	3.0923 +9	6.4779 +8				
	1.9031+11	3.9870+10					
	8.3530 +9						
$V2_{exakt}$	5.4062 +9	1.2567 +9	-9.6460 +9	-2.2426 +9	-5.2134 +8	-5.9514+10	-1.2362+10
	2.9213 +8	-2.2423 +9	-5.2131 +8	-1.2119 +8	-1.3099+10	-2.7019 +9	
	1.7211+10	4.0014 +9	9.3019 +8	1.1439+11	2.3962+10		
	9.3029 +8	2.1626 +8	2.5278+10	5.2652 +9			
	5.0274 +7	5.5709 +9	1.1528 +9				
	3.6596+11	7.2635+10					
	1.4293+10						

Test med 100 datapar och korrekta indata.

Behandlade data var genererade av :

$$y(t) - 1.5y(t-1) + 0.7y(t-2) = 1.0u(t-1) + 0.5u(t-2) + e(t) - 1.0e(t-1) + 0.2e(t-2)$$

De exakta värdena på koefficienterna användes som indata. Efter fjärde iterationen av identifieringsalgoritmen erhölls resultatet:

$$y(t) - 1.439y(t-1) + 0.646y(t-2) = -0.025 + 1.284u(t-1) + 0.318u(t-2) + e(t) - 0.972e(t-1) + 0.127e(t-2)$$

Efter beräkning av spridningen erhålles följande värden på koefficienterna

	De exakta värdena
$a_1 = -1.439 \pm 0.032$	-1.5
$a_2 = 0.646 \pm 0.023$	0.7
$b_o = -0.025 \pm 0.14$	0
$b_1 = 1.284 \pm 0.25$	1.0
$b_2 = 0.318 \pm 0.19$	0.5
$c_1 = -0.972 \pm 0.061$	-1.0
$c_2 = 0.127 \pm 0.057$	0.2

1. Iteration

V	1.91265409 +2

	-3.02434007 +2
	-1.83339434 +2
	1.13041634 +1
V1	1.73320492 +1
	3.33091723 +1
	5.49860160 +1
	5.96886076 +1

	6.6451 +4	6.4581 +4	-6.0090 +3	-6.7897 +3	-7.4394 +3	-4.7487 +2	-6.9780 +2
	6.6425 +4	-5.1116 +3	-5.9896 +3	-6.7712 +3	-1.4169 +1	-4.8532 +2	
	1.0565 +3	9.9897 +2	8.8434 +2	7.8874 +1	1.1812 +2		
V2 approx	1.0403 +3	9.8429 +2	2.8753 +1	8.5601 +1			
	1.0270 +3	-1.9212 +1	3.4857 +1				
	1.0189 +3	8.0843 +2					
	1.0161 +3						

	6.6451 +4	6.4581 +4	-6.0090 +3	-6.7897 +3	-7.4394 +3	-1.0243 +3	-1.7183 +3
	6.6425 +4	-5.1113 +3	-5.9686 +3	-6.7712 +3	-1.0352 +3	-1.8918 +3	
V2 exakt	1.0565 +3	9.9897 +2	6.8434 +2	3.2571 +1	9.4100 +1		
	1.0403 +3	9.8429 +2	4.7357 +0	1.1037 +2			
	1.0270 +3	5.5547 +0	1.1223 +2				
	8.6397 +2	7.9251 +2					
	1.1143 +3						

	8.0412 -4	-5.6542 -4	9.7194 -4	-1.4079 -3	2.6354 -3	4.7057 -4	-2.6229 -4
	-5.6542 -4	4.5368 -4	-6.2960 -4	6.1509 -4	-1.1751 -3	-3.0708 -4	2.2704 -4
V2^-1	9.7194 -4	-6.2960 -4	1.9498 -2	-3.0648 -2	1.5335 -2	-1.4955 -3	1.3385 -3
	-1.4079 -3	6.1509 -4	-3.0648 -2	6.0009 -2	-3.7011 -2	2.3735 -3	-2.4433 -3
	2.6354 -3	-1.1751 -3	1.5335 -2	-3.7011 -2	3.4527 -2	6.7715 -4	4.8191 -4
	4.7057 -4	-3.0708 -4	-1.4955 -3	2.3735 -3	6.7715 -4	3.9156 -3	-2.7573 -3
	-2.6229 -4	2.2704 -4	1.3385 -3	-2.4433 -3	4.8191 -4	-2.7573 -3	2.9197 -3

ΔΔ 3.9261+19*

VV
-5.09172845 -2
4.57107509 -2
2.91572254 -2
-2.54739246 -1
1.78688805 -1
1.30759522 -2
4.80995856 -2

a
-1.44908272 +0 -1.01307595 +0
6.54289249 -1 1.51900414 -1

b
-2.91572254 -2
1.25473925 +0
3.21311195 -1

V 1,85528083 +2

	8,97264453 +1
	8,55214347 +1
	-1,54804344 +1
V1	-1,35522294 +1
	-1,06092539 +1
	-3,50918944 +1
	-2,75684522 +1

	9,6488 +4	9,4383 +4	-9,4117 +3	-1,0367 +4	-1,1112 +4	-3,2620 +3	-3,3403 +3
	9,6470 +4	-8,2656 +3	-9,3931 +3	-1,0351 +4	-2,8662 +3	-3,2622 +3	
V2 approx	1,5096 +3	1,4488 +3	1,3255 +3	3,6877 +2	4,0978 +2		
	1,4907 +3	1,4325 +3	3,0572 +2	3,6895 +2			
	1,4767 +3	2,4897 +2	3,0587 +2				
	1,8107 +3	1,6231 +3					
	1,8107 +3						

	9,6488 +4	9,4383 +4	-9,4117 +3	-1,0367 +4	-1,1112 +4	-4,7060 +3	-4,9481 +3
	9,6470 +4	-8,2656 +3	-9,3931 +3	-1,0351 +4	-4,4739 +3	-5,0415 +3	
	1,5096 +3	1,4488 +3	1,3255 +3	5,0752 +2	5,5742 +2		
V2 exakt	1,4907 +3	1,4325 +3	4,5336 +2	5,2936 +2			
	1,4767 +3	4,0938 +2	4,7359 +2				
	2,3701 +3	2,2157 +3					
	2,4431 +3						

	a_1	a_2	b_0	b_2	b_{e}	c_1	e_2
	9.1854 -4	-6.3725 -4	9.6173 -4	-1.1876 -3	2.7270 -3	7.8930 -4	-6.6122 -4
	-6.3725 -4	4.9876 -4	-6.6599 -4	4.2050 -4	-1.1220 -3	-5.4996 -4	5.1571 -4
$V2^{-1}$	9.6173 -4	-6.6599 -4	1.9428 -2	-3.0945 -2	1.5168 -2	-7.8956 -4	6.2154 -4
	-1.1876 -3	4.2050 -4	-3.0945 -2	6.2390 -2	-3.8543 -2	1.8087 -3	-2.1644 -3
	2.7270 -3	-1.1220 -3	1.5168 -2	-3.8543 -2	3.6813 -2	1.0733 -3	-1.1460 -5
	7.8930 -4	-5.4996 -4	-7.8956 -4	1.8087 -3	1.0733 -3	3.5996 -3	-3.2207 -3
	-6.6122 -4	5.1571 -4	6.2154 -4	-2.1644 -3	-1.1460 -5	-3.2207 -3	3.3846 -3

s_0 1.2566+20*

VV
-9.27411627 -3
7.07353350 -3
-2.39397369 -3
-3.19740847 -2
8.35321556 -3
-3.74164522 -2
2.43184568 -2

a
-1.43980860 +0 $\curvearrowleft$ -9.75659500 -1
6.47215715 -1 1.27581958 -1

b
-2.67632517 -2
1.28671333 +0
3.12957979 -1

3. Iterationen

V

1.85100385 +2

V1

1.33445720 +1
1.37009463 +1
-1.34955348 +0
-1.74700032 +0
-1.61094740 +0
-5.97641713 +0
-4.08309512 +0

V2 approx

.4907 +4 8.2970 +4 -8.1970 +3 -9.0593 +3 -9.7402 +3 -3.2470 +3 -2.4032 +3
8.4887 +4 -7.1672 +3 -8.1787 +3 -9.0434 +3 -1.9770 +3 -2.3425 +3
1.3327 +3 1.2753 +3 1.1605 +3 2.5984 +2 3.0131 +2
1.3155 +3 1.2603 +3 1.9870 +2 2.6214 +2
1.3024 +3 1.4629 +2 2.0071 +2
1.4477 +3 1.2593 +3
1.4474 +3

V2 exakt

8.4907 +4 8.2970 +4 -8.1970 +3 -9.0593 +3 -9.7402 +3 -3.1912 +3 -3.4041 +3
8.4887 +4 -7.1672 +3 -8.1787 +3 -9.0434 +3 -2.9712 +3 -3.5270 +3
1.3327 +3 1.2753 +3 1.1605 +3 3.3183 +2 3.8630 +2
1.3155 +3 1.2603 +3 2.8370 +2 3.6153 +2
1.3024 +3 2.4567 +2 3.0724 +2
1.7563 +3 1.6224 +3
1.8839 +3

	a_1	a_2	b_0	b_1	b_2	c_1	c_2
	9.9983 -4	-6.9776 -4	1.0351 -3	-1.2431 -3	2.9137 -3	8.8292 -4	-7.0899 -4
	-6.9776 -4	5.4615 -4	-7.1464 -4	4.6247 -4	-1.2514 -3	-6.1697 -4	5.5491 -4
	1.0351 -3	-7.1464 -4	1.9544 -2	-3.0751 -2	1.5120 -2	-6.7514 -4	5.4154 -4
$V2^{-1}$	-1.2431 -3	4.6247 -4	-3.0751 -2	6.1848 -2	-3.8345 -2	1.7214 -3	-2.1727 -3
	2.9137 -3	-1.2514 -3	1.5120 -2	-3.8345 -2	3.7261 -2	1.3002 -3	-1.6148 -5
	8.8292 -4	-6.1697 -4	-6.7514 -4	1.7214 -3	1.3002 -3	3.7116 -3	-3.1601 -3
	-7.0899 -4	5.5491 -4	5.4154 -4	-2.1727 -3	-1.6148 -5	-3.1601 -3	3.3187 -3

sO   7.2339+19*

VV
-1.18199797 -3
7.71279718 -4
-1.71260670 -3
2.20649717 -3
-4.76248442 -3
-3.22211644 -3
1.39809286 -4

a
-1.43862660 +0   c   -9.72437383 -1
6.46444435 -1   1.27442148 -1

b
-2.50506450 -2
1.28450683 +0
3.17720464 -1

4. Iterationen.

V

1.85091739 +2

V1

4.48568240 -1
4.46730338 -1
-5.10433614 -2
-5.00868559 -2
-4.88994345 -2
-1.56926228 -1
-1.49000332 -1

V2 approx

8.3025 +4	8.1108 +4	-7.9889 +3	-8.8394 +3	-9.5130 +3	-2.2133 +3	-2.2757 +3
8.3005 +4	-6.9748 +3	-7.9706 +3	-8.8234 +3	-1.8438 +3	-2.2164 +3	
1.3041 +3	1.2470 +3	1.1330 +3	2.4400 +2	2.8563 +2		
1.2870 +3	1.2321 +3	1.8325 +2	2.4682 +2			
1.2741 +3	1.3137 +2	1.8572 +2				
1.3951 +3	1.2061 +3					
1.3947 +3						

V2 exakt

8.3025 +4	8.1108 +4	-7.9889 +3	-8.8394 +3	-9.5130 +3	-2.9473 +3	-3.1718 +3
8.3005 +4	-6.9748 +3	-7.9706 +3	-8.8234 +3	-2.7399 +3	-3.2986 +3	
1.3041 +3	1.2470 +3	1.1330 +3	3.0549 +2	3.6056 +2		
1.2870 +3	1.2321 +3	2.5819 +2	3.3655 +2			
1.2741 +3	2.2110 +2	2.8277 +2				
1.6586 +3	1.5295 +3					
1.7979 +3						

	a_1	a_2	b_0	b_1	b_2	c_1	c_2
	1.0088 -3	-7.0484 -4	1.0417 -3	-1.2524 -3	2.9384 -3	8.9864 -4	-7.1462 -4
	-7.0484 -4	5.5181 -4	-7.1756 -4	4.6968 -4	-1.2725 -3	-6.2825 -4	5.5955 -4
$V2^{-1}$	1.0417 -3	-7.1756 -4	1.9541 -2	-3.0707 -2	1.5122 -2	-6.6544 -4	5.3804 -4
	-1.2524 -3	4.6968 -4	-3.0707 -2	6.1718 -2	-3.8291 -2	1.7040 -3	-2.1701 -3
	2.9384 -3	-1.2725 -3	1.5122 -2	-3.8291 -2	3.7266 -2	1.3524 -3	-2.7355 -5
	8.9864 -4	-6.2825 -4	-6.6544 -4	1.7040 -3	1.3524 -3	3.7463 -3	-3.1527 -3
	-7.1462 -4	5.5955 -4	5.3804 -4	-2.1701 -3	-2.7355 -5	-3.1527 -3	3.3069 -3

sO 6.6374+19*

VV

-3.10347452 -5
2.08852474 -5
-2.79471325 -5
5.24981819 -5
-1.34843419 -4
-1.13215303 -4
1.39934941 -5

a

-1.43859557 +0
6.46423550 -1

c

-9.72324168 -1
1.27428155 -1

b

-2.50226979 -2
1.28445434 +0
3.17855307 -1

5. Iterationen.

V 1.85091731 +2

FFFFFFF

$\lambda^2 = \frac{V}{2U}$ $\sigma =$

(1) Bo Sevård

Program för bestämning av en-stegsprediktions- och deterministisk modell samt kovariansfunktionen.

```
begin integer n,N;
  n:=read;
  N:=read;
  begin integer i,k;
  real array a,b,c[0:2],E,U,Y[-2:134];
    for k:=-n step 1 until 0 do
      E[k]:=U[k]:=Y[k]:=0;
    for k:=0 step 1 until n do
      begin
        a[k]:=read; b[k]:=read; c[k]:=read
      end;
    for i:=0 step 1 until N do
      begin
        U[i]:=read; Y[i]:=read;
        print(3,0,i+1); punch(0); punch(0);
        print(0,6,U[i]); punch(0); punch(0);
        print(Y[i]); punch(1)
      end inläsning och utskrift av in- och utdata;
    punch(15);
L2: for i:=0 step 1 until N do
      begin
        E[i]:=0;
        for k:=i step -1 until i-n do
          E[i]:=E[i]-c[i-k]×E[k]+a[i-k]×Y[k]-b[i-k]×U[k];
      end vektorn E;
for i:=0 step 1 until Ndo
      begin
        print(3,0,i+1); punch(0); punch(0);
        print(0,4,E[i]);punch(1)
      end punch(15);
    begin real array YP[0:99];
      for i:=0 step 1 until N do
        begin
          YP[i]:=Y[i]-E[i];
          print(3,0,i+1); punch(0); punch(0);
          print(0,4,YP[I]); punch(1)
        end punch(15);
    end;
    begin real array YD[-2:99];
      for k:=-n step 1 until N do
        YD[k]:=0;
      a[0]:=0;
      for i:=0 step 1 until N do
        for k:=i step -1 until i-n do
          YD[i]:=YD[i]-a[i-k]×YD[k]+b[i-k]×U[k];
      for i:=0 step 1 until N do
        begin
          print(3,0,i+1);punch(0);punch(0);
            print(0,4,YD[i]); punch(1)
        end, punch(15);
      end;;/
```

(2)

```
begin real tau; real array R[0:10];
  for i:=0 step 1 until 10 do
    R[i]:=0;
  for tau:=0 step 1 until 10 do
    for i:=0 step 1 until N-tau do
      R[tau]:=R[tau]+E[i]×E[i+tau];
  for i:=0 step 1 until 10 do
    begin
      print(3,0,i+1);punch(0);punch(0);
      print(0,4,R[i]); punch(1)
    end; punch(15)
  end
end
end
```

Test med 100 datapar och korrekta indata.

bil. 2.

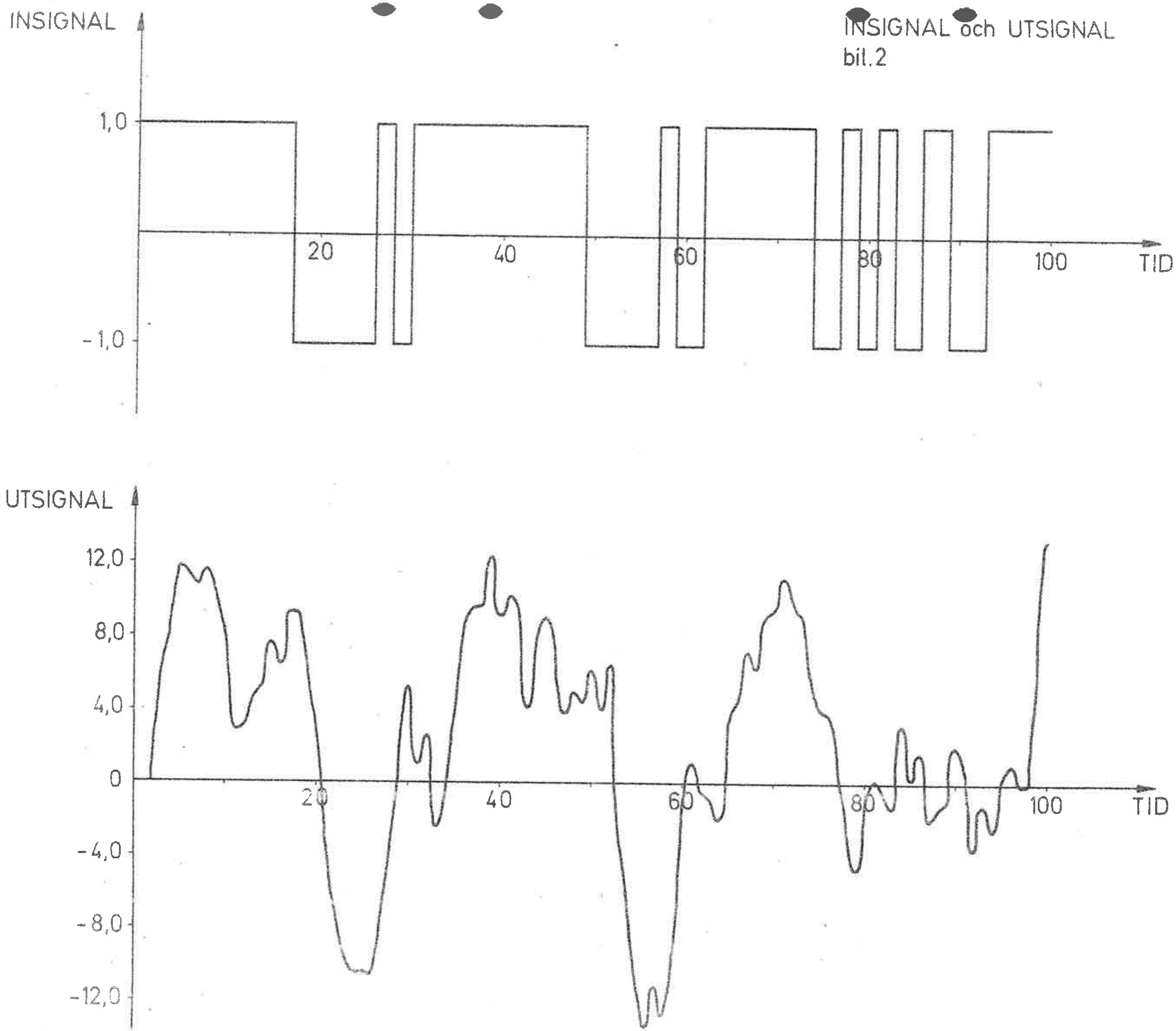
	insignal	utsignal		insignal	utsignal
1	1,000000 +0	0,000000 +0	36	1,000000 +0	8,550000 +0
2	1,000000 +0	0,000000 +0	37	1,000000 +0	9,610000 +0
3	1,000000 +0	6,155000 +0	38	1,000000 +0	9,651000 +0
4	1,000000 +0	8,325000 +0	39	1,000000 +0	1,238100 +1
5	1,000000 +0	1,185100 +1	40	1,000000 +0	9,241000 +0
6	1,000000 +0	1,149900 +1	41	1,000000 +0	1,022900 +1
7	1,000000 +0	1,083500 +1	42	1,000000 +0	9,336000 +0
8	1,000000 +0	1,161900 +1	43	1,000000 +0	4,223000 +0
9	1,000000 +0	1,010500 +1	44	1,000000 +0	7,829000 +0
10	1,000000 +0	8,349000 +0	45	1,000000 +0	9,014000 +0
11	1,000000 +0	2,870000 +0	46	1,000000 +0	7,300000 +0
12	1,000000 +0	3,117000 +0	47	1,000000 +0	3,922000 +0
13	1,000000 +0	4,708000 +0	48	1,000000 +0	5,024000 +0
14	1,000000 +0	5,264000 +0	49	1,000000 +0	4,505000 +0
15	1,000000 +0	7,687000 +0	50	-1,000000 +0	6,106000 +0
16	1,000000 +0	6,540000 +0	51	-1,000000 +0	4,089000 +0
17	1,000000 +0	9,266000 +0	52	-1,000000 +0	6,393000 +0
18	-1,000000 +0	9,271000 +0	53	-1,000000 +0	-3,035000 +0
19	-1,000000 +0	6,330000 +0	54	-1,000000 +0	-6,267000 +0
20	-1,000000 +0	2,756000 +0	55	-1,000000 +0	-1,092200 +1
21	-1,000000 +0	-3,084000 +0	56	-1,000000 +0	-1,331400 +1
22	-1,000000 +0	-6,701000 +0	57	-1,000000 +0	-1,120200 +1
23	-1,000000 +0	-9,305000 +0	58	1,000000 +0	-1,276400 +1
24	-1,000000 +0	-1,038900 +1	59	1,000000 +0	-9,779000 +0
25	-1,000000 +0	-1,041000 +1	60	-1,000000 +0	-1,205000 +0
26	-1,000000 +0	-1,053500 +1	61	-1,000000 +0	1,104000 +0
27	1,000000 +0	-7,172000 +0	62	-1,000000 +0	-5,330000 -1
28	1,000000 +0	-3,776000 +0	63	1,000000 +0	-9,290000 -1
29	-1,000000 +0	9,500000 -1	64	1,000000 +0	-1,877000 +0
30	-1,000000 +0	5,270000 +0	65	1,000000 +0	3,342000 +0
31	1,000000 +0	1,100000 +0	66	1,000000 +0	4,453000 +0
32	1,000000 +0	2,590000 +0	67	1,000000 +0	7,240000 +0
33	1,000000 +0	-2,362000 +0	68	1,000000 +0	6,332000 +0
34	1,000000 +0	-7,070000 -1	69	1,000000 +0	9,001000 +0
35	1,000000 +0	3,671000 +0	70	1,000000 +0	9,504000 +0

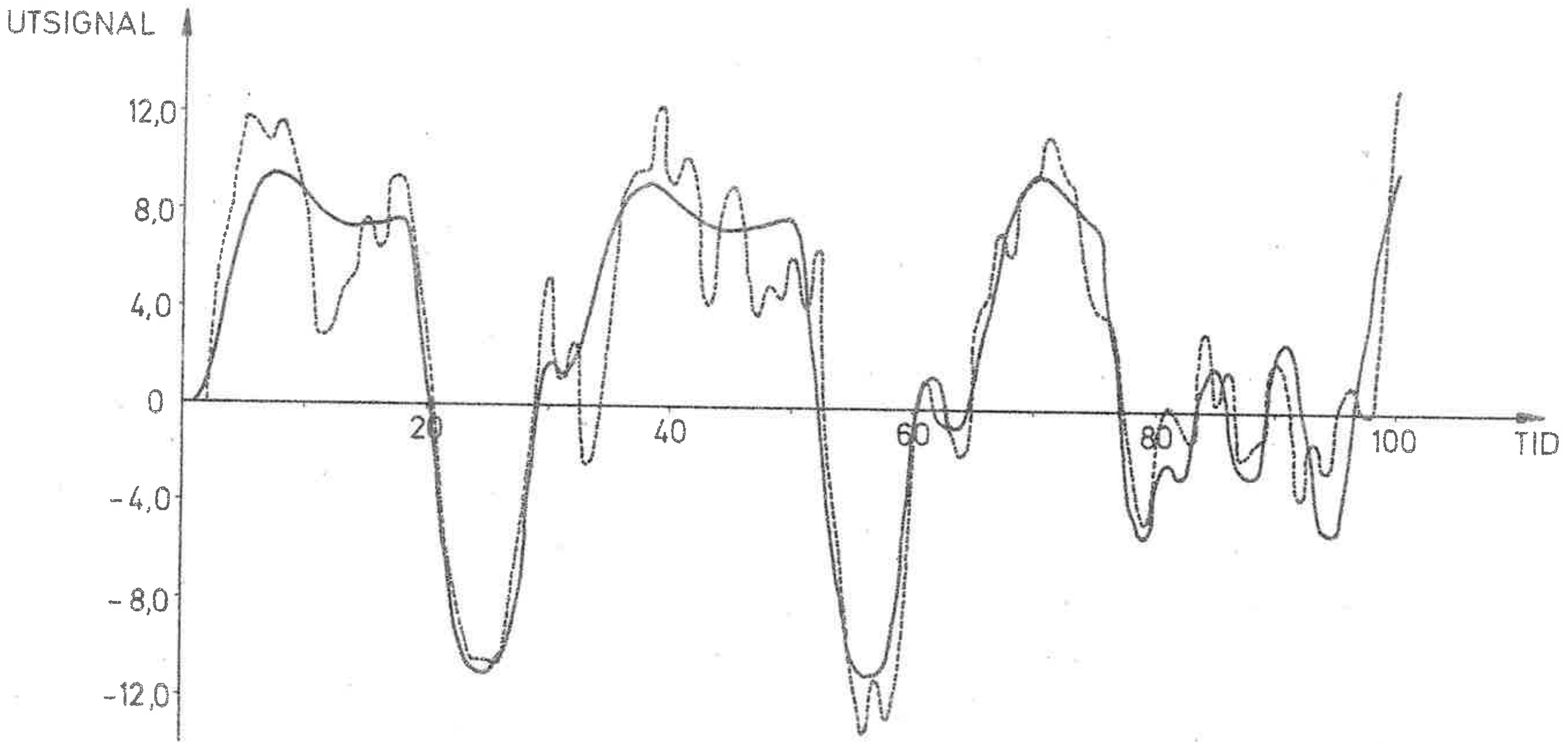
Felen e(t) (utskrift av vektorn E). bil 2.

1	2.5000	-2	36	1.5709	+0	71	1.7648	+0
2	-1.2307	+0	37	-3.8384	-1	72	1.9163	-1
3	3.3826	+0	38	-8.0970	-1	73	2.7482	-1
4	1.3330	+0	39	2.3803	+0	74	-1.9862	+0
5	3.1322	+0	40	-1.5096	+0	75	-2.2855	+0
6	1.1137	+0	41	1.5775	+0	76	8.3036	-1
7	1.0439	+0	42	7.2795	-1	77	7.7880	-1
8	2.7452	+0	43	-3.6787	+0	78	1.2712	+0
9	1.3358	+0	44	2.5378	+0	79	-9.3478	-1
10	6.8044	-1	45	1.8292	+0	80	1.6834	+0
11	-3.7060	+0	46	-7.3990	-1	81	1.3506	+0
12	-8.8398	-1	47	-3.2915	+0	82	1.2791	+0
13	1.1197	-1	48	-5.8621	-1	83	-2.7450	-1
14	-8.5384	-1	49	-1.9207	+0	84	2.6717	+0
15	7.3106	-1	50	-5.5119	-1	85	-1.5966	+0
16	-1.8827	+0	51	-1.1482	+0	86	2.9099	+0
17	1.3184	+0	52	4.9762	+0	87	5.1475	-1
18	5.0353	-2	53	-3.0437	+0	88	1.6296	+0
19	-2.1589	-1	54	2.1582	-1	89	-1.8897	-1
20	9.8662	-1	55	-1.6888	+0	90	4.8680	-1
21	-4.0404	-1	56	-1.7307	+0	91	-1.6943	+0
22	5.7530	-1	57	1.0198	+0	92	-3.1223	+0
23	5.3569	-1	58	-2.4000	+0	93	2.9341	+0
24	7.0198	-1	59	-2.0346	+0	94	1.9823	+0
25	7.2642	-1	60	1.3354	+0	95	3.1842	+0
26	-6.5966	-2	61	-9.8465	-1	96	8.5955	-1
27	2.7402	+0	62	-2.4549	+0	97	-2.6026	+0
28	1.4809	+0	63	-8.6373	-2	98	-3.6724	+0
29	1.2228	+0	64	-1.5927	+0	99	1.3579	+0
30	3.4001	+0	65	2.3346	+0	100	4.7379	+0
31	-1.1024	+0	66	-6.7354	-1			
32	1.9700	+0	67	4.6244	-1			
33	-4.8991	+0	68	-2.2549	+0			
34	-2.2177	+0	69	7.3644	-1			
35	5.6789	-2	70	6.2224	-2			

	insignal		utsignal	
71	1,000000	+0	1,124200	+1
72	1,000000	+0	1,010600	+1
73	1,000000	+0	9,176000	+0
74	1,000000	+0	6,029000	+0
75	-1,000000	+0	4,057000	+0
76	-1,000000	+0	3,810000	+0
77	-1,000000	+0	9,740000	-1
78	1,000000	+0	-2,062000	+0
79	1,000000	+0	-4,733000	+0
80	-1,000000	+0	-1,107000	+0
81	-1,000000	+0	1,220000	-1
82	1,000000	+0	-5,520000	-1
83	1,000000	+0	-1,283000	+0
84	-1,000000	+0	3,233000	+0
85	-1,000000	+0	3,190000	-1
86	-1,000000	+0	1,599000	+0
87	1,000000	+0	-2,043000	+0
88	1,000000	+0	-1,539000	+0
89	1,000000	+0	-1,031000	+0
90	-1,000000	+0	2,010000	+0
91	-1,000000	+0	4,320000	-1
92	-1,000000	+0	-3,663000	+0
93	-1,000000	+0	-1,373000	+0
94	1,000000	+0	-2,500000	+0
95	1,000000	+0	-1,460000	-1
96	1,000000	+0	9,940000	-1
97	1,000000	+0	6,500000	-2
98	1,000000	+0	-9,000000	-3
99	1,000000	+0	6,115000	+0
100	1,000000	+0	1,333600	+1

INSIGNAL och UTSIGNAL
bil. 2

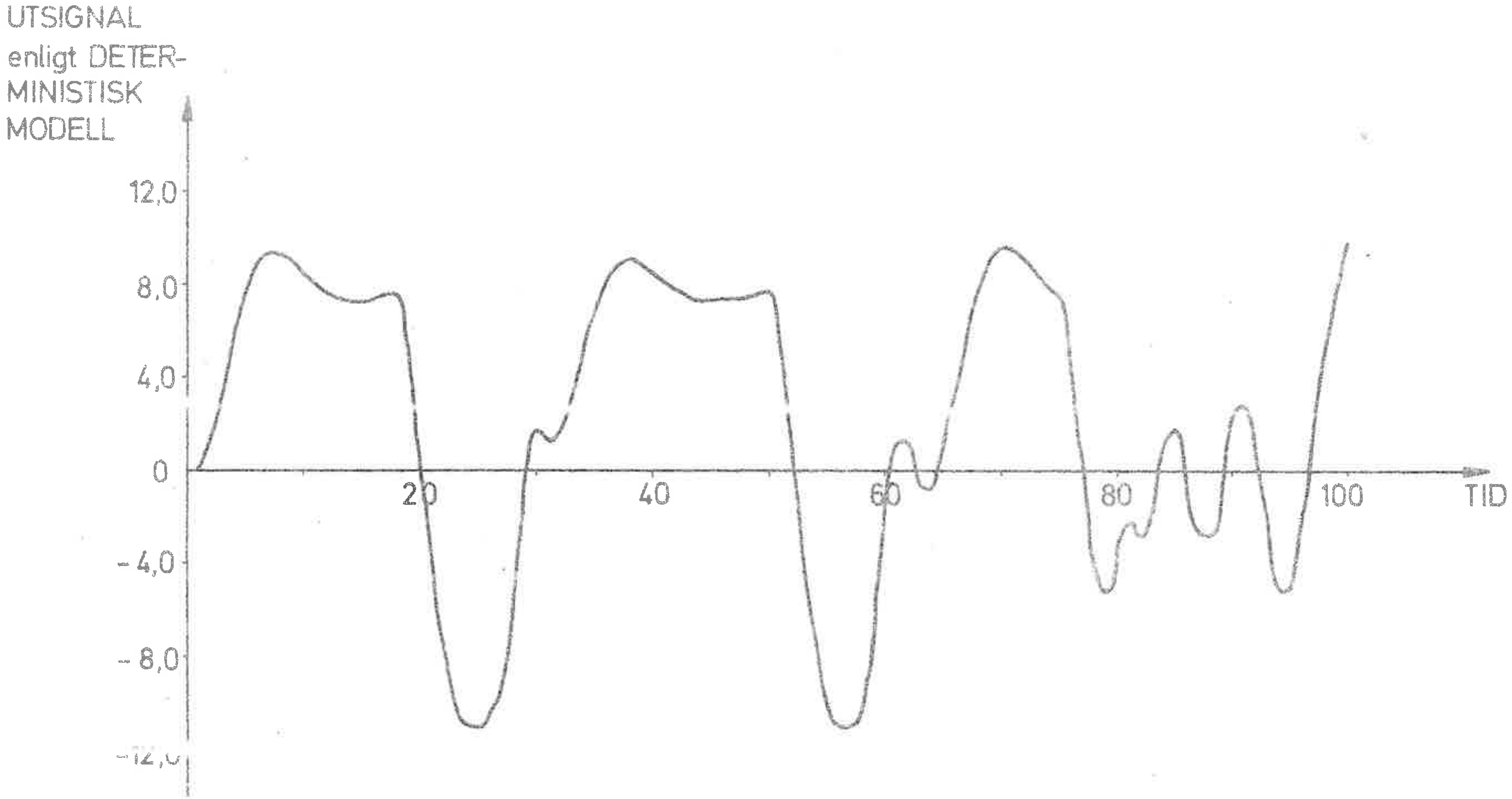



UTSIGNAL
12,0
8,0
4,0
0
-4,0
-8,0
-12,0
20
40
60
80
100
TID

Utsignalen för deterministisk modell (utskrift av vektorn YD). bil. 2.

1	-2.5000	-2
2	1.2190	+0
3	3.3445	+0
4	5.6016	+0
5	7.4788	+0
6	8.7238	+0
7	9.3040	+0
8	9.3352	+0
9	9.0053	+0
10	8.5101	+0
11	8.0101	+0
12	7.6100	+0
13	7.3569	+0
14	7.2509	+0
15	7.2617	+0
16	7.3458	+0
17	7.4599	+0
18	7.6198	+0
19	5.2165	+0
20	1.0163	+0
21	-3.4793	+0
22	-7.2398	+0
23	-9.7507	+0
24	-1.0937	+1
25	-1.1023	+1
26	-1.0381	+1
27	-9.4511	+0
28	-5.9662	+0
29	-8.6289	-1
30	1.6746	+0
31	1.3458	+0
32	1.7932	+0
33	3.2858	+0
34	5.1462	+0
35	6.8608	+0
36	8.1282	+0
37	8.8455	+0
38	9.0597	+0
39	8.9048	+0
40	8.5433	+0
41	8.1229	+0
42	7.7510	+0
43	7.4870	+0
44	7.3472	+0
45	7.3163	+0
46	7.3622	+0
47	7.4483	+0
48	7.5425	+0
49	7.6226	+0
50	7.7271	+0
51	5.2658	+0
52	1.0181	+0
53	-3.5087	+0
54	-7.2832	+0
55	-9.7942	+0
56	-1.0972	+1
57	-1.1045	+1
58	-1.0440	+1
59	-6.9619	+0
60	-1.6577	+0
61	1.1733	+0
62	1.1874	+0
63	-6.7104	-1
64	-7.9638	-1
65	8.5970	-1
66	3.3254	+0
67	5.8063	+0
68	7.7858	+0
69	9.0337	+0
70	9.5519	+0
71	9.4920	+0
72	9.0709	+0
73	8.5033	+0
74	7.9579	+0
75	7.5893	+0
76	4.8508	+0
77	5.0942	-1
78	-4.0230	+0
79	-5.1853	+0
80	-3.2449	+0
81	-2.2600	+0
82	-2.7811	+0
83	-1.6079	+0
84	1.1042	+0
85	1.6918	+0
86	1.4985	-1
87	-2.5001	+0
88	-2.7600	+0
89	-7.8627	-1
90	2.2737	+0
91	2.8451	+0
92	1.0551	+0
93	-1.8916	+0
94	-5.0285	+0
95	-5.0820	+0
96	-2.4967	+0
97	1.2607	+0
98	5.0013	+0
99	7.9605	+0
100	9.8052	+0

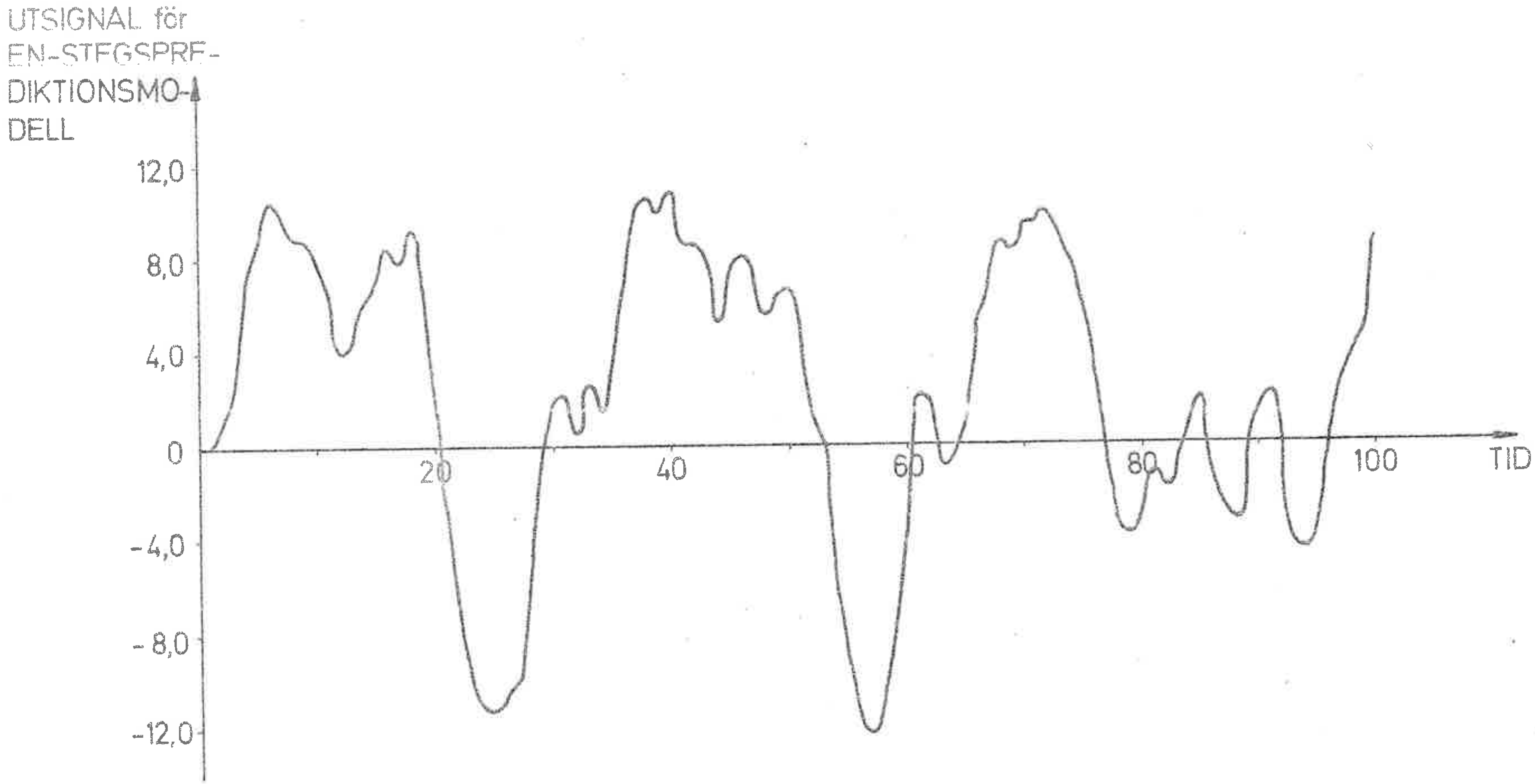
UTSIGNAL enligt DETERMINISTISK MODELL
bil.2



Utsignalen för en-stegs-prediktionsmodellen (utskrift av vektorn YP). bil. 2.

1	-2.5000 -2	36	6.9791 +0	71	9.4772 +0
2	1.2307 +0	37	9.9938 +0	72	9.9144 +0
3	2.7724 +0	38	1.0461 +1	73	8.9012 +0
4	6.9920 +0	39	1.0001 +1	74	8.0152 +0
5	8.7183 +0	40	[illegible] +1	75	[illegible] +0
6	1.0385 +1	41	8.6515 +0	76	2.9796 +0
7	9.7911 +0	42	8.6080 +0	77	1.9520 -1
8	8.8738 +0	43	7.9017 +0	78	-3.3332 +0
9	8.7692 +0	44	5.2912 +0	79	-3.7982 +0
10	7.6686 +0	45	7.1848 +0	80	-2.7904 +0
11	6.5760 +0	46	8.0399 +0	81	-1.2286 +0
12	4.0010 +0	47	7.2135 +0	82	-1.8311 +0
13	4.5960 +0	48	5.6102 +0	83	-1.0085 +0
14	6.1178 +0	49	6.4257 +0	84	5.6134 -1
15	6.9559 +0	50	6.6572 +0	85	1.9156 +0
16	8.4227 +0	51	5.2372 +0	86	-1.3109 +0
17	7.9476 +0	52	1.4168 +0	87	-2.5577 +0
18	9.2206 +0	53	8.7042 -3	88	-3.1686 +0
19	6.5459 +0	54	-6.4828 +0	89	-8.4203 -1
20	1.7694 +0	55	-9.2332 +0	90	1.5232 +0
21	-2.6800 +0	56	-1.1583 +1	91	2.1263 +0
22	-7.2763 +0	57	-1.2222 +1	92	-5.4074 -1
23	-9.8407 +0	58	-1.0364 +1	93	-4.3071 +0
24	-1.1091 +1	59	-7.7444 +0	94	-4.4823 +0
25	-1.1136 +1	60	-2.5404 +0	95	-3.3302 +0
26	-1.0469 +1	61	2.0887 +0	96	1.3445 -1
27	-9.9122 +0	62	1.9219 +0	97	2.6676 +0
28	-5.2569 +0	63	-8.4263 -1	98	3.6634 +0
29	-2.7275 -1	64	-2.8426 -1	99	4.7571 +0
30	1.8699 +0	65	1.0074 +0	100	8.5981 +0
31	2.2024 +0	66	5.1265 +0		
32	6.1998 -1	67	6.7776 +0		
33	2.5371 +0	68	8.5869 +0		
34	1.5107 +0	69	8.2646 +0		
35	3.6142 +0	70	9.4418 +0		

UTSIGNAL enligt EN- STEGSPREDIKTION
bil. 2



Kovariansfunktionsdata.

tau+1	R	
1	3.7035	+2
2	-1.9314	+0
3	1.6717	+1
4	-4.7978	+1
5	-1.1481	+1
6	2.6181	+1
7	2.2157	+1
8	2.5943	+1
9	-2.8871	+1
10	-1.5741	-1
11	4.6727	+0