JÄMFÖRELSE MELLAN TRE OLIKA METODER FÖR REELLTIDSIDENTIFIERING

GERT INGVAR KNUTSSON

Examensarbete vid institutionen för Regleringsteknik vid LTH

Ansvarig handledare:
Johan Weislander

Rapport RE - 86, augusti 1970

JÄMFÖRELSE MELLAN TRE OLIKA METODER FÖR REELLTIDSIDENTIFIERING

Examensarbete utfört vid Institutionen för regleringsteknik, Lunds Tekniska Högskola

GERT INGVAR KNUTSSON

Ansvarig handledare: Johan Wieslander

SAMMANFATTNING

I denna rapport beskrivs tre metoder för att identifiera parametrarna hos ett linjärt, tidsvariabelt system. Huvuduppgiften har varit att skriva ett modulärt FORTRAN program, som kan anpassas till de tre metoderna och olika modeller genom att man endast behöver göra smärre ändringar i programmet. Det erhållna programmet har sedan använts för testkörning på tre system:

1. Ett första ordningens system
2. Ett andra ordningens system
3. Ett första ordningens olinjärt system

En metod bygger på teorin för Kalmanestimering, en annan på minsta-kvadrat metoden och den tredje på teorin för stokastisk approximation.

ABSTRACT

This report describes three different methods of identifying a time varying system, least squares with exponential weighting, Kalman fittering, stochastic approximation. A modular FORTRAN-program, which can be adapted to different models with little effort, has been written. The program has been used on three digitaly simulated examples. Differences and similarities in the behavior of the three real-time identification algorithms are discussed.

INNEHÅLLSFÖRTECKNING

1. INLEDNING

Vid många industriella processer är det förenat med stora svårigheter att fastställa det system av differentialekvationer, som beskriver processen. Ofta känner man emellertid den generella strukturen hos det system man vill identifiera. Problemet är då att exakt kunna bestämma vissa parametrars numeriska värden.

För att styra processer som varierar med tiden t.ex. genom omgivningens störande inverkan, är det av stor betydelse att kunna identifiera processer i reell tid.

I denna rapport kommer tre metoder för reelltidsidentifiering av parametrar att anvisas. För att visa hur olika faktorer påverkar metodernas användbarhet, har ett antal testexempel simulerats på datamaskin. De frågeställningar som med dessa exempel vill besvaras, kommer att presenteras först sedan den teori, som ligger till grund för metoderna har beskrivits.

2. LITTERATURSTUDIE

I detta kapitel kommer den nödvändiga teorin för identifieringsalgoritmerna kortfattat att behandlas. De olika metoderna har beskrivits i följande litteratur:
J. Wieslander, Reelltidsidentifiering med hjälp av minsta-kvadrat metoden. Försök på pilotdata. ref.(1),
K.J. Åström, Introduction to Stochastic Control Theory. ref.(2),
Ya.Z.Tsypkin,Adaption, Learning and Self-Learning in Control Systems. ref.(3).

Från ref.(2) har hämtats teorin för Kalmanestimering och i ref.(3) beskrivs den stokastiska approximationsmetoden. För ett mera ingående studium hänvisas till respektive referens.

Reelltidsidentifiering enligt minsta-kvadrat metoden (RTLS)

Tidsinvarianta fallet

Betrakta följande systemmodell:

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

där e(t) är mätvärdenas avvikelse från systemmodellen vid tidpunkten t.

Inför

$$Y = \begin{bmatrix} y(n) \\ y(n+1) \\ . \\ . \\ y(N) \end{bmatrix} \qquad \varnothing = \begin{bmatrix} -y(n-1)\ldots\ldots-y(0) & u(n-1) & \ldots & u(0) \\ & & & \\ & & & \\ -y(N-1) & \ldots & -y(N-n) & u(N-1) \quad .. \quad u(N-n) \end{bmatrix}$$

$\Theta = (a_1 \ldots\ldots a_n \; b_1 \ldots b_n)^T$ och $E = (e(n) \ldots\ldots e(N))^T$

Resultatet av mätningarna vid tidpunkterna t = 0, 1, ..., N kan beskrivas med matrisekvationen

$$Y = \phi \cdot \theta + E$$

Vi bildar kvadratsumman av avvikelserna e(t) från tidpunkten n till N och skriver den

$$\sum_{t=n}^{N} e^2(t) = E^T E$$

Det går att visa, att förlustfunktionen

$$V(N) = \sum_{t=n}^{N} e^2(t)$$

har minimum för parameterskattningen $\hat{\theta} = (\phi^T \phi)^{-1} \phi^T Y$. Om man nu tänker sig det fall, då systemmodellens parametrar varierar med tiden, inses att skattningen väsentligen skall bero av de senaste och mest aktuella mätvärdena, medan man bör ta mindre hänsyn till de längre bort i tiden liggande värdena, Detta kan göras genom en modifiering av förlustfunktionen ovan. Inför i stället följande förlustfunktion

$$V(N) = \sum_{t=n}^{N} \lambda^{N-t} e^2(t) \qquad \lambda < 1$$

Detta innebär, att kvadratavvikelserna har viktats med exponentialfunktionen λ^t.

<u>Algoritm för reelltidsidentifiering enligt minsta-kvadrat metoden</u>

Vid tidpunkten N gäller, att skattningen $\hat{\theta}(N) = (\phi^T_N \phi_N)^{-1} \phi^T_N Y_N$ minimerar förlustfunktionen $V(N) = E^T_N E_N$. Vi erhåller en ny mätpunkt vid t = N+1 och vill beräkna skattningen $\hat{\theta}(N+1)$. Det är den skattning, som minimerar förlustfunktionen

$$V(N+1) = \sum_{t=n}^{N+1} \lambda^{N+1-t} e^2(t) = \lambda \sum_{t=n}^{N} \lambda^{N-t} e^2(t) + e^2(N+1) =$$

$$= \lambda E^T_N E_N + e^2(N+1) \qquad (2.1)$$

För att bestämma skattningen $\hat{\theta}(N+1)$ inför vi beteckningarna $y = y(N+1)$ och $\varphi = (-y(n) \ldots -y(N-n+1)\ u(N) \ldots u(N-n+1))$
För tidpunkterna $t = 0,1,\ldots..,N+1$ kan vi nu skriva

$$\begin{bmatrix} \mu Y_N \\ y \end{bmatrix} = \begin{bmatrix} \mu \Phi_N \\ \varphi \end{bmatrix} \theta + \begin{bmatrix} \mu E_N \\ e(N+1) \end{bmatrix} \qquad 0 < \mu \leq 1$$

Vi har till det tidigare ekvationssystemet, som beskriver förhållandena vid de N föregående mätningarna, lagt ytterliggare en ekvation samtidigt som de övriga multiplicerats med en faktor μ. Om $\lambda = \mu^2$ fås genom insättning i (2.1)

$$V(N+1) = \lambda E_N^T E_N + e^2(N+1) = \begin{bmatrix} \mu E_N \\ e(N+1) \end{bmatrix}^T \begin{bmatrix} \mu E_N \\ e(N+1) \end{bmatrix}$$

Minsta-kvadrat skattningen erhålles nu som i det tidsinvarianta fallet:

$$\hat{\theta}(N+1) = \left(\begin{bmatrix} \mu \Phi_N \\ \varphi \end{bmatrix}^T \begin{bmatrix} \mu \Phi_N \\ \varphi \end{bmatrix} \right)^{-1} \begin{bmatrix} \mu \Phi_N \\ \varphi \end{bmatrix}^T \begin{bmatrix} \mu Y \\ y \end{bmatrix}$$

Efter en del matrisräkning och förenklingar, vilka beskrives i detalj i ref(1), erhålles den sökta skattningen $\hat{\theta}(N+1)$.

$$\hat{\theta}(N+1) = \hat{\theta}(N) + K(N)\ (\ y - \varphi \hat{\theta}(N))$$

med K(N) enligt nedan. Införes $P(N) = \frac{1}{\lambda}(\Phi^T \Phi)^{-1}$ kan vi skriva

$$K(N) = P(N) \varphi^T\ (1 + \varphi P(N) \varphi^T)^{-1}$$

Vidare visas i ref (1), hur man kommer fram till en rekursiv formel för beräkning av P.

Sammanfattning av resultaten

Vi utgår från systemmodellen

$$y(t) = -a_1 y(t-1) - \ldots -a_n y(t-n) + b_1 u(t-1) + \ldots. + b_n u(t-n) + e(t)$$

och skriver parametrarna i vektorn $\Theta = (a_1\ a_2\ \ldots\ b_n)^T$
Vidare införes $y = y(N+1)$ och $\varphi = (-y(N)\ -y(N-1)\ \ldots u(N-n+1))$
Den skattning, som vid tidpunkten N+1 minimerar förlustfunktionen

$$V(N+1) = \sum_{t} \lambda^{N+1-t} e^2(t) \qquad \lambda < 1$$

ges av de rekursiva ekvationerna:

$$1.\quad \hat{\Theta}(N+1) = \hat{\Theta}(N) + K(N)\ (\ y - \varphi \hat{\Theta}(N))$$

$$2.\quad K(N) = P(N)\varphi^T\ (\ 1 + \varphi P(N)\varphi^T)^{-1} \qquad (2.2)$$

$$3.\quad P(N+1) = \frac{1}{\lambda}(\ P(N) - K(N)\varphi P(N)) = \frac{1}{\lambda}(\ P(N) - K(N)(\ 1+\varphi P(N)\varphi^T)K^T(N)$$

I ekvation 1 i (2.2) är y den verkliga utsignalen från systemet, medan däremot $\varphi\hat{\Theta}(N)$ är den med den aktuella skattningen av systemets parametrar väntade utsignalen. Man ser vidare, att den nya skattningen erhålles ur den gamla genom addition av en term, som beror dels på hur bra den förra skattningen var, dels av en vektor K, som beror av P.

Elementen i P mäter osäkerheten i parameterskattningen $\hat{\Theta}$. Om elementen i P är små, får man ur ekvationen 2, att elementen i K också blir små, vilket i sin tur medför att korrektionen blir liten. För skattningar enligt minsta-kvadrat metoden gäller, att de blir noggrannare ju fler mätpunkter, som finns tillgängliga. Ekvationen 3 visar nu, hur reellttidsidentifieringen fungerar. Då P divideras med $\lambda < 1$, får man en motsatt verkan, som försöker att öka osäkerheten, så att de senaste mätvärdena hela tiden kommer att påverka skattningen av $\hat{\Theta}$.

Reelltidsidentifiering enligt Kalman teori (KALID)

Betrakta följande stokastiska system:

$$\begin{cases} x(t+1) = \Phi\, x(t) + v(t) \\ y(t) = \Theta\, x(t) + e(t) \end{cases}$$

Här är x en n-dimensionell tillståndsvektor, y en p-dimensionell utsignalvektor. Vidare är v(t) en n-dimensionell brusvektor med medelvärde noll och kovariansmatrisen R1 samt e(t) en p-dimensionell brusvektor med medelvärde noll och kovariansmatrisen R2. Matriserna Φ, Θ, R1 och R2 kan bero av tiden.

Man kan visa, vilket är gjort i ref (2), att skattningen av tillståndsvektorn vid tidpunkten t+1, baserad på utsignalens värden y(0), y(1),....., y(t), ges av följande rekursiva ekvationer:

1. $\hat{x}(t+1) = \Phi\, \hat{x}(t) + K(t)(y(t) - \Theta\, \hat{x}(t))$

2. $K(t) = \Phi P(t)\Theta^T (R2 + \Theta P(t)\Theta^T)^{-1}$

3. $P(t+1) = \Phi P(t)\Phi^T + R1 - \Phi P(t)\Theta^T(\Theta P(t)\Theta^T + R2)^{-1}\Theta P(t)\Phi^T =$

$$= (\Phi - K(t)\Theta)P(t)\Phi^T + R1$$

Betrakta nu samma systemmodell som tidigare:

$$y(t) = -a_1 y(t-1) \ldots. -a_n y(t-n) + b_1 u(t-1) \ldots + b_n u(t-n) + e(t) \qquad \S$$

eller kortare

$$y(t) = \varphi(t-1)\theta(t) + e(t)$$

Inför modellens parametrar som tillståndsvariabler

$$x_1(t) = a_1 \qquad x_{n+1}(t) = b_1$$
$$x_2(t) = a_2 \qquad x_{n+2}(t) = b_2$$
$$\cdot \qquad\qquad \cdot$$
$$\cdot \qquad\qquad \cdot$$
$$x_n(t) = a_n \qquad x_{2n}(t) = b_n$$

och definiera vektorn $\theta = (-y(t-1)\ -y(t-2)\ \ldots\ u(t-n))$
§ kan nu skrivas $y(t) = \theta\, x(t) + e(t)$ och identifiering enligt minsta-kvadrat metoden kan behandlas som en Kalmanestimering med $\emptyset$ lika med enhetsmatrisen och R2 = 1. De rekursiva ekvationerna för beräkning av parameterskattningen ser följaktligen ut:

$$1.\ \hat{x}(t+1) = \hat{x}(t) + K(t)(\ y(t) - \theta\, \hat{x}(t))$$

$$2.\ K(t) = P(t)\theta^T(\ 1 + \theta P(t)\theta^T)^{-1} \qquad (2.3)$$

$$3.\ P(t+1) = P(t) - K(t)\theta P(t) + R1$$

För att erhålla full överensstämmelse med teorin för minsta-kvadrat metoden bör vektorn, som i denna framställning betecknas θ kallas φ och vektorn x betecknas med θ. De tre ekvationerna i (2.3) får då följande utseende:

$$1.\ \hat{\theta}(t+1) = \hat{\theta}(t) + K(t)(\ y(t) - \varphi(t-1)\hat{\theta}(t))$$

$$2.\ K(t) = P(t)\,\varphi^T(t-1)(\ 1 + \varphi(t-1)P(t)\,\varphi^T(t-1))^{-1}$$

$$3.\ P(t+1) = P(t) + R1 - K(t)\varphi(t-1)P(t) =$$

$$= P(t) + R1 - K(t)(\ 1 + \varphi(t-1)P(t)\varphi^T(t-1))K^T(t)$$

Reelltidsidentifiering enligt stokastisk approximationsmetoden (STAPP)

Stokastisk approximation

I åtskilliga problem i modern reglerteknik är det önskvärt att finna en extrempunkt hos en flervariabel funktion som t.ex.

$$I = Q(c_1, c_2, \ldots, c_n) = Q(\bar{c}) \tag{2.5}$$

där $\bar{c} = (c_1, c_2, \ldots, c_n)$ är en n-dimensionell vektor.

Låt $\bar{c} = \bar{c}^*$ vara extrempunkten och antag, att den utgör ett minimum. Existensen och entydigheten för det önskade minimat ges av ganska enkla villkor. Om funktionen $Q(\bar{c})$ är känd och differentierbar, så kan man finna $\bar{c} = \bar{c}^*$ genom att lösa ekvationen

$$\nabla Q(\bar{c}) = \bar{0} \tag{2.6}$$

där $\nabla Q = \left(\frac{\partial Q}{\partial c_1}, \frac{\partial Q}{\partial c_2} \ldots \frac{\partial Q}{\partial c_n} \right)$ är gradienten till $Q(\bar{c})$. För lösning av ekvation (2.6) kommer iterativa metoder till användning. En sådan metod är gradientmetoden, som definieras genom algoritmen:

$$\bar{c}(n) = \bar{c}(n-1) + \gamma(n) \nabla Q(\bar{c}(n-1)) \tag{2.7}$$

där $\gamma(n)$ bestämmer stegstorleken och kan bero av n. $\bar{c}(n)$ går mot $\bar{c}^*$, när n går mot oändligheten. Det har antagits, att $\bar{c}^*$ är den enda roten till ekvation (2.6).

I de fall då $Q(\bar{c})$ är okänd, kan $\nabla Q(\bar{c})$ approximeras med faktorn:

$$\frac{Q_+(\bar{c},\alpha) - Q_-(\bar{c},\alpha)}{2\alpha}$$

där α är ett positivt tal och

$$Q_\pm(\bar{c},\alpha) = (Q(\bar{c} \pm \alpha e_1),\ Q(\bar{c} \pm \alpha e_2), \ldots\ldots, Q(\bar{c} \pm \alpha e_n)) \tag{2.8}$$

och $e_1 = (1,0,\ldots 0)$, $e_2 = (0,1,0\ldots 0),\ldots,$ $e_n = (0,0,\ldots .1)$ (2.9)

Vidare kommer då (2.7) att ersättas med följande differensekvation:

$$\bar{c}(n) = \bar{c}(n-1) + \gamma(n)\left\{\frac{Q_+(\bar{c}(n-1),\alpha(n)) - Q_-(\bar{c}(n-1),\alpha(n))}{2\alpha(n)}\right\} \quad (2.10)$$

I denna framställning har antagits, att $Q(\bar{c})$ är en deterministisk funktion. Om emellertid $Q = Q(\bar{x}|\bar{c})$ är en stokastisk funktion av $\bar{c}$ och en slumptalsvektor $\bar{x}$ med frekvensfunktionen $P(x)$, vore det naturligt att söka extrempunkten för det förväntade värdet av Q, som kan skrivas:

$$I(c) = \int_{\bar{x}} Q(\bar{x}|\bar{c})\, P(\bar{x})\, d\bar{x} = M_x(Q(\bar{x}|\bar{c})) \quad (2.11)$$

Ekvationen, som bestämmer $\bar{c}^*$ får då formen:

$$\nabla I(\bar{c}) = M_x(\nabla_c Q(\bar{x}|\bar{c})) = \bar{0} \quad (2.12)$$

Lösningen till ekvation (2.12) kan man finna med hjälp av (2.7) och (2.10) endast då $P(x)$ är känd, så att uttrycket (2.11) kan evalueras. Det finns emellertid åtskilliga fall, då $P(x)$ är okänd. För att då finna den optimala vektorn $\bar{c}^*$ använder man gradientmetoden på samplingar av $\nabla_c Q(\bar{x}|\bar{c})$ hellre än på det förväntade värdet enligt (2.11).

Ekvationen för att finna $\bar{c} = \bar{c}^*$, med stokastisk approximationsteori, ges alltså av

$$\bar{c}(n) = \bar{c}(n-1) + \gamma(n)\nabla_c Q(\bar{x}(n-1)|\bar{c}(n-1)) \quad (2.13)$$

eller om $Q(\bar{x}|\bar{c})$ ej är bekant

$$\bar{c}(n) = \bar{c}(n-1) + \frac{\gamma(n)}{2\alpha(n)}\left\{Q_+(\bar{x}(n-1)|\bar{c}(n-1),\alpha(n)) - Q_-(\bar{x}(n-1)|\bar{c}(n-1),\alpha(n))\right\} \quad (2.14)$$

För att ekvation (2.13) skall konvergera med sannolikheten 1, dvs $P(\lim_{n\to\infty} (\bar{c}(n) - \bar{c}^*(n)) = \bar{0}) = 1$ och i medelkvadrat dvs $\lim_{n\to\infty} M(\|\bar{c}(n) - \bar{c}^*(n)\|^2) = 0$ måste vissa restriktioner läggas på Q och $\gamma(n)$. Bland annat måste stegstorleken $\gamma(n)$ avta mot noll, då n går mot oändligheten. Hastigheten med vilket detta sker får emellertid inte vara för stor. För närmare detaljer se ref (3) eller ref (4).

Användande av stokastisk approximationsteori vid parameteridentifiering

Vi betraktar ett samplat, dynamiskt system, som kan skrivas:

$$x(n) = f(x(n-1).....x(n-l)\ u(n-1)..... u(n-l_1)) \tag{2.15}$$

där f inte är känd. Inför den $l + l_1$-dimensionella vektorn:

$$\bar{z} = (x(n-1).... x(n-l)\ u(n-1).... u(n-l_1)) \tag{2.16}$$

Ekvation (2.15) kan nu skrivas $x(n) = f(\bar{z})$, där $f(\bar{z})$ kan approximeras med uttrycket $\hat{f}(\bar{z}) = \bar{c}^T\bar{z}$. Genom att använda stokastisk approximation fås nu skattningen från ekvationerna (2.13) eller (2.14) beroende på förutsättningarna.

Då man har en linjär modell, omformas ekvation (2.15) till

$$x(n) = \sum_{m=1}^{l} a_m x(n-m) + \sum_{m=1}^{l_1} b_m u(n-m) \tag{2.17}$$

där några av koefficienterna a_m, b_m får vara noll. Genom att införa vektorn

$$\bar{c} = (a_1....a_l\ b_1.....b_{l1})$$

och använda (2.16) kan den approximerade funktionen $\hat{f}(\bar{z})$ skrivas som en skalärprodukt

$$\hat{f}(\bar{z}) = \hat{\bar{c}}^T\bar{z} \tag{2.18}$$

Som ett mått på avvikelsen mellan $\hat{f}(\bar{z})$ och $f(\bar{z})$ kan man använda medelvärdet av någon strängt konvex funktion F med argumentet $(f(\bar{z}) - \hat{f}(\bar{z}))$.

Genom lämplig modifiering erhålles av(2.13) och (2.14) följande uttryck på skattningen av $\bar{c}$:

$$\bar{c}(n) = \bar{c}(n-1) + \gamma(n) F'(\bar{x}(n) - \bar{c}^T(n-1)\bar{z}(n))\ \bar{z}(n) \qquad (2.19)$$

respektive

$$\bar{c}(n) = \bar{c}(n-1) + \frac{\gamma(n)}{2\alpha(n)} \left\{ F_+(\bar{c}(n-1), \alpha(n)) - F_-(\bar{c}(n-1), \alpha(n)) \right\} \qquad (2.20)$$

Om vi antager, att insignalerna är oberoende, F är en kvadratisk funktion och

$$\gamma(n) = \frac{1}{\|z(n)\|^2}$$

så gäller ekvation (2.19).

För att få överensstämmelse med de tidigare beskrivna metoderna antar vi, att systemet kan beskrivas genom modellen:

$$y(t) = \varphi(t-1)\theta(t) + e(t)$$

Inför

$$F = -\tfrac{1}{2}(y(t) - \varphi(t-1)\theta(t))^2$$

Härur erhålles

$$F' = \varphi^T(t-1)(y(t) - \varphi(t-1)\theta(t))$$

Insättning i ekvation (2.19) ger algoritmen för skattningen av θ enligt stokastisk approximation:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \gamma(t)\varphi^T(t-1)(y(t) - \varphi(t-1)\theta(t)) \qquad (2.21)$$

Vidare bör alltså nämnaren av $\gamma(t)$ väljas:

$$(\varepsilon + \varphi(t-1)\varphi^T(t-1))$$

där ε är ett litet tal. Härigenom undvikes att nämnaren blir noll.

I ref (4) visas att ett lämpligt värde på $\gamma(t)$, som uppfyller restriktionerna är 1/t. Om däremot $\gamma(t)$ väljes konstant = γ_0 kommer inte längre algoritmen (2.21) att konvergera. I frånvaro av brus kan emellertid följande algoritm användas:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \gamma_0 \varphi^T(t-1)(y(t) - \varphi(t-1)\hat{\theta}(t)) \qquad (2.22)$$

som konvergerar snabbare än (2.21). Se ref (3).

<u>En teoretisk jämförelse mellan de tre algoritmerna</u>

Genom att betrakta algoritmerna (2.2),(2.4) och (2.21) finner man, att dessa uppvisar stora likheter. I själva skattningsekvationen är faktorn

$$y(t) - \varphi(t-1)\,\hat{\theta}(t)$$

som benämnes residualen, gemensam för samtliga. I Kalman- och minsta-kvadrat metoden föregås denna faktor av följande termer

$$P(t)\varphi(t-1)(1 + \varphi(t-1)P(t)\varphi^T(t-1))^{-1}$$

och i fallet med stokastisk approximation är motsvarande termer

$$\gamma(t)\varphi(t-1)(\varepsilon + \varphi(t-1)\varphi^T(t-1))^{-1}$$

Skillnaden utgöres av att man vid identifieringen i de förstnämnda metoderna, utnyttjar informationen i P-matrisen för att erhålla skattningen. Det har visats, att om $\lambda = 1$ eller R1 = 0

kommer P-matrisen att gå mot noll som 1/t. Vid tidsinvariant stokastisk approximation bör $\gamma(t)$ tilldelas värden motsvarande 1/t.

Man ser, att skillnaden mellan algoritmerna (2.2) och (2.4) finns i P-ekvationen. Verkan av divisionen med lambda i (2.2) och additionen av R1-matrisen i (2.4) är i stort sett lika.

Vidare kan man konstatera, att den stokastiska approximationsmetoden verkar betydligt enklare än de övriga. Detta bör även medföra, att det är den snabbaste av metoderna.

3. METOD OCH PROBLEMSTÄLLNING

Den teori, som redogjorts för i föregående kapitel, kommer nu att användas som en grund för de praktiska metoder, vilka använts för att identifiera de okända systemparametrarna. För att testa metoderna har ett antal system simulerats på dator. I detta avsnitt anges de frågeställningar, som vill besvaras med denna rapport och presentation av de simulerade systemen. I ett kommande kapitel behandlas datamaskinprogrammen och tillvägagångssättet för identifieringen.

Problemställning

Uppgiften har varit i alla försöken att minimera den aktuella förlustfunktionen, som behandlas närmare nedan, vid de olika betingelser som rått,t.ex. varierande brusstorlek verkande på systemet. Det har alltså gällt att bestämma parameteridentifieringen vid minimum av förlustfunktionen, även om skattningen inte varit den bästa. Hur skall man då finna minimum av förlustfunktionen? I samtliga fall utom ett har det uppnådda resultatet erhållits genom "manuella" variabeländringar. Vidare frågar man sig, hur kommer kovariansmatrisen R1, lambda och gamma att förändras vid olika brusförhållanden? En annan fråga av intresse är, huruvida man kan uttala sig om att den ena metoden är bättre än den andra.

Metod

Det första steget, när man vill identifiera okända systemparametrar, är att skaffa sig ett antal par av in- och utsignaler. Dessa får sedan styra ett datorprogram, som löser estimeringsekvationerna.

Följande metoder har använts för anskaffande av dessa signalvärden:

1. En sträckvis konstant insignalfunktion har fått styra den process, vi önskar identifiera, varvid utsignalen erhålles i en mot varje insignal svarande samplingspunkt.

2. In - och utsignalvärden har funnits stansade på hålkort, varvid ingen generering har varit nödvändig.

3. In - och utsignaler har beräknats genom att explicita uttryck för u(t) och y(t) varit givna.

Presentation av testexempel

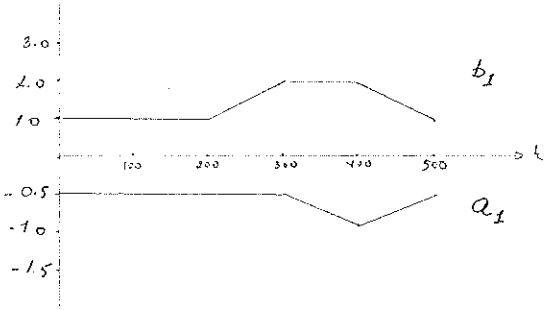
De system, på vilka undersökningarna har utförts, är följande:

Ex.1. Första ordningens system.

I detta försök har använts artificiellt genererade data enligt modellen:

$$y(t) = -a_1 y(t-1) + b_1 u(t-1) + k\, e(t)$$

där e(t) är normalfördelade N(0,1), insignalen u(t) en följd av 500 slumpvis utvalda värden, -1 eller +1, och k har antagit värdena 0.1, 0.3, 0.6 och 1.0. Parametrarna a_1 och b_1 slutligen varierar med tiden enligt figuren nedan:



Ex. 2. Andra ordningens system.

Systemmodell:

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) + b_1 u(t-1) + b_2 u(t-2) + k\, e(t)$$

I detta exempel har ut - och insignalvärden lästs in från hålkort. Endast ett värde på k har använts, nämligen k = 1. Parametrarnas variation med tiden är obekant.

Ex. 3. Första ordningens, olinjärt system. (Segerståhls exempel)

$$\begin{cases} u(t) = 5(\,1 + \sin 0.015t) + 0.5 \sin 0.5t \\ y(t) = (\,a_1 + a_2 \exp(-a_3 u(t) + a_4))u(t) + (b_1 + b_2(\exp -b_3 u(t) + b_4)) \cdot \\ \qquad \cdot u(t-1) + 1.225\, e(t) \end{cases}$$

med följande parametervärden:

$a_1 = 0 \qquad b_1 = 0.5$
$a_2 = 2.0 \qquad b_2 = 2.0$
$a_3 = 0.2 \qquad b_3 = 0.15$
$a_4 = 0.9 \qquad b_4 = 0.7$

Efter förenkling kan alltså systemmodellen skrivas:

$$y(t) = a\,u(t) + b\,u(t-1) + 1.225\,e(t)$$

4. DATAMASKINPROGRAMMET

Programmen är skrivna i FORTRAN, och körningarna har skett dels från en terminal i Lund på Uppsalamaskinen CD 3600, dels på den nyinstallerade UNIVAC-datorn i Lund. Övergången från CD:n till UNIVAC-maskinen medförde endast smärre ändringar i programmen, under det att en stor tidsvinst erhölls.

Programmens uppbyggnad

Eftersom programmet från början såg ut att bli ganska besvärligt, bestämde jag mig för att dela upp det i tre huvudprogram, var och ett hörande till sin metod, och i subrutiner. Programmet blir då mera flexibelt, dvs det kan användas för många ändamål med små förändringar t.ex. genom att byta ut någon subrutin. En annan fördel är, att man kan testa varje programdel för sig och därigenom lättare finna eventuella fel i programmet. På följande sidor följer en förteckning över använda huvudprogram och subrutiner samt deras funktion.

PROGRAM RTID PROGRAM BORIS PROGRAM FLEPO	Förberedelser och administration av identifieringen Författare: GI Knutsson
SUBROUTINE INPUT	Inläsning eller generering av in- och utsignaler
SUBROUTINE RTLS SUBROUTINE KALID SUBROUTINE STAPP	Identifieringsalgoritmerna för de tre metoderna Författare: J Wieslander
SUBROUTINE PLOTS SUBROUTINE UPLOTS	Används för plottning
SUBROUTINE PRBN	Genererar PRBS-signaler Författare: J Valis
SUBROUTINE RANSS SUBROUTINE MCNODI	Genererar normalfördelade slumptal

SUBROUTINE GEPODA	Genererar pseudodata Författare: J Wieslander
SUBROUTINE GFLPW	Utför minimering enligt Fletcher-Powells metod

Program FLEPO

Beroende på att programmen till sin uppbyggnad är ganska lika, kommer här endast att närmare granskas FLEPO, vilket är det mest komplicerade. För övrigt hänvisas till programutskrifterna i appendix B.

I huvudprogrammet sker följande:

1. INPUT anropas. Utskrift erhålles av in- och utsignalvärden, brus och de verkliga värdena på de parametrar, som skall identifieras.

2. Vektorn ALPHA tilldelas begynnelsevärden, och GFLPW anropas. GFLPW anropar i sin tur F, som är en function subroutine, ett antal gånger tills minimum av förlustfunktionen funnits. Efter ett visst antal iterationer erhålles resultat och utskrift av förlustfunktionens och vektorn ALPHAs värden. Uthopp från GFLPW sker då testvillkoren, som anges av parametrarna i anropet, är uppfyllda.

3. Kovariansmatrisen R2M beräknas ur de slutgiltiga ALPHA-värdena och skrives ut.

4. Av erfarenhet har man funnit det lämpligt, att som startvärden för P ansätta en diagonalmatris med stora värden (100 - 1000) i diagonalen. Med detta startvärde på P är initialvärdet på parameterskattningen T likgiltigt. Vidare sker bildandet av vektorn FI, som innehåller de senaste mätpunkterna, och dess nollställning.

5. Identifieringen utföres och beräkning av tre olika förlustfunktioner göres. Förlustfunktionernas värden och de identifierade parametrarna skrives ut.

5. KÖRNINGAR PÅ DATAMASKIN

Förlustfunktionerna

Som jämförelsekriterium har i samtliga exempel använts en förlustfunktion E1, vilken utgöres av kvadratsumman av residualerna för mätpunkterna. För att minska inverkan av alltför stora residualer i början av identifieringen har nollställning skett efter de 25 första identifieringarna. Residualen utgör skillnaden mellan den verkliga utsignalen och den enligt den aktuella av systemets parametrar förväntade utsignalen.

I KALID - identifieringen införs även en normerad förlustfunktion med beteckningen E i programutskrifterna. Normeringen består i att E1 divideras med faktorn ($1 + \varphi P \varphi^T$).

Ytterliggare en förlustfunktion kommer till användning i exempel 3. Den betecknas med V och har följande utseende:

$$\sum_{N=26}^{400} \log \mathrm{DENOM} + N(\log(E/N) + 1)$$

DENOM utgöres just av den tidigare nämnda faktorn ($1 + \varphi P \varphi^T$). Denna förlustfunktion är härledd i ref (5) och utgör likelihoodfunktionen för skattningen av R1 och R2, när mätpunkterna är givna.

Problemet blir nu alltså att minimera den aktuella förlustfunktionen med hjälp av de möjligheter, som står till buds i de tre metoderna. Redan nu kan omtalas, att något minimum aldrig hittats för funktionen E.

Ex.1. Första ordningens system

Systemet som skall identifieras ges av:

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

där $e(t) \in N(0,1)$ och parametrarna har utseendet:

$$a(t) = \begin{cases} -0.5 & t \leq 300 \\ -0.5 - 0.0045(t - 300) & 300 < t \leq 400 \\ -0.95 + 0.0045(t - 400) & 400 < t \leq 500 \end{cases}$$

$$b(t) = \begin{cases} 1.0 & t \leq 200 \\ 1.0 + 0.02(t - 200) & 200 < t \leq 300 \\ 3.0 & 300 < t \leq 400 \\ 3.0 - 0.02(t - 400) & 400 < t \leq 500 \end{cases}$$

Exemplet har valts, dels för att det är enkelt, av första ordningen, dels för att det är intressant ur parametersynpunkt. I nedanstående tabell visas, hur parametrarna varierar.

Intervall	a	b
0 - 200	konstant	konstant
201 - 300	konstant	↗
301 - 400	↘	konstant
401 - 500	↗	↘

Reelltidsidentifiering enligt minsta-kvadrat metoden

Identifieringsalgoritmen, som använts, ges av (2.2) med variationen av lambda för att erhålla minimum av förlustfunktionen

$$E1 = \sum_{t=6}^{500} res^2(t)$$

Identifieringen har utförts för fyra olika k, nämligen k = 0.1, 0.3, 0.6 och 1.0. Figurerna 1 och 2 visar resultatet av identifieringen för det minsta och största k-värdet. För små k-värden

visar det sig, att överensstämmelsen mellan de identifierade och sanna värdena är god. Med ökande k-värden, vilket medför mer brus på systemet, finner man, att skattningen blir allt sämre och även får svårt att följa parametervariationerna. Diagrammen 1 och 2 visar E1:s beroende av lanbda för k = 0.1 och 1,0. Ur dessa ser man, att lambda och förlustfunktionen ökar med stigande k. Då k ökar, blir det svårare att identifiera parametrarna på grund av brusets större inverkan, residualen blir större och alltså även förlustfunktionen.

k	λ	E1
0.1	0.66	9.011
0.3	0.830	59.76
0.6	0.890	214.8
1.0	0.925	568.4

Värden på lambda för minimum av
$E1 = \sum_{26}^{500} res^2(t)$ för olika k.

Identifiering med KALID

Vid identifieringen har använts algoritmen (2.4) beskriven i kapitel 2. Matrisen R1 skall nu tilldelas värden, så att E1 minimeras. Hur bör R1 se ut? I allmänhet vet man inte,hur parametrarna varierar. Om så är fallet, har det visat sig, att ett gott val är att sätta R1 = r·I. I detta exempel har detta val på R1 använts. Man skulle emellertid eventuellt få ett bättre resultat genom ett annorlunda val av kovariansmatrisen. Detta har inte undersökts för detta exempel men däremot för de båda senare. Problemet blir alltså att bestämma värdet på r.

Resultatet av estimeringen för k = 0.1 och 1.0 visas i figurerna 3 och 4. Av försöket framgår, att för de erhållna r-värdena är metoden känslig för brus, då i synnerhet för de tre högre k-värdena. Däremot följes parametervariationerna tillfredsställande i samtliga fall.

I diagrammen 3 och 4 har förlustfunktionen plottats som funktion av r. E1 ökar med stigande k medan r avtar.

k	r	E1
0.1	0.050	8.873
0.3	0.013	72.01
0.6	0.004	282.6
1.0	0.001	781.1

Värden på r för minimum av

$$E1 = \sum_{t=26}^{500} res^2(t) \text{ för olika k}$$

Identifiering enligt stokastisk approximation

Algoritmen (2.21) med

$$\gamma(t) = \frac{GAM}{GAM + \varphi(t-1)^T \varphi(t-1)}$$

har använts vid identifieringen och GAM har haft följande utseende:

$$GAM = \begin{cases} 1.0 & t \leq 10 \\ 1.0 - \dfrac{GAM - GAMZ}{40}\,(t - 10) & 10 < t \leq 50 \\ GAMZ & t > 50 \end{cases}$$

Gam har ett högt värde i början, $t \leq 10$, för att så snabbt som möjligt förmå skattningen att komma i närheten av de sanna värdena. Därefter avtar GAM linjärt mot det konstanta värdet GAMZ, som bibehålles för $t > 50$.

Resultatet av identifieringen visas i figurerna 5 och 6. Man lägger märke till att metoden inte verkar så känslig för brus och att den får svårt att följa parametervariationerna vid litet γ. Observera att skalindelningen för parameter a_1 i figur 6 är annorlunda än i de övriga figurerna och kan därför verka missvisande.

I nedanstående tabell visas värden på γ och $(E1)_{min}$ vid varierande brusstorlek.

k	GAMZ	E1
0.1	0.815	13.11
0.3	0.40	69.61
0.6	0.20	249.0
1.0	0.10	628.6

Sammanfattning av resultaten

KALID verkar att kunna följa parametervariationerna tillfredsställande, medan de båda övriga tycks ha svårare, åtminstone vid stora störningar. Däremot kan STAPP användas med fördel, då parametrarna inte varierar med tiden, i synnerhet på grund av sin enkelhet. RTLS har givit ett gott resultat, till och med något bättre än KALID, bland annat ur den synpunkten, att den har gett de minsta värdena på förlustfunktionen.

Som jämförelse mellan metoderna kan man även använda nedanstående tabell, som visar förlustfunktionens värde för minpunkten och vid fyra olika k.

k	RTLS	STAPP	KALID
0.1	9.011	13.11	8.873
0.3	59.76	69.61	72.01
0.6	214.8	249.0	282.6
1.0	568.4	628.6	781.1

Ex. 2. Andra ordningens system

I detta exempel ges systemmodellen av:

$$y(t) = -a_1\, y(t-1) - a_2\, y(t-2) + b_1\, u(t-1) + b_2\, u(t-2) + e(t)$$

där a_1 och a_2 hålls konstanta medan b_1 och b_2 varierar med tiden och visar sig vara ganska starkt kopplade. Påpekas bör, att detta förhållande inte var känt vid försökets början utan har kommit fram under försökets gång. e(t) är vitt brus. Ut - och insignal är plottade i figur 7.

Identifiering med STAPP

GAM har haft samma utseende som i exempel 1. Parameterskattningarna kan beskådas i figurerna 8 och 9. Det bekräftas ytterligare i detta försök, vad som tidigare sagts, att STAPP har svårt att hinna med vid snabba förändringar i parametrarna. Förlustfunktionens minvärde är E1 = 9749 och då har GAMZ värdet 0.5.

Identifiering med KALID

Algoritmen för Kalmanestimeringen beskriven i kapitel 2 har använts. Till en början har matrisen R1 valts på formen r·I. Minimum för E1 ges av r = 0.06 och förlustfunktionens värde 3220. Tilläggas kan, att minimat är flackt. Samma värden som ovan har erhållits för följande värden på r: 0.04(0.005)0.08

Kan man sänka förlustfunktionens värde genom att välja kovariansmatrisens element på annorlunda sätt? Genom att betrakta parameterskattningarna erhållna ovan, kan man draga några slutsatser angående elementen i R1. Man ser, att b_1 tycks minska hela tiden, samtidigt som b_2 ökar. Vidare verkar a_1 och a_2 vara konstanta. Ett gott försök bör vara att välja negativt tecken på elementen r_{34} och r_{43}. Eftersom R1 är symmetrisk, skall de båda elementens numeriska värde även vara lika.

I tabellen på nästa sida visas förlustfunktionens värde för några olika kovariansmatriser.

R1				E1
0.06	0	0	0	
0	0.06	0	0	
0	0	0.06	0	3220
0	0	0	0.06	
0.001	0	0	0	
0	0.001	0	0	
0	0	0.005	0	2450
0	0	0	0	
0.01	0	0	0	
0	0.1	0	0	
0	0	0.1	-0.5	311000
0	0	-0.5	0.01	
0.001	0	0	0	
0	0.001	0	0	
0	0	0.05	-0.025	2200
0	0	-0.025	0.05	
0.0001	0	0	0	
0	0.0001	0	0	
0	0	0.01	-0.01	2751
0	0	-0.01	0.01	

Värden på förlustfunktionen E1 = $\sum_{t=26}^{500} res^2(t)$ för olika R1

Resultatet av parameteridentifieringen visas i figurerna 10 och 11, varvid följande matris har använts:

$$\begin{bmatrix} 0.001 & 0 & 0 & 0 \\ 0 & 0.001 & 0 & 0 \\ 0 & 0 & 0.05 & -0.025 \\ 0 & 0 & -0.025 & 0.05 \end{bmatrix} \qquad (5.1)$$

Förlustfunktionens värde med denna matris blir 2200. Resultatet har förbättrats betydligt med detta val av kovariansmatrisen. Jämför det tidigare värdet på samma förlustfunktion, 3220.

Identifiering med RTLS

Även i detta exempel uppvisar minsta-kvadrat metoden det minsta värdet på förlustfunktionen E1, 2044, och det inträffar för lambda = 0.795. Parameterskattningarna är plottade i figurerna 12 och 13. Vid en jämförelse med figurerna 10 och 11, som visar skattningarna enligt KALID, finner man att överensstämmelsen är stor.

Sammanfattning

Ur försöket framgår klart, att STAPP inte är användbar vid snabba ändringar i parametrarna. Däremot verkar RTLS och KALID att vara tämligen likvärdiga och tycks identifiera parametrarna tillfredsställande. Noteras bör emellertid, att även i detta exempel uppvisar RTLS det minsta värdet på E1.

Av försöket framgår vidare att a priori kunskap om R1-matrisen är av stort värde. Ur tabellen på föregående sida ser man, hur ogynnsamt en felaktig gissning av elementen i kovariansmatrisen inverkar på algoritmen och därmed förlustfunktionen. Har man ingen a priori kunskap om R1, bör man åtminstone till en början använda matriser av formen R1 = r*I.

I nedanstående tabell visas E1:s värden för de tre metoderna:

Metod	E1
KALID	2201
RTLS	2044
STAPP	9749

Ex.3. Boris Segerståhls exempel

Algoritmen (2.4) har använts på detta exempel, somhämtats från ref (6). Systemet ges av:

$$y(t) = a(t)\, u(t) + b(t)\, u(t-1) + 1.225\, e(t)$$

där e(t) är oberoende N(0,1) slumptalsvariabler och vidare gäller:

$$\begin{cases} a(t) = 2 \exp(-0.2\, u(t) + 0.9) \\ b(t) = 0.5 + 2 \exp(-0.15\, u(t) + 0.7) \end{cases}$$

Man ser, att parametrarna i detta exempel är ickelinjära funktioner av insignalen.

I detta försök bestämmes matrisen R1 på två sätt, dels som den matris som minimerar förlustfunktionen

$$E1 = \sum_{t=26}^{200} res^2(t)$$

dels som den matris som minimerar likelihoodfunktionen

$$V = \sum_{N=26}^{200} \log DENOM + N(\log(E/N) + 1)$$

Minimeringen enligt det första sättet har skett på samma sätt som tidigare, medan för minimering av V har använts en standardsubrutin baserad på Fletcher - Powells metod för minimering. I tabellen nedan visas E1:s värde för några olika R1.

Plottning av parameterskattningarna och de verkliga värdena visas i figurerna 14 och 15. Följande R1-matris har använts vid minimeringen av E1:

$$R1 = \begin{bmatrix} 0.0060 & 0.0003 \\ 0.0003 & 0.0001 \end{bmatrix} \qquad (5.2)$$

Motsvarande skattningar för V finns i figurerna 16 och 17 med R1-matrisen:

$$R1=\begin{bmatrix} 0.00788 & 0. \\ 0. & 0.00022 \end{bmatrix} \qquad (5.3)$$

Vid en jämförelse mellan de båda sätten, visar skattningarna stor likhet. Man konstaterar vidare att parameterskattningarna betydligt avviker från de verkliga värdena. Detta kan åtminstone i senare fallet förklaras. För att utnyttja maximum likelihood metoder skall modellen vara linjär och tidsinvariant, vilka förutsättningar inte alls är uppfyllda i det här exemplet.

R1		E1
0.00600	0.00030	
0.00030	0.00010	495.98
0.00500	0.00030	
0.00030	0.00010	496.20
0.00400	0.00030	
0.00030	0.00020	496.54
0.00330	0.00030	
0.00030	0.00270	496.84
0.00600	0.00300	
0.00300	0.00010	502.48

Värden på E1 = $\sum_{t=26}^{200} res^2(t)$ för olika R1

6. SAMMANFATTNING AV RESULTAT

I denna rapport har tre olika metoder för reelltidsidentifiering presenterats och genom ett antal exempel jämförts under lika omständigheter. Ett genomgående drag för metoderna har varit motsättningen mellan okänslighet för mätbrus och förmåga att följa snabba parametervariationer. Minsta-kvadrat metoden har i dessa exempel visat sig vara något bättre än Kalmanestimeringen. Båda metoderna har också visat sig funktionsdugliga och har på ett tillfredsställande sätt utfört identifieringen. Den stokastiska approximationsmetoden har emellertid uppenbara svårigheter att följa snabba parametervariationer. Metoden har dock en stor fördel i sin enkelhet och därmed snabba arbetssätt.

APPENDIX A

REFERENSER

(1) J. Wieslander: Reelltidsidentifiering med hjälp av minsta-kvadrat metoden, Försök på pilotdata, Rapport 6810 december 68, Lunds tekniska högskola, Institutionen för reglerteknik.

(2) K.J. Åström: Introduction to Stochastic Control Theory, Academic Press, 1970.

(3) Ya.Z. Tsypkin: Adaption, Learning and Self-Learning in Control Systems, Survey Paper, IFAC, London, 1966.

(4) D.J. Wilde: Optimum Seeking Methods, Prentice Hall, Englewood, Cliffs., N.J., 1964.

(5) T. Bohlin: Real-Time Estimation of Time-Variable Process Characteristics, TP 18.190, IBM Nordic Laboratory, Lidingö, Sweden.

(6) B. Segerståhl: A Theory of Parameter Tracking Applied to Slowly Varying Nonlinear Systems, Technical Session 5, IFAC, Warsaw, 1969.

APPENDIX B

PROGRAMUTSKRIFTER

FTN5.5B 23/02-70

```
      PROGRAM RTID
      DIMENSION T(10),FI(30),U(2000),Y(2000),A(500),B(500)
      NPOI=500
      NMAX=4
      CALL INPUT(U,Y,NPOI)
      READ 205,NX
      NN=2
      DO 50 II=1,NX
      READ 210,GAMZ
      DO 5 I=1,NN
      T(I)=0.
    5 FI(I)=0.
      GAM=1.
      RIKT=(GAM-GAMZ)/40.
      E1=0.
      J=0
      Y1=0.
      U1=0.
      PRINT 215
      DO 40 I=1,NPOI
      IF(50-I) 30,20,20
   20 IF(10-I) 25,30,30
   25 GAM=GAM-RIKT
   30 CONTINUE
      FI(1)=-Y1
      FI(2)=U1
      Y1=Y(I)
      U1=U(I)
      CALL STAPP(T,FI,Y1,NN,NMAX,GAM,RES)
      A(I)=T(1)
      B(I)=T(2)
      E1=E1+RES*RES
      IF(1.E0.25)E1=0.
      J=J+1
      IF(J-10) 40,35,35
   35 J=0
      PRINT 220,I,RES,E1,A(I),B(I),GAM
   40 CONTINUE
   50 CONTINUE
      CALL EXIT
  205 FORMAT(I10)
  210 FORMAT(F10.5)
  215 FORMAT(1H1,*TIME*,10X,3HRES,12X,2HE1,14X,2HA1,13X,2HB1,13X,3HGAM,/
     * /)
  220 FORMAT(I5,2E15.3,3F15.3)
      END
```

FTN5.58 23/02-70

```
      SUBROUTINE INPUT(U,Y,NPOI)
      DIMENSION FI(30),PAR(30),U(2000),Y(2000),IRE(256)
      N=1
      NPAR=3*N
      READ 101,(PAR(I),I=1,NPAR)
      JBY=100
      J=0
      NPG=1
      DO 1 I=1,NPAR
    1 FI(I)=0.
      IA=256
      LENGTH=127
      AMP=1.
      IPR=1
      CALL PRBN(LENGTH,IRE,IA)
      IRAN=3249
      NPOI=500
      PRINT 100
      DO 50 L=1,NPOI
      J=J+1
      CALL RANSS(IRAN,E)
      E=1.0*E
      U(L)=PRB(IPR,IRE,LENGTH,AMP,IA)
      IF (J-JBY) 35,30,30
   30 J=0
      NPG=NPG+1
   35 GO TO(40,40,36,37,38),NPG
   36 PAR(2)=PAR(2)+0.02
      GO TO 40
   37 PAR(1)=PAR(1)-0.0045
      GO TO 40
   38 PAR(2)=PAR(2)-0.02
      PAR(1)=PAR(1)+0.0045
   40 CALL GEPODA(PAR,FI,N,Y(L),E,U(L))
   50 PRINT 111,L,PAR(1),PAR(2),U(L),Y(L)
      RETURN
  100 FORMAT(* TIME      A=PAR      B=PAR      INPUT      OUTPUT*,//)
  101 FORMAT(8F10.4)
  111 FORMAT(I4,1X,4F10.3)
      END
REWIND,69
LGOEDIT

EXECUTION STARTED AT   1935 -44
```

```
0005    NRUCS
0006    NIO2S
0007    NPRTS
0010    NIO1S
0011    NSTOPS
```

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

```
0000    004155 103F      0000    004157 104F      0000    004163 105F      0000    004205 106F      0001
0001    000053 123G      0001    000061 131G      0001    000062 134G      0001    000103 150G      0001
0001    000173 202G      0001    000177 30L       0000 R  004147 E1      0000 R  000012 FI        0000
0000 I  004144 J         0000 I  004141 JPKT      0000 I  004151 K       0000 I  004140 K1        0000
0000 I  004154 M         0000 I  004150 N         0000 I  004137 NN      0000 I  004136 NPAR      0000
0000 I  004135 N1        0000 R  000050 P         0000 R  004153 RES     0000 R  004142 RL        0000
0000 R  000214 U         0000 R  004145 U1        0000 R  002164 Y       0000 R  004146 Y1
```

```
00101     1*          DIMENSION T(10),FI(30),P(10,10),U(1000),Y(1000)
00103     2*          NPOI=500
00104     3*          CALL INPUT(U,Y,NPOI)
00105     4*          N1=2
00106     5*          NPAR=2*N1
00107     6*          NN=2*N1
00110     7*          DO 50 K1=1,10
00113     8*          JPKT=10
00114     9*          READ 103,RL
00117    10*          PRINT 104,RL
00122    11*          DO 5 I=1,NN
00125    12*          T(I)=0.0
00126    13*        5 FI(I)=0.0
00130    14*          DO 15 I=1,NN
00133    15*          DO 10 J=1,NN
00136    16*       10 P(I,J)=0.0
00140    17*       15 P(I,I)=100.
00142    18*          U1=0.0
00143    19*          Y1=0.0
00144    20*          E1=0.0
00145    21*          PRINT 105
00147    22*          DO 30 N=1,NPOI
00152    23*          K=NPAR+1
00155    24*          DO 20 L=2,NPAR
```

```
00156     25*           K=N-1
00157     26*       20  FI(K)=FI(K-1)
00161     27*           FI(1)=-Y1
00162     28*           FI(N1+1)=U1
00163     29*           Y1=Y(N)
00164     30*           U1=U(N)
00165     31*           CALL RTLS(T,P,FI,Y1,NN,10,RL,RES)
00166     32*           E1=E1+RES*RLS
00167     33*           IF(N.EQ.25)E1=0.0
00171     34*           IF(N-JPRT) 30,25,25
00174     35*        25 JPRT=JPRT+10
00175     36*           PRINT 106,N,RES,E1,(T(M),M=1,NN)
00206     37*        30 CONTINUE
00210     38*        50 CONTINUE
00212     39*       103 FORMAT(F10.5)
00213     40*       104 FORMAT(1H1,10X,3HRL=,F10.5)
00214     41*       105 FORMAT(1H0,6X,92HTIME           RES              E1               T1
00214     42*          *          T2              T3              T4,//)
00215     43*       106 FORMAT(6X,I5,2E15.3,4F15.4)
00216     44*           END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    0 SEC.
PHASE 2 TIME =    1 SEC.
PHASE 3 TIME =    0 SEC.
PHASE 4 TIME =    0 SEC.
PHASE 5 TIME =    0 SEC.
PHASE 6 TIME =    0 SEC.


TOTAL COMPILATION TIME =    1  SEC
```

```
0005    NIO2$
0006    NERR3$


   STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   000001 101F       0001   000012 107G       0000 I 000000 I          0000   000004 INJP$



00101      1*           SUBROUTINE INPUT(U,Y,NPOI)
00103      2*           DIMENSION U(500),Y(500)
00104      3*           NPOI=500
00105   *DIAGNOSTIC*    POSSIBLE REDUNDANT LOADING OF U .
00105   *DIAGNOSTIC*    POSSIBLE REDUNDANT LOADING OF Y .
00105      4*           READ 101,((U(I),Y(I)),I=1,500)
00114      5*           RETURN
00115      6*       101 FORMAT(10F8.3)
00116      7*           END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     2 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    0 SEC.
PHASE 2 TIME =    0 SEC.
PHASE 3 TIME =    0 SEC.
PHASE 4 TIME =    0 SEC.
PHASE 5 TIME =    1 SEC.
PHASE 6 TIME =    0 SEC.


TOTAL COMPILATION TIME =    1  SEC
```

```
0005    GFLPW
0006    KALID
0007    NEXP6$
0010    NPRT$
0011    NI02$
0012    NI01$
0013    ALOG
0014    NSTOP$
```

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

0001		000112	131G	0001		000113	133G	0001		000125	141G	0001		000133	147G	0001
0001		000155	166G	0001		000164	172G	0001		000273	226G	0001		000250	24L	0001
0000		000407	303F	0000		000421	304F	0000		000424	308F	0000		000444	309F	0000
0000	R	000404	DENOM	0000	R	000370	DUMMY	0000	R	000375	E	0000	R	000374	E1	0003
0000	R	000161	FI	0000	I	000372	I	0000	I	000373	J	0000	I	000371	JPKT	0000
0000	I	000401	L	0000	I	000406	M	0000	I	000367	MSEC	0000	I	000377	N	0000
0000	I	000365	NPAR	0000	I	000363	NPOI	0000	I	000364	N1	0000	R	000217	P	0000
0000	R	000000	R2M	0000	R	000147	T	0002	R	000000	U	0000	R	000405	V	0000
0002	R	000310	Y	0000	R	000402	Y1									

```
00101     1*          DIMENSION R2M(10,10),ALPHA(3)
00103     2*          DIMENSION T(10),FI(30),P(10,10)
00104     3*          COMMON U(200),Y(200)
00105     4*          EXTERNAL F
00106     5*          NPOI=200
00107     6*          CALL INPUT(U,Y,NPOI)
00110     7*          N1=1
00111     8*          NPAR=2*N1
00112     9*          NN=2*N1
00113    10*          ALPHA(1)=-3.0
00114    11*          ALPHA(2)=-3.0
00115    12*          ALPHA(3)=-7.0
00116    13*          MSEC=180
00117    14*          CALL GFLPW(F,DUMMY,ALPHA,3,1.0E-6,-1.0E+6,1.0E-6,1.0E-3,2453,MSEC)
00120    15*          JPKT=10
00121    16*          R2M(1,1)=10.**ALPHA(1)
00122    17*          R2M(2,2)=10.**ALPHA(2)
00123    18*          R2M(1,2)=10.**ALPHA(3)
00124    19*          R2M(2,1)=R2M(1,2)
```

```
00125    20*          PRINT 303
00127    21*          PRINT 304,((R2M(I,J),J=1,NN),I=1,NN)
00140    22*          DO 5 I=1,NN
00143    23*          T(I)=0.0
00144    24*        5 FI(I)=0.0
00146    25*          DO 15 I=1,NN
00151    26*          DO 10 J=1,NN
00154    27*       10 P(I,J)=0.0
00156    28*       15 P(I,I)=100.
00160    29*          E1=0.0
00161    30*          E=0.
00162    31*          VI=0.
00163    32*          PRINT 308
00165    33*          DO 30 N=1,NPOI
00170    34*          K=NPAR+1
00171    35*          DO 20 L=2,NPAR
00174    36*          K=K-1
00175    37*       20 FI(K)=FI(K-1)
00177    38*          FI(1)=U(N)
00200    39*          Y1=Y(N)
00201    40*          CALL KALID(T,P,FI,Y1,NN,10,R2M,RES,DENOM)
00202    41*          E1=E1+RES*RES
00203    42*          E=E+RES*RES/DENOM
00204    43*          VI=ALOG(DENOM)+VI
00205    44*          V=VI+N*(ALOG(E/N)+1.)
00206    45*          IF (N-25)24,23,24
00211    46*       23 E1=0.
00212    47*          E=0.
00213    48*          V=0.
00214    49*       24 IF(N-JPKT) 30,25,25
00217    50*       25 JPKT=JPKT+10
00220    51*          PRINT 309, N,E,E1,V,(T(M),M=1,NN)
00232    52*       30 CONTINUE
00234    53*          STOP
00235    54*      303 FORMAT(1H1,10X,43HR2M IS COVARIANCE MATRIX OF PARAMETER NOISE,//)
00236    55*      304 FORMAT(11X,2F10.5,/)
00237    56*      308 FORMAT(1H0,6X,77HTIME            E              E1              V
00237    57*         *          T1              T2,//)
00240    58*      309 FORMAT(6X,I5,3E15.6,2F15.4)
00241    59*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    1 SEC.
PHASE 2 TIME =    0 SEC.
```

```
0005    NIO2$
0006    SIN
0007    EXP
0010    NIO1$
0011    NERR3$


 STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

 0000    001133 100F        0000    001153 102F        0001    000027 111G        0001    000075 122G        0001
 0000 R 000310 A            0000 R 000620 B            0000 R 000000 E            0000 I 001131 I            0000
 0000 I 001130 IRAN         0000 R 001132 RND


00101     1*          SUBROUTINE INPUT(U,Y,NPOI)
00103     2*          DIMENSION U(200),Y(200),E(200),A(200),B(200)
00104     3*          NPOI=200
00105     4*          PRINT 100
00107     5*          IRAN=19
00110     6*          DO 10 I=1,NPOI
00113     7*          CALL MCNODI(IRAN,RND)
00114     8*          E(I)=1.225*RND
00115     9*       10 CONTINUE
00117    10*          U(1)=5.*(1.+SIN(0.015))+0.5*SIN(0.5)
00120    11*          Y(1)=2.*EXP(0.9-0.2*U(1))*U(1)+E(1)
00121    12*          DO 50 I=2,NPOI
00124    13*          U(I)=5.*(1.+SIN(0.015*I))+0.5*SIN(0.5*I)
00125    14*          Y(I)=2.*EXP(0.9-0.2*U(I))*U(I)+(0.5+2.*EXP(0.7-0.15*U(I)))*U(I-1)+
00125    15*         *E(I)
00126    16*          A(I)=2.*EXP(0.9-0.2*U(I))
00127    17*          B(I)= 0.5+2.*EXP(0.7-0.15*U(I))
00130    18*       50 CONTINUE
00132    19*          PRINT 102,(I,U(I),Y(I),E(I),A(I),B(I),I=1,200)
00145    20*          RETURN
00146    21*      100 FORMAT(1H1,5X,79HTIME          INPUT          OUTPUT          NOISE
00146    22*         *          A                B   ,/)
00147    23*      102 FORMAT(5X,I5,5F15.4)
00150    24*          END

      END OF UNIVAC 1108 FORTRAN V COMPILATION.     0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    1 SEC.
PHASE 2 TIME =    0 SEC.
```

```
0004    NEXP2$
0005    ALOG
0006    NERR3$
```

STORAGE ASSIGNMENT FOR VARIABLES (BLOCK, TYPE, RELATIVE LOCATION, NAME)

```
0001    000026 111G      0001    000034 117G      0001    000035 122G      0001    000100 140G      0001
0001    000173 30L       0000 R  000376 DENOM     0000 R  000367 E         0000 R  000366 E1        0000
0000 R  000013 FI        0000 I  000364 I         0000    000406 INJP$     0000 I  000365 J         0000
0000 I  000373 L         0000 I  000371 N         0000 I  000363 NN        0000 I  000362 NPAR      0000
0000 R  000051 P         0000 R  000375 RES       0000 R  000215 R1M       0000 R  000001 T         0002
0000 R  000377 V         0000 R  000370 VI        0002 R  000310 Y         0000 R  000374 Y1
```

```
00101     1*          FUNCTION F(ALPHA,NARG)
00103     2*          DIMENSION T(10),FI(30),P(10,10),R1M(10,10),ALPHA(3)
00104     3*          COMMON U(200),Y(200)
00105     4*          NPOI=200
00106     5*          NPAR=2
00107     6*          NN=2
00110     7*          DO 5 I=1,NN
00113     8*          T(I)=0.0
00114     9*        5 FI(I)=0.0
00116    10*          DO 15 I=1,NN
00121    11*          DO 10 J=1,NN
00124    12*       10 P(I,J)=0.0
00126    13*       15 P(I,I)=100.
00130    14*          E1=0.0
00131    15*          E=0.
00132    16*          VI=0.
00133    17*          R1M(1,1)=10**ALPHA(1)
00134    18*          R1M(2,2)=10**ALPHA(2)
00135    19*          R1M(1,2)=10**ALPHA(3)
00136    20*          R1M(2,1)=R1M(1,2)
00137    21*          DO 30 N=1,NPOI
00142    22*          K=NPAR+1
00143    23*          DO 20 L=2,NPAR
00146    24*          K=K-1
00147    25*       20 FI(K)=FI(K-1)
00151    26*          FI(1)=U(N)
00152    27*          Y1=Y(N)
00153    28*          CALL KALID(T,P,FI,Y1,NN,10,R1M,RES,DENOM)
```

```
00154     29*           E1=E1+RES*RES
00155     30*           E=E+RES*RES/DENOM
00156     31*           VI=ALOG(DENOM)+VI
00157     32*           V=VI+N*(ALOG(E/N)+1.)
00160     33*           IF (N-25)30,23,30
00163     34*        23 E1=0.
00164     35*           E=0.
00165     36*           V=0.
00166     37*        30 CONTINUE
00170     38*           F=V
00171     39*           RETURN
00172     40*           END

        END OF UNIVAC 1108 FORTRAN V COMPILATION.      0 *DIAGNOSTIC* MESSAGE(S)
PHASE 1 TIME =    0 SEC.
PHASE 2 TIME =    0 SEC.
PHASE 3 TIME =    1 SEC.
PHASE 4 TIME =    0 SEC.
PHASE 5 TIME =    0 SEC.
PHASE 6 TIME =    0 SEC.


TOTAL COMPILATION TIME =    1   SEC
```

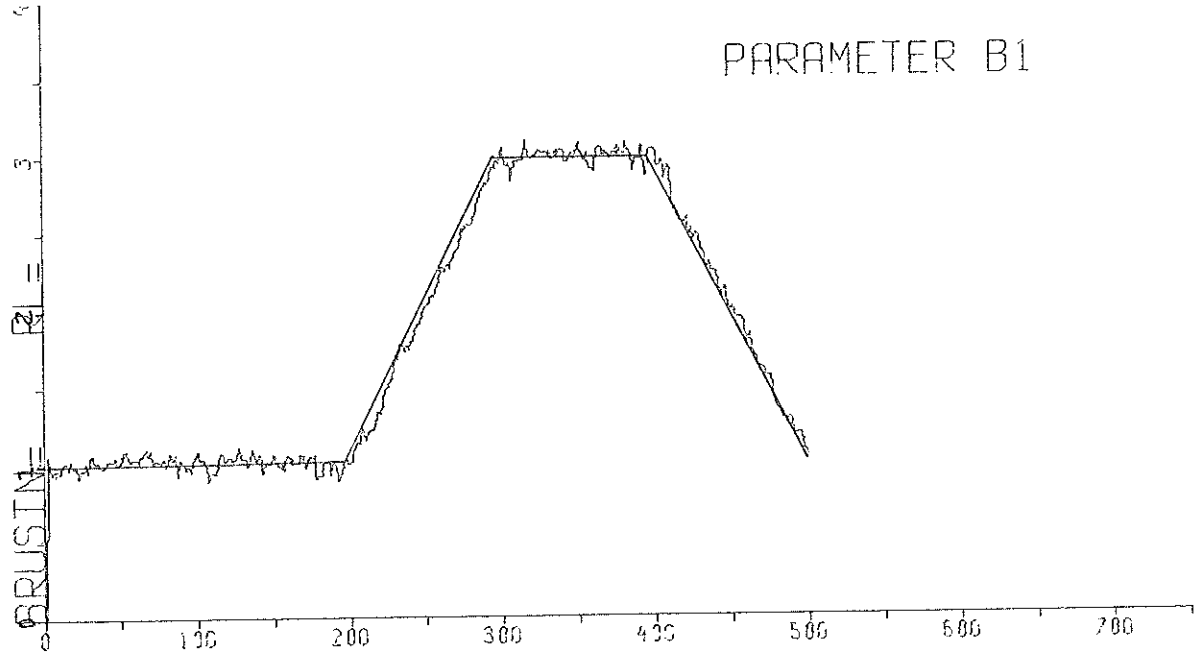
```
      SUBROUTINE KALID(T,P,FI,NN,NMAX,R1M,RES,DENOM)
C     REAL-TIME IDENTIFICATION USING KALMAN THEORY. SUBROUTINE UP-
C     DATES ARGUMENTS T,P,RES AND DENOM.
C     REFERENCE NONE
C     AUTHOR JOHAN WIESLANDER 21/08 1969
C     T IS VECTOR CONTAINING ESTIMATED SYSTEM PARAMETERS. FI CON-
C     TAINS OLD INPUT AND OUTPUT VALUES. FI SHOULD BE UPDATED OUT-
C     SIDE THE ROUTINE.
C     P IS COVARIANCE MATRIX OF THE A-POSTERIORI ESTIMATION ERROR.
C     Y IS LAST OUTPUT FROM THE SYSTEM.
C     NN IS NUMBERS OF PARAMETERS.
C     NMAX IS DIMENSION PARAMETER. MAX = 10.
C     R1M IS COVARIANCE MATRIX OF PARAMETER NOISE.
C     R1M IS ASSUMED TO BE SYMMETRIC.
C     RES IS THE RESIDUAL.
C     DENOM IS THE FACTOR 1 + FI*P*FIT.
C     SUBROUTINES REQUIRED NONE.
      DIMENSION T(NMAX), P(NMAX,NMAX), FI(NMAX), R1M(NMAX,NMAX), RK(11)
      N = NN
      DO 5 I = 1,N
      SL = 0.
      DO 4 J = 1,N
    4 SL = SL + P(I,J)*FI(J)
    5 RK(I) = SL
      SL = 1.
      DO 10 I = 1,N
   10 SL = SL + FI(I)*RK(I)
      DENOM = SL
      DO 15 I = 1,N
   15 RK(I) = RK(I)/SL
      DO 20 I = 1,N
      DO 20 J = 1,N
      P(I,J) = P(I,J) - RK(I)*RK(J)*SL + R1M(I,J)
   20 P(J,I) = P(I,J)
      DO 30 J = 1,N
   30 SL = SL - FI(J)*T(J)
      DO 35 J = 1,N
   35 T(J) = T(J) + RK(J)*SL
      RES = SL
      RETURN
      END
```

```
      SUBROUTINE RTLS(T,P,FI,Y,NN,NMAX,RL,RES)
C     SUBROUTINE UPDATES ARGUMENTS T, P AND RES
C     T IS VECTOR CONTAINING ESTIMATED SYSTEMPARAMETERS
C     P CORRESPONDS TO THE INVERSE OF THE INF. MATRIX
C     FI CONTAINS OLD INPUT AND OUTPUT VALUES FI SHOULD BE UPDATED
C     OUTSIDE THE ROUTINE
C     Y IS LAST OUTPUT FROM THE SYSTEM
C     NN IS NUMBERS OF PARAMETERS
C     RL IS THE BASE OF THE EXPONENTIAL WEIGHTING FUNCTION
C     AUTHOR JOHAN WIESLANDER
C     SUBROUTINES REQUIRED NONE
      DIMENSION T(NMAX), P(NMAX,NMAX), FI(NMAX), RK(10), S(10)
      N = NN
      DO 5 I = 1,N
      SL = 0.
      DO 10 J = 1,N
   10 SL = SL + P(I,J)*FI(J)
      RK(I) = SL
    5 S(I) = SL
      SL = 1.
      DO 15 I = 1,N
   15 SL = SL + FI(I)*RK(I)
      SK = Y
      DO 25 I = 1,N
      RK(I) = RK(I)/SL
      DO 20 J = 1,I
      P(I,J) = ( P(I,J) - RK(I)*S(J) )/RL
   20 P(J,I) = P(I,J)
   25 SK = SK - FI(I)*T(I)
      DO 35 I = 1,N
   35 T(I) = T(I) + RK(I)*SK
      RES = SK
      RETURN
      END
```

```
      SUBROUTINE STAPP(T,FI,Y,N,NMAX,GAM,RES)
         DIMENSION T(NMAX), FI(NMAX)
         SK = GAM
         SL = Y
         DO 5 I = 1,N
         SK = SK + FI(I)*FI(I)
       5 SL = SL - FI(I)*T(I)
         DO 10 I = 1,N
      10 T(I) = T(I) + FI(I)*GAM*SL/SK
         RES = SL
         RETURN
         END
```

APPENDIX C

FIGURER OCH DIAGRAM



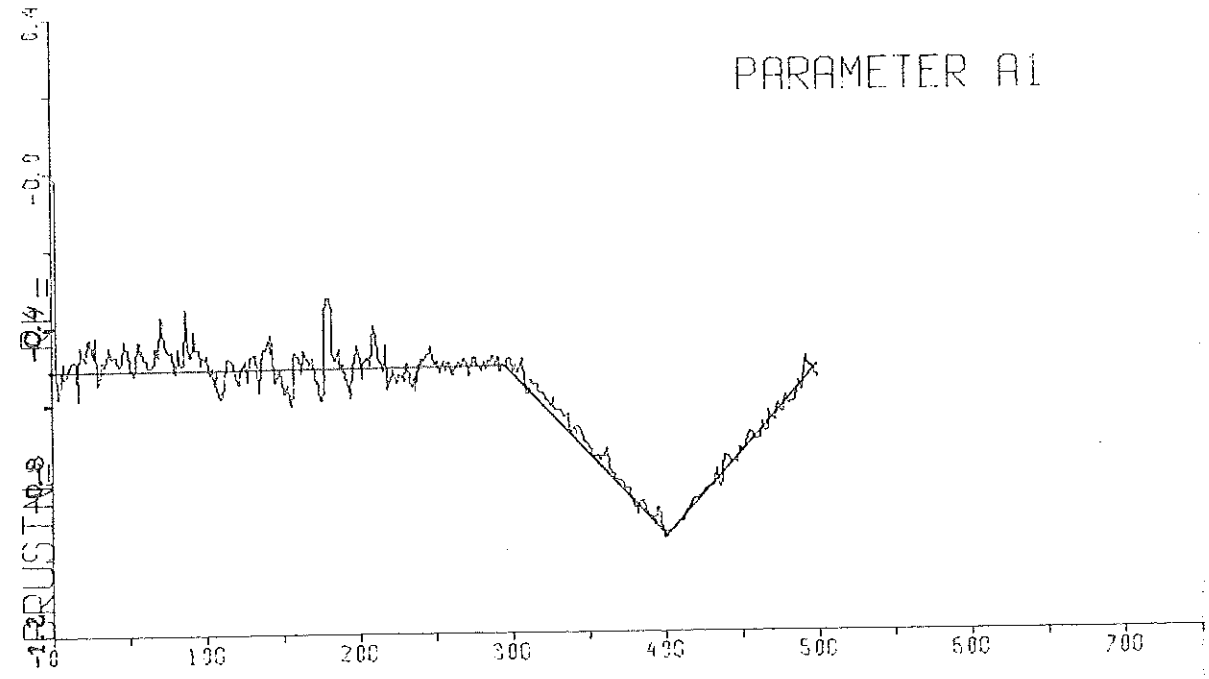
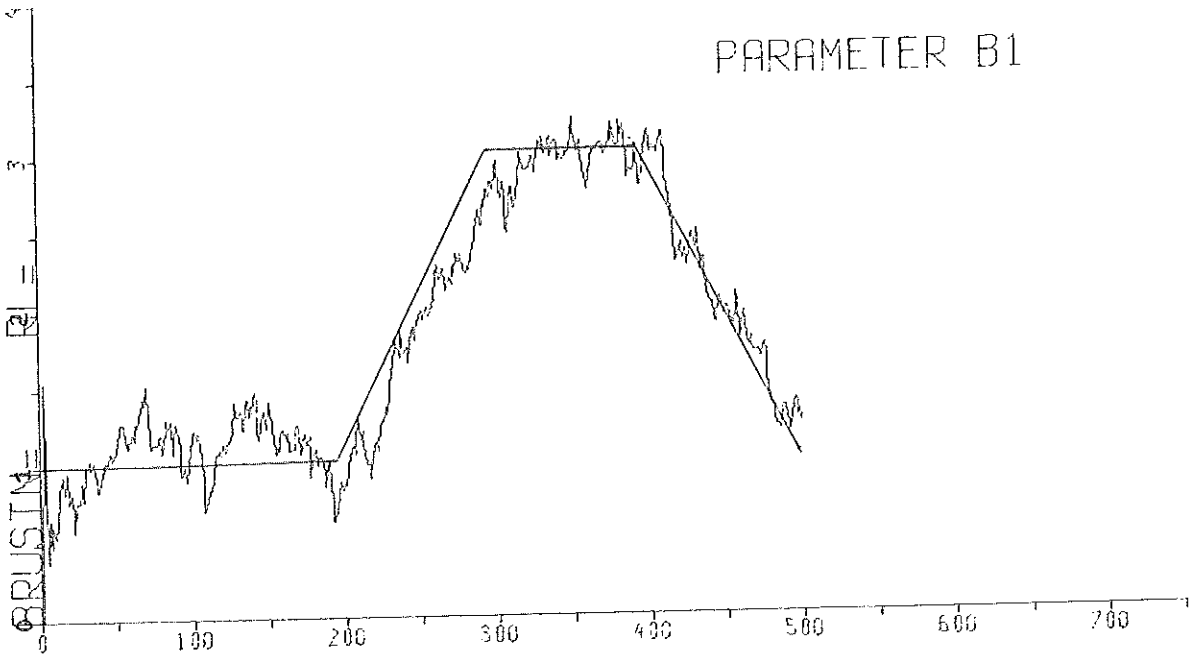


Fig.1 Reelltidsidentifiering enligt minsta-kvadrat metoden k = 0.1



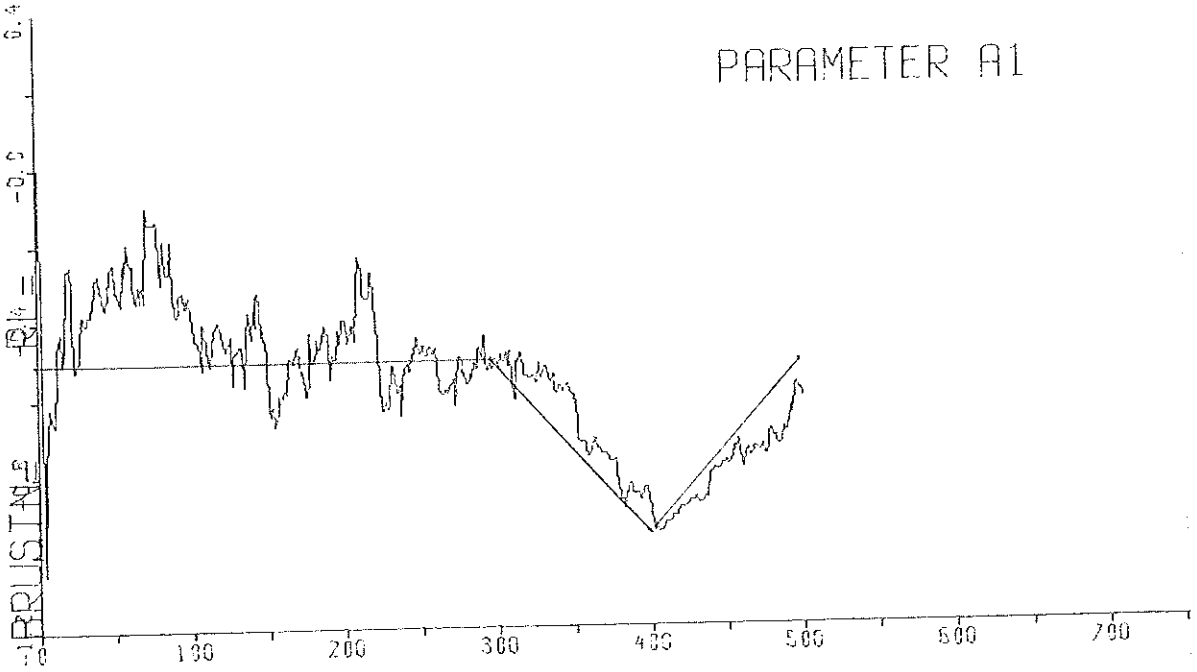
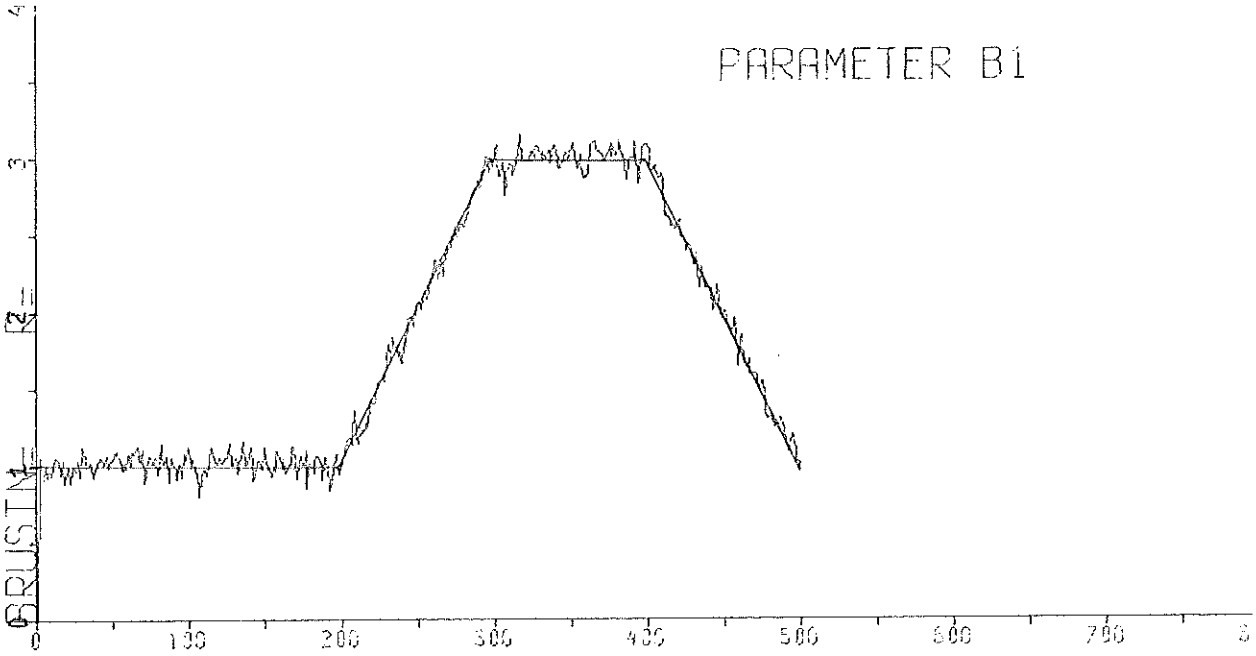


Fig.2 Reelltidsidentifiering enligt minsta-kvadrat metoden med k = 1.0



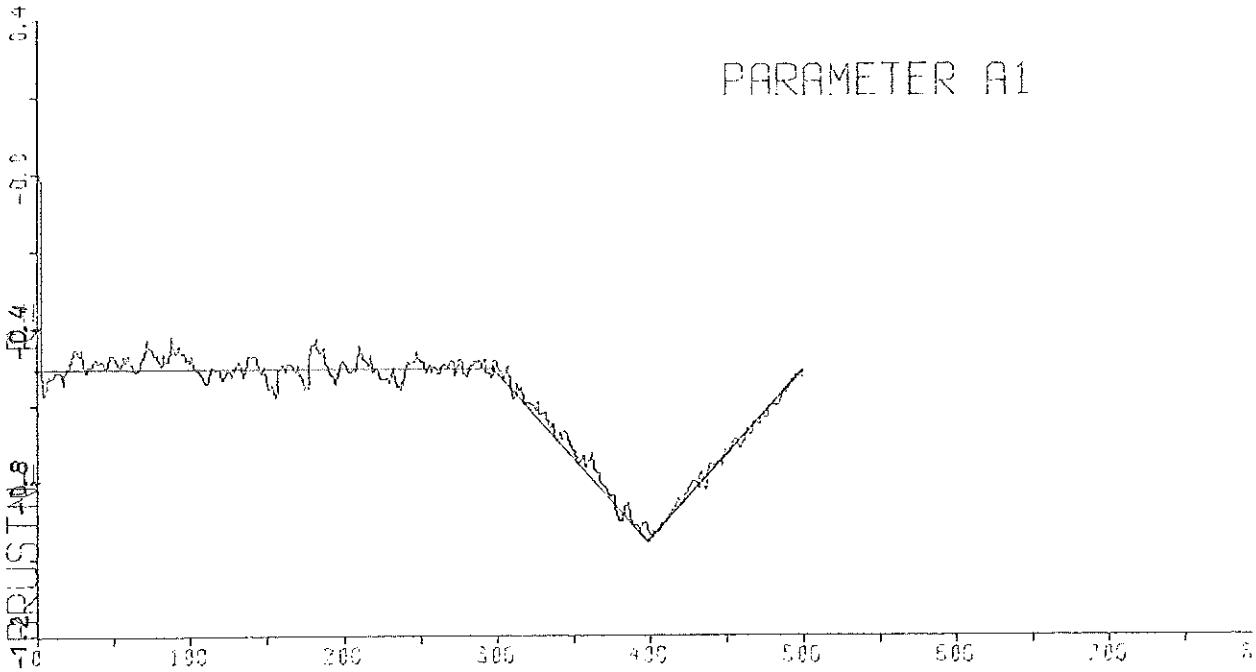
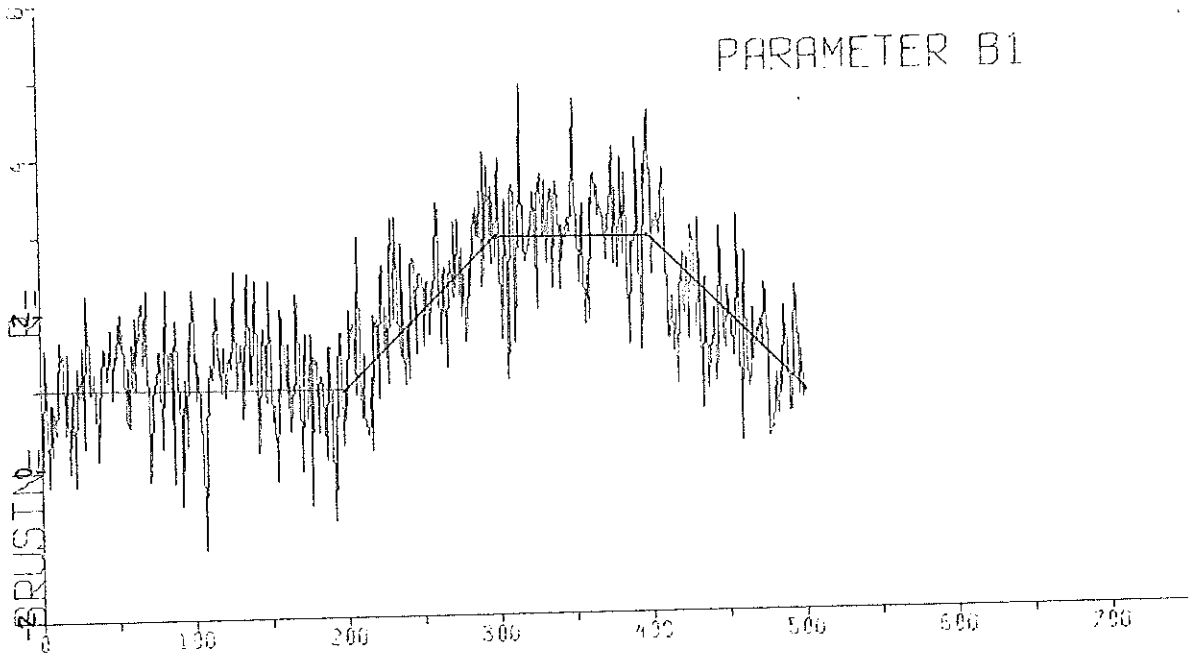


Fig. 3 Reelltidsidentifiering enligt KALID
för k = 0.1



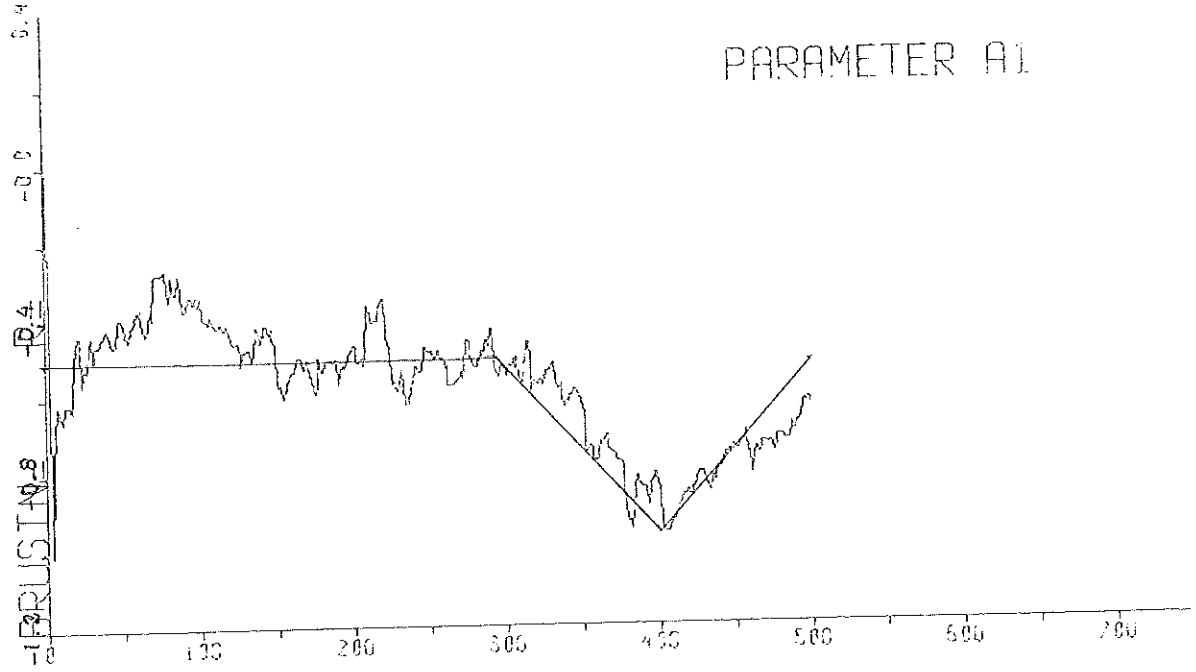
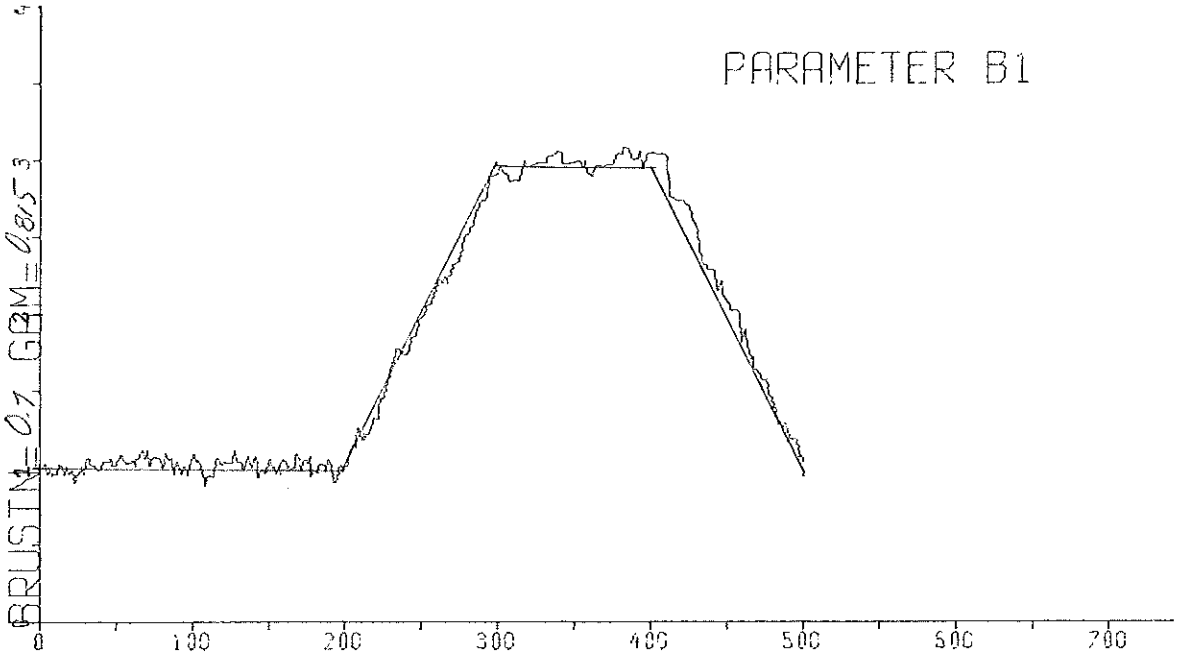


Fig. 4 Reelltidsidentifiering enligt KALID
med k = 1.0



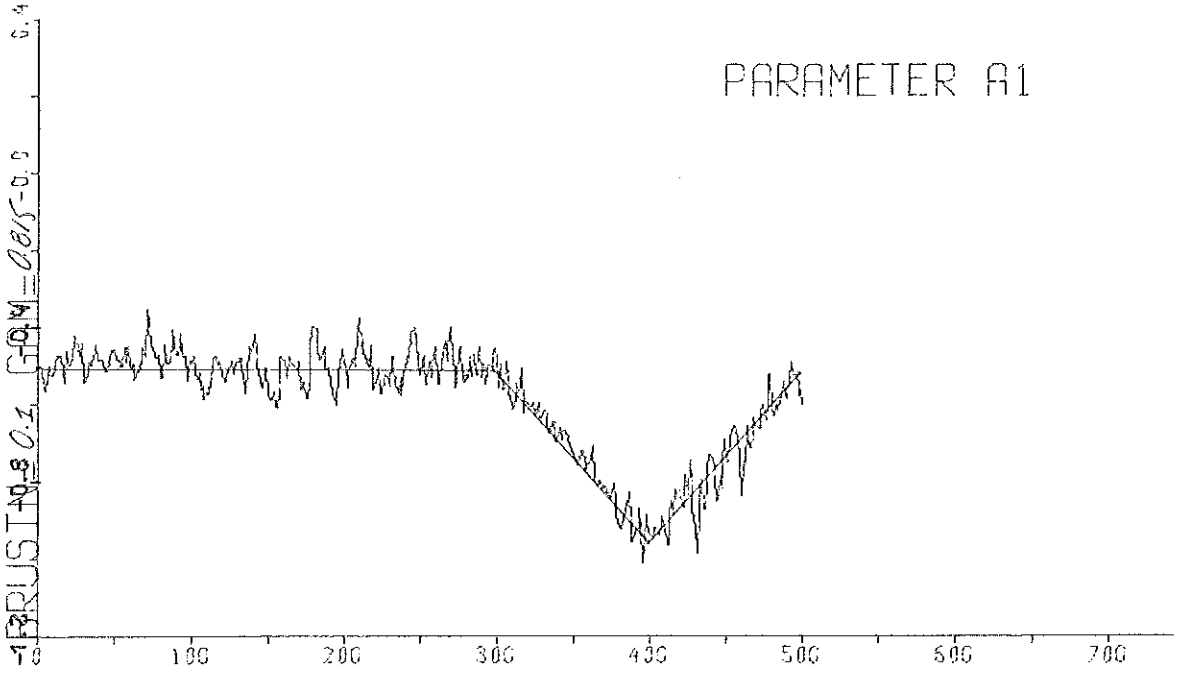
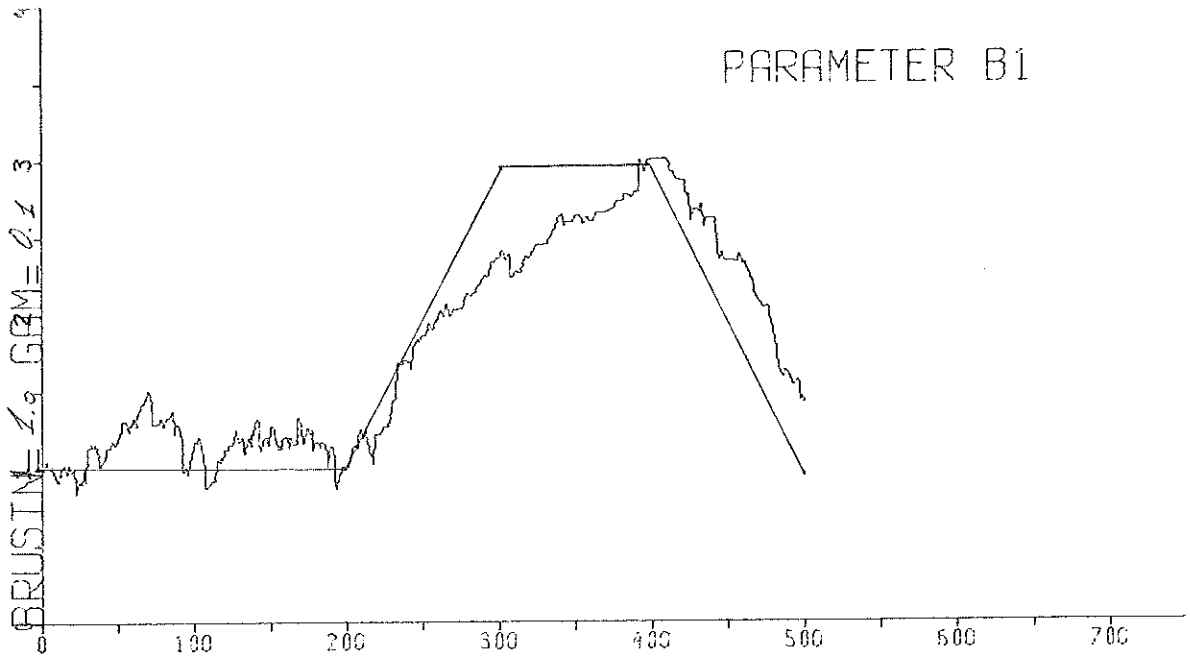


Fig. 5 Reelltidsidentifiering enligt stokastisk approximationsmetoden med k = 0.1



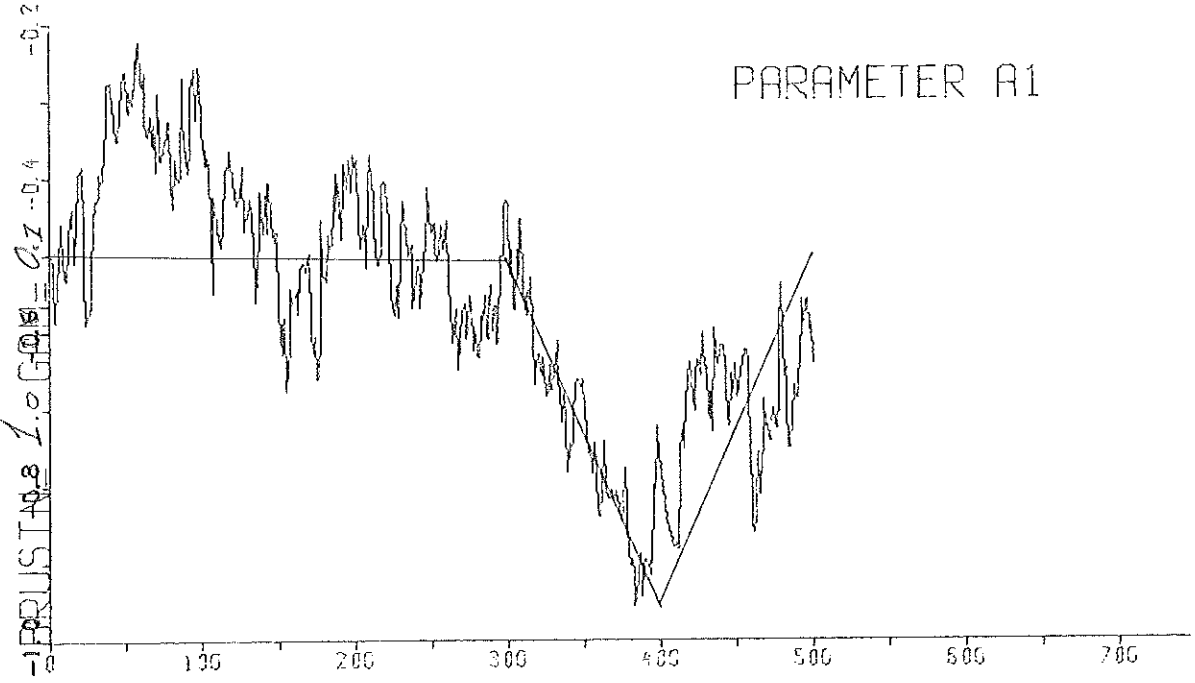


Fig. 6 Reelltidsidentifiering enligt stokastisk approximationsmetoden med k = 1.0

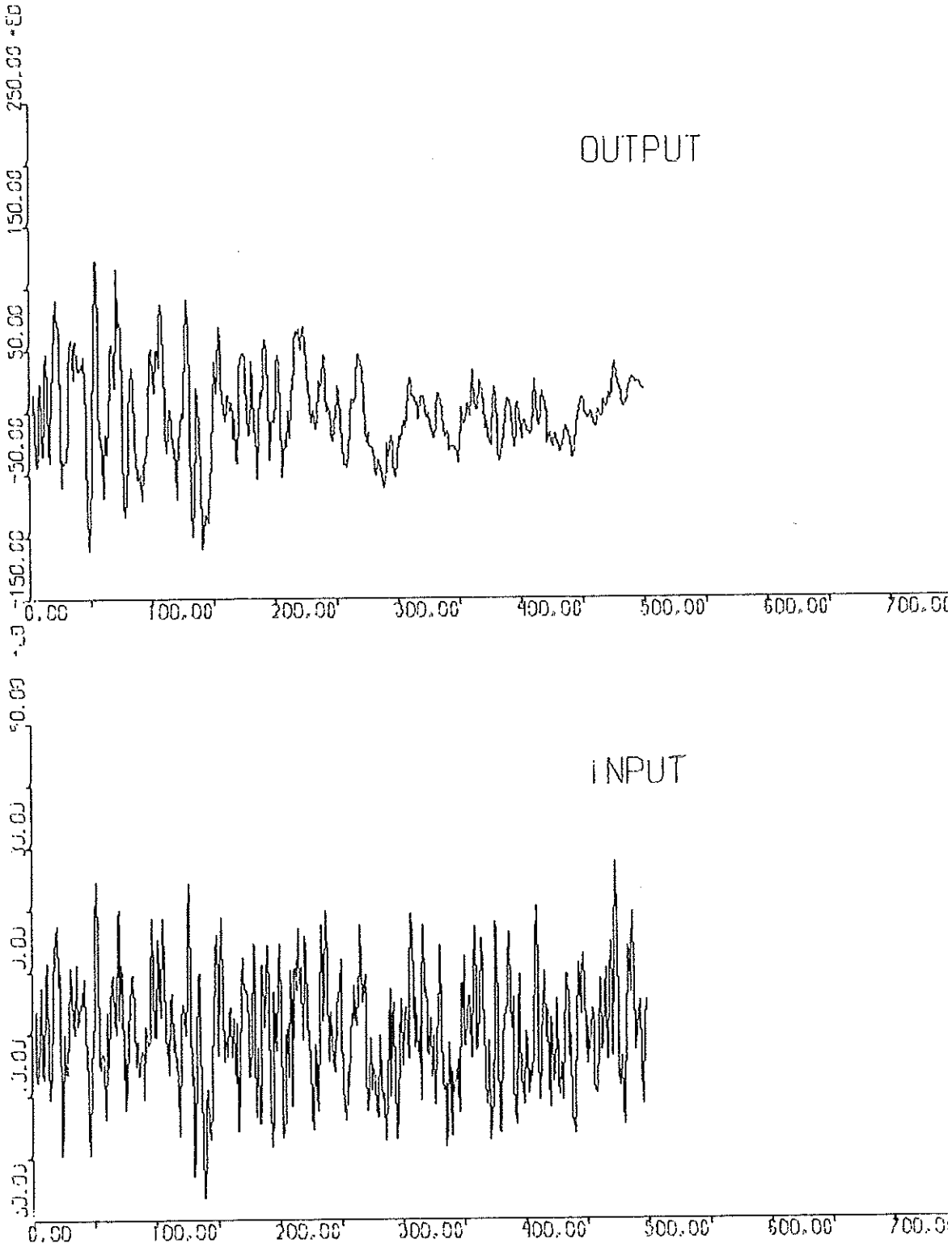
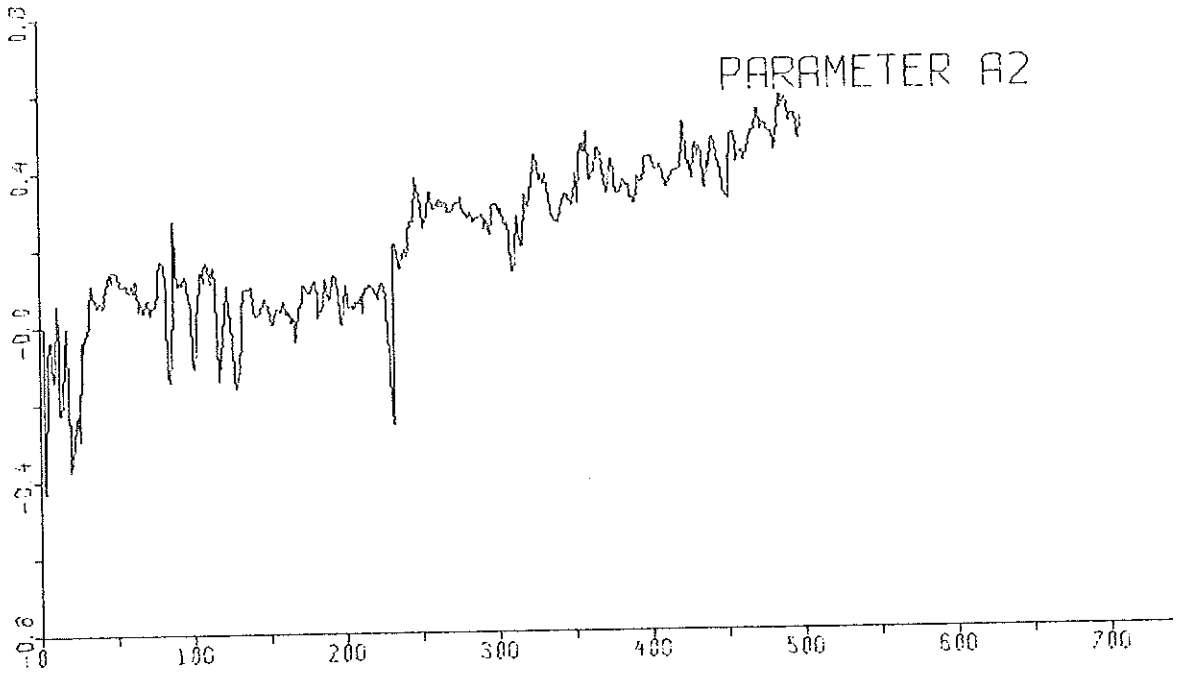


Fig. 7 In - och utsignaler från exempel 2



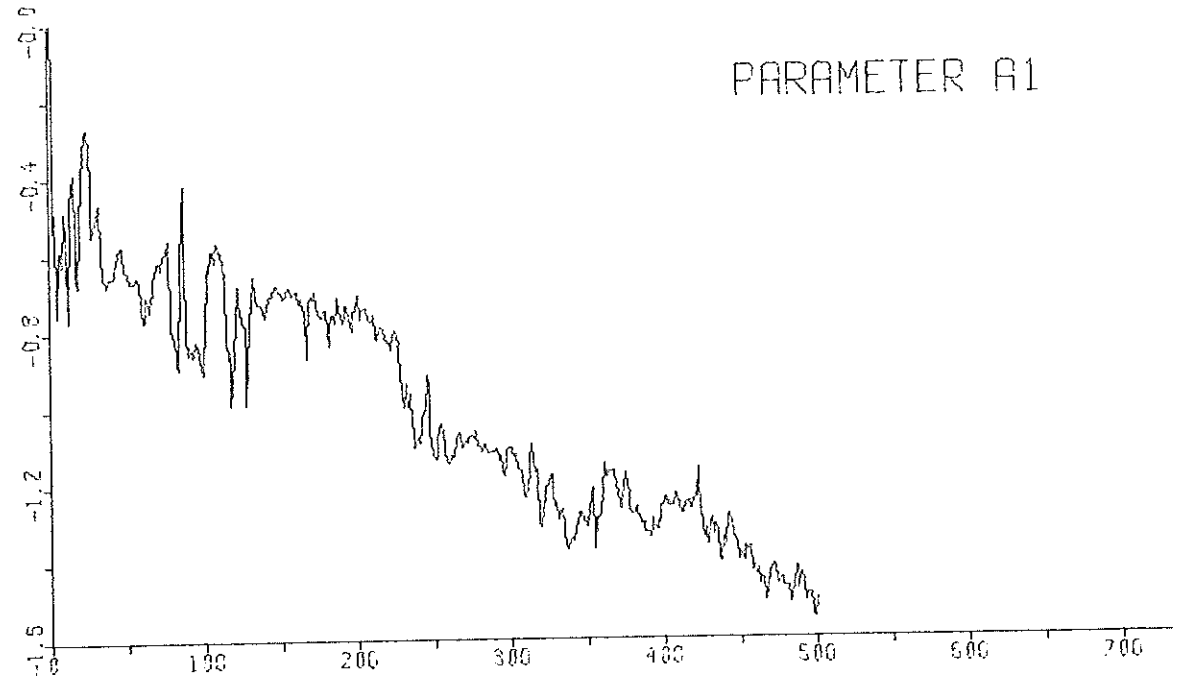
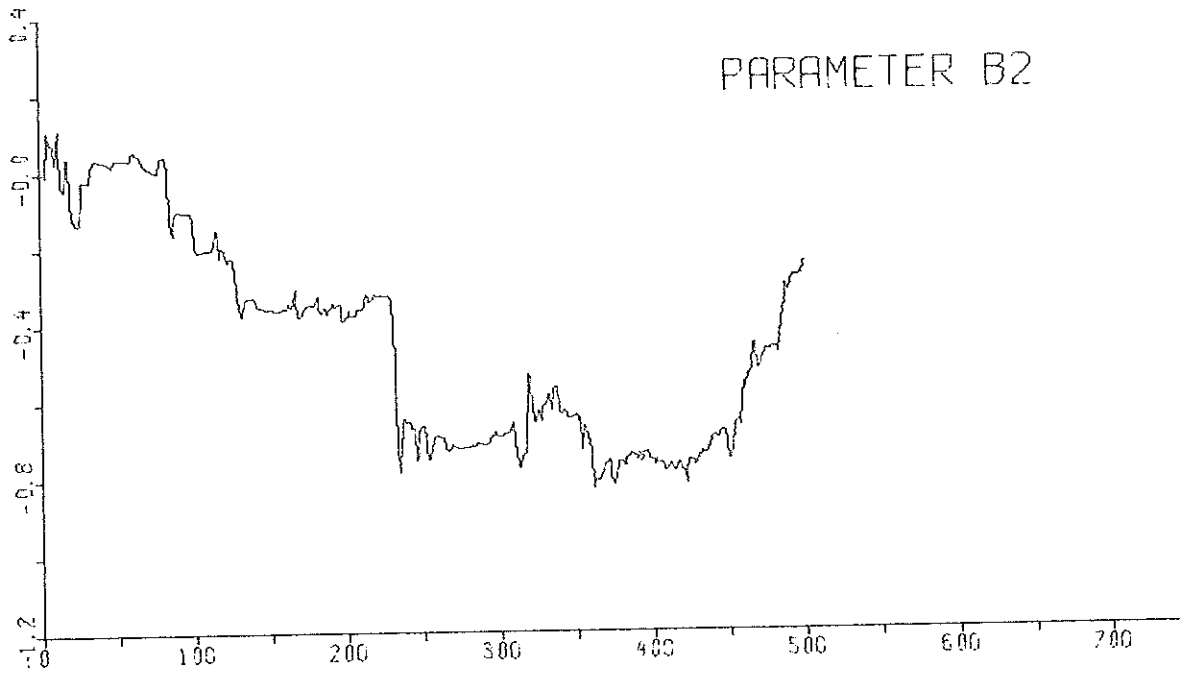


Fig. 8 Identifiering av systemet i exempel 2 enligt stokastisk approximationsmetoden med GAMZ = 0.5



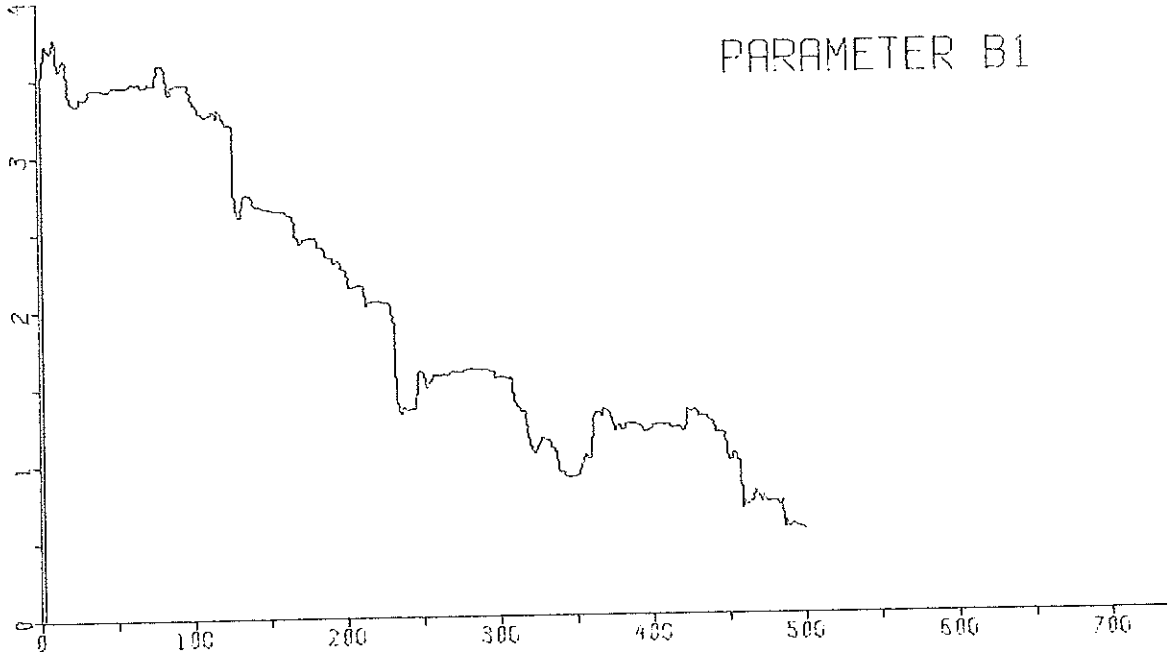
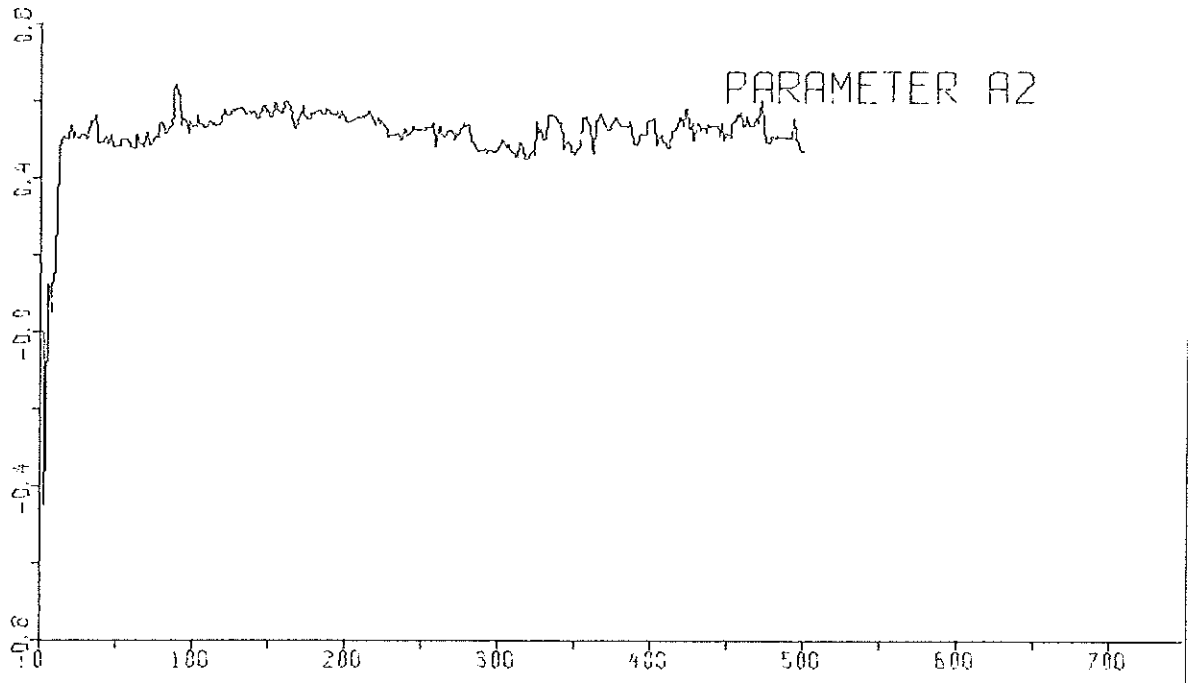


Fig. 9 Identifiering av systemet i exempel 2 enligt stokastisk approximationsmetoden med GAMZ = 0.5



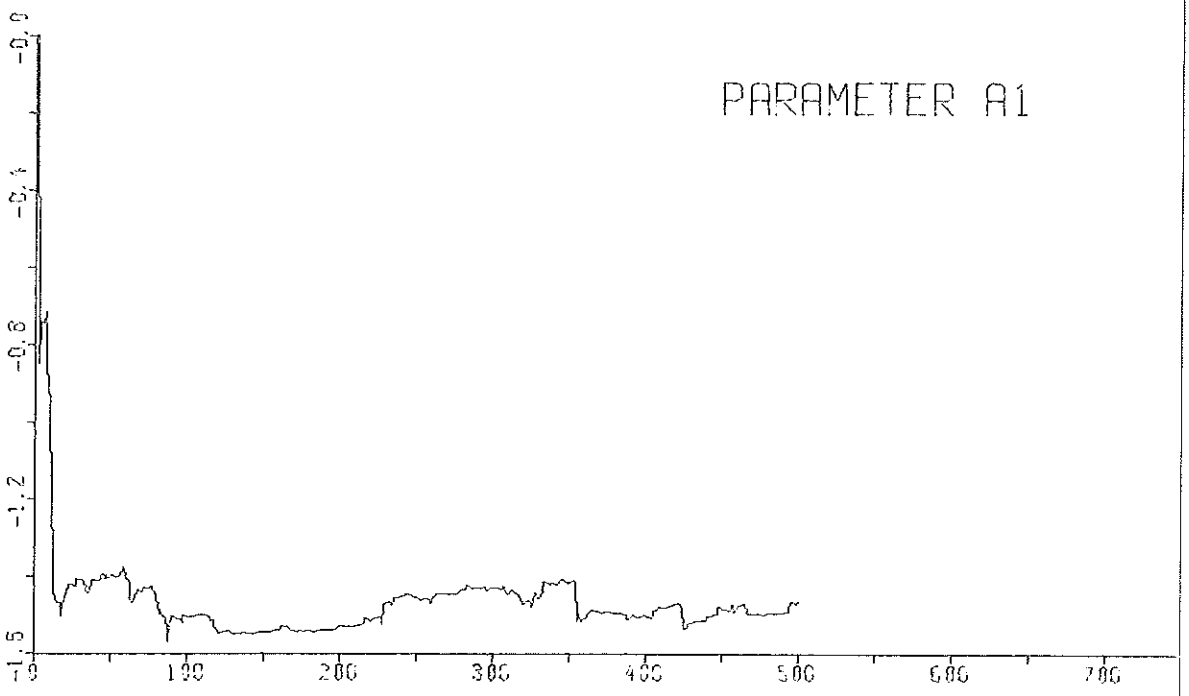
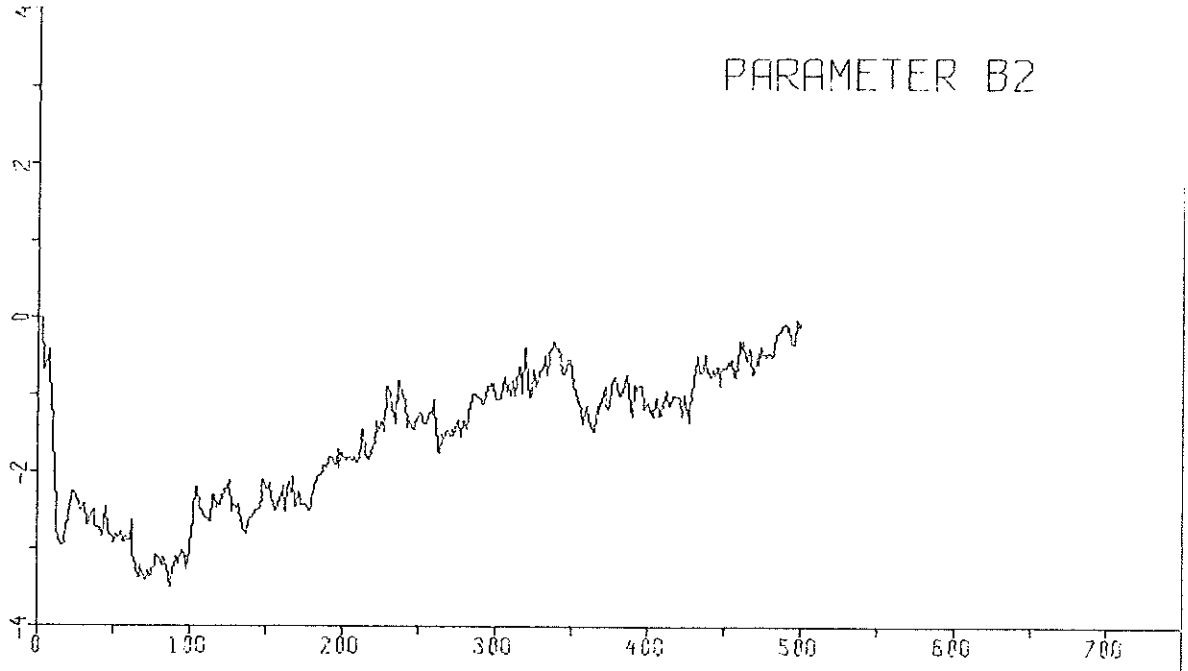


Fig. 10 Identifiering av systemet i exempel 2 användande Kalmanestimering. R1-matrisen ges av (5.1)



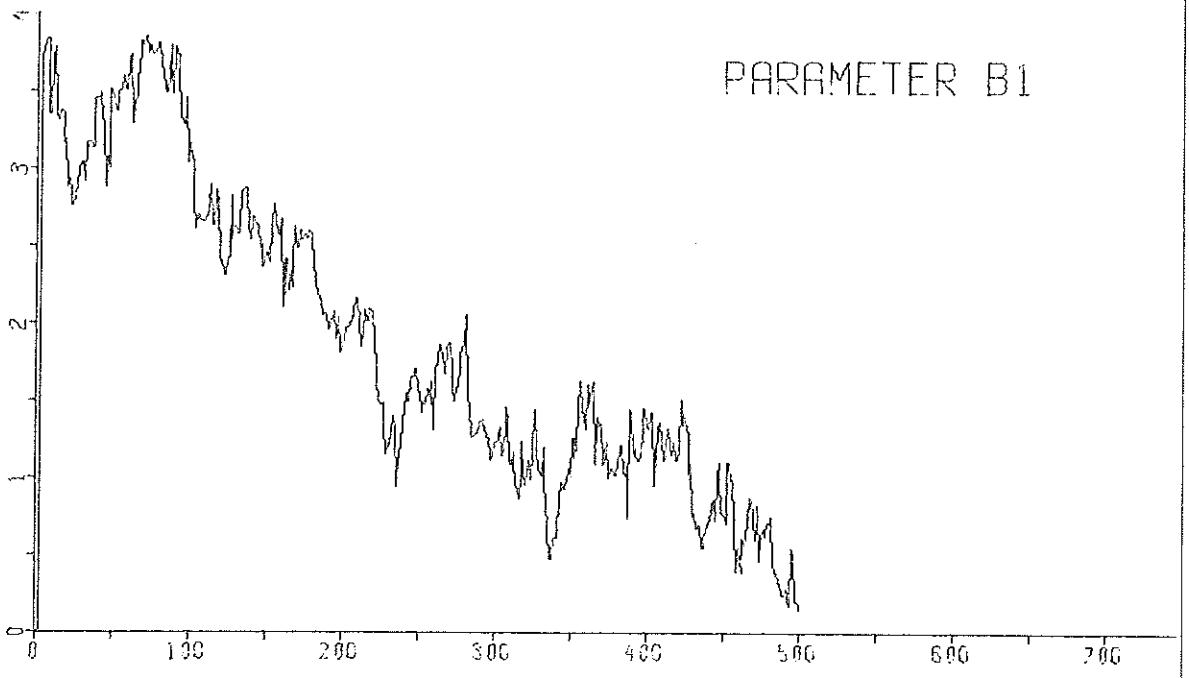
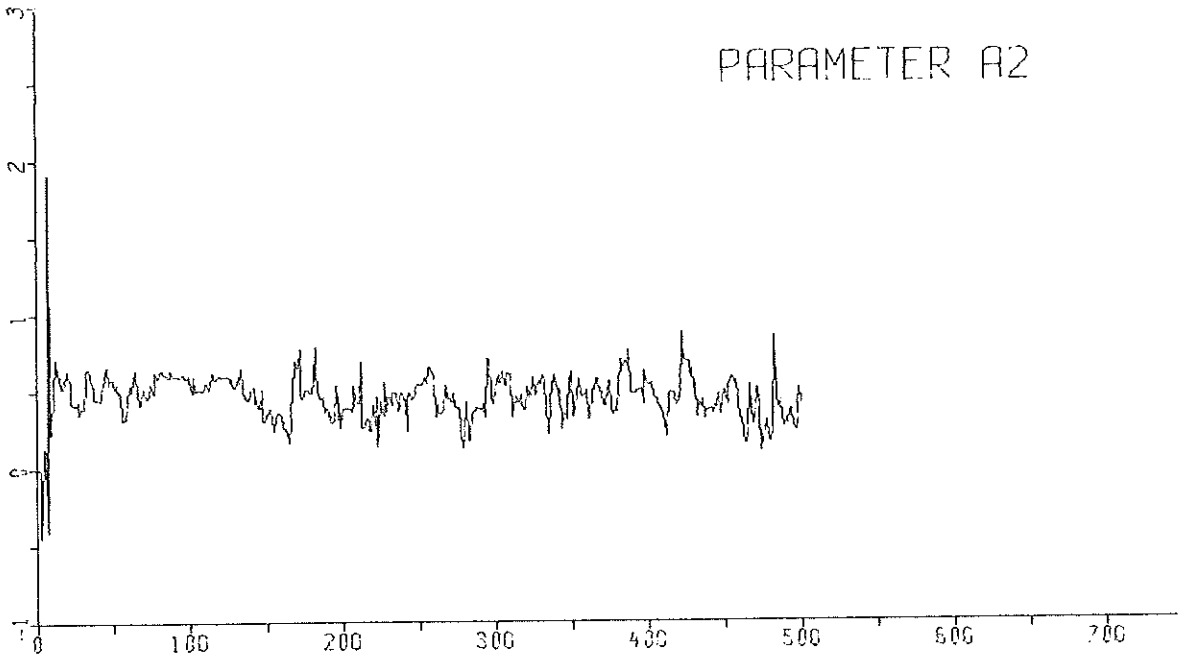


Fig. 11 Identifiering av systemet i exempel 2 användande Kalmanestimering. R1-matrisen ges av (5.1)



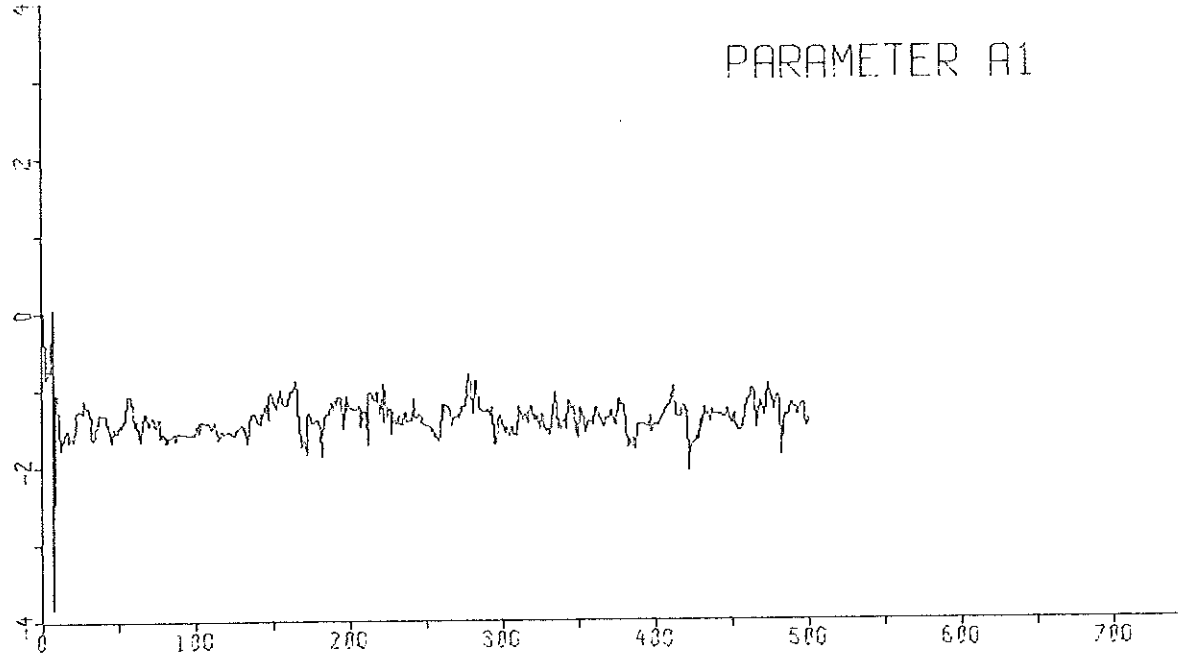
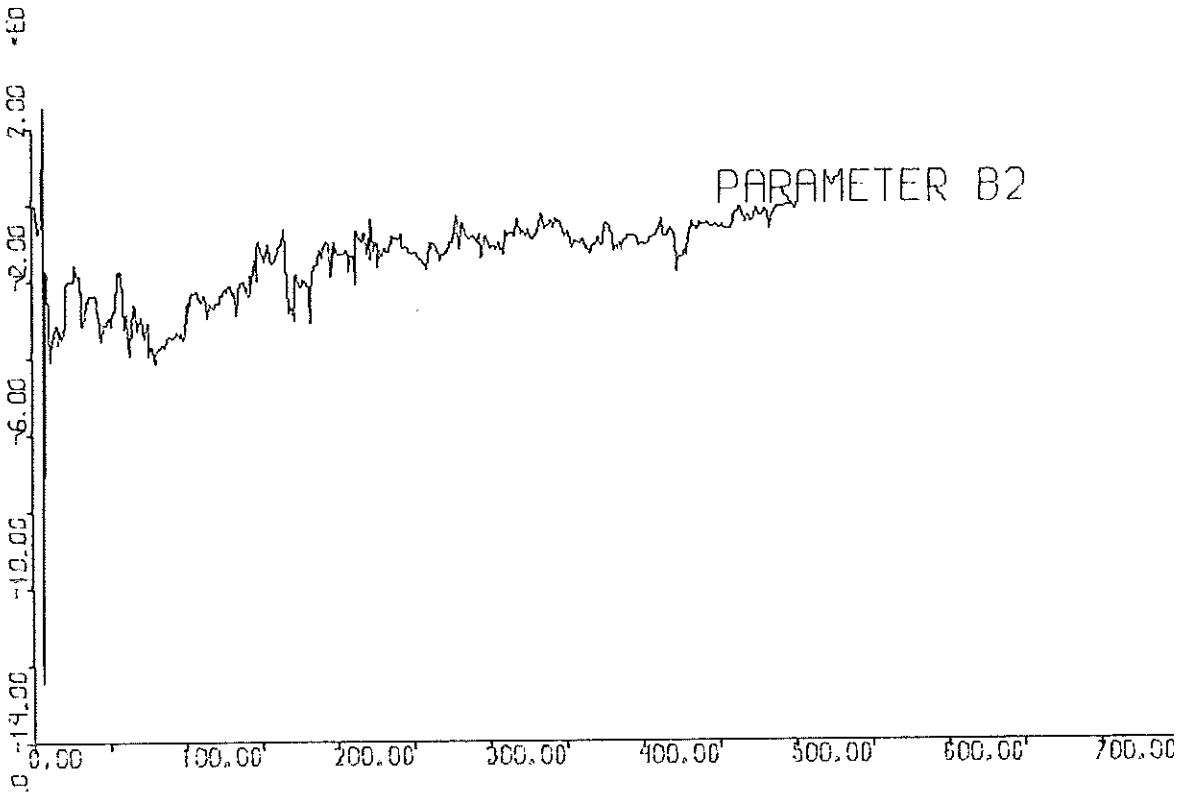


Fig. 12 Reelltidsidentifiering enligt minsta-kvadrat metoden av systemet i exempel 2 med $\lambda = 0.795$



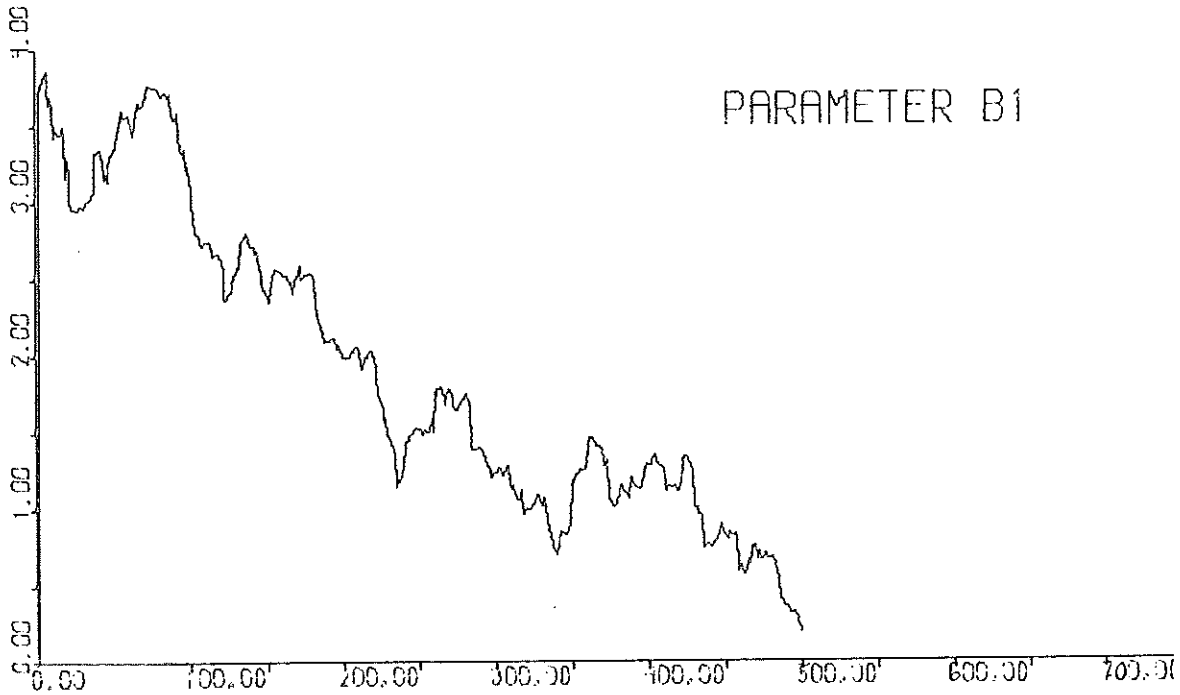


Fig. 13 Reelltidsidentifiering enligt minsta-kvadrat metoden av systemet i exempel 2 med $\lambda = 0.795$

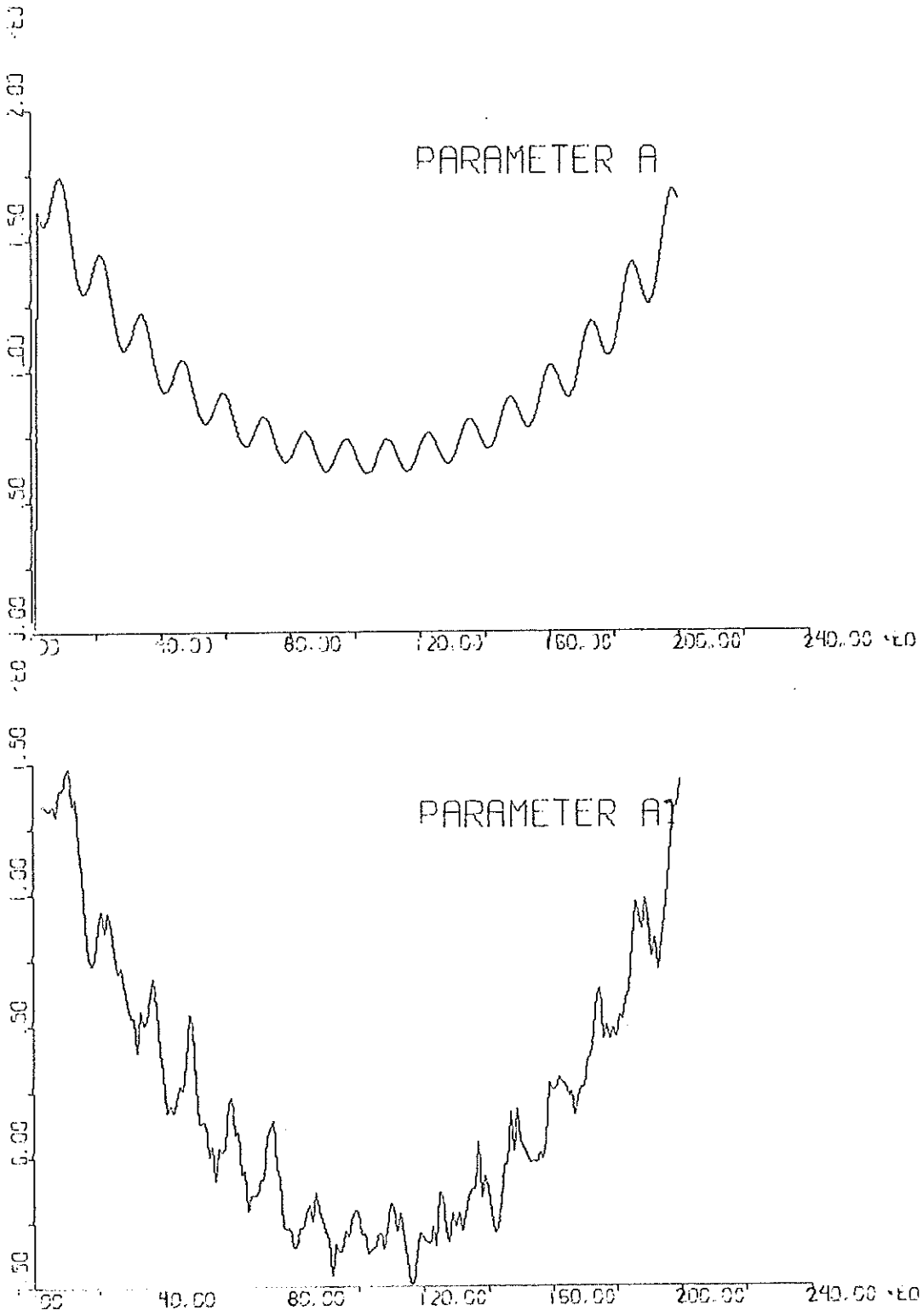


Fig. 14 Skattningen (underst) och det verkliga värdet (överst) av parametern a i exempel 3. R1-matrisen ges av (5.2)

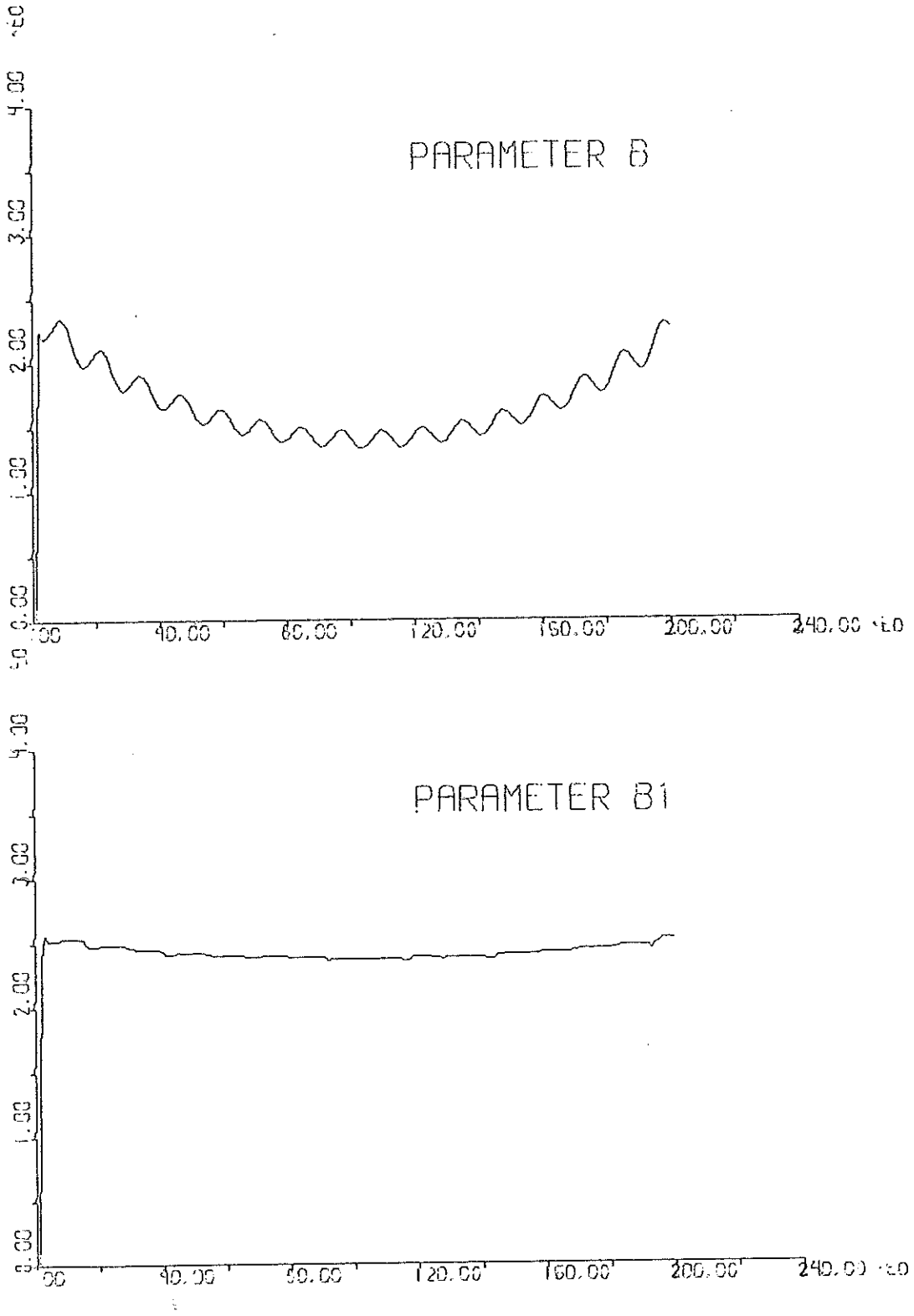


Fig. 15 Skattningen (underst) och det verkliga värdet (överst) av parametern b i exempel 3. R1-matrisen ges av (5.2)

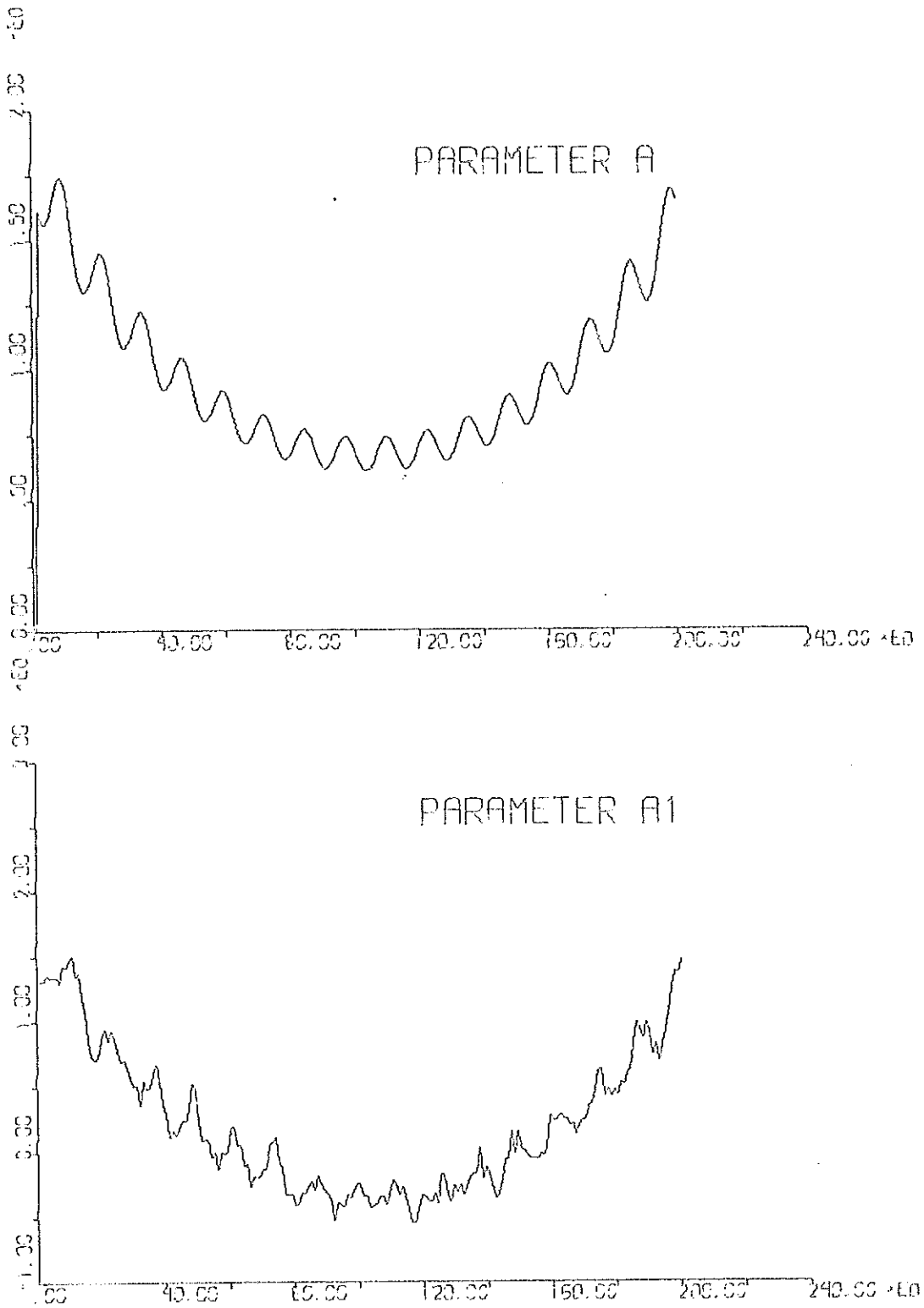


Fig. 16 Skattningen (underst) och det verkliga värdet (överst) av parametern a i exempel 3.
R1-matrisen ges av (5.3)

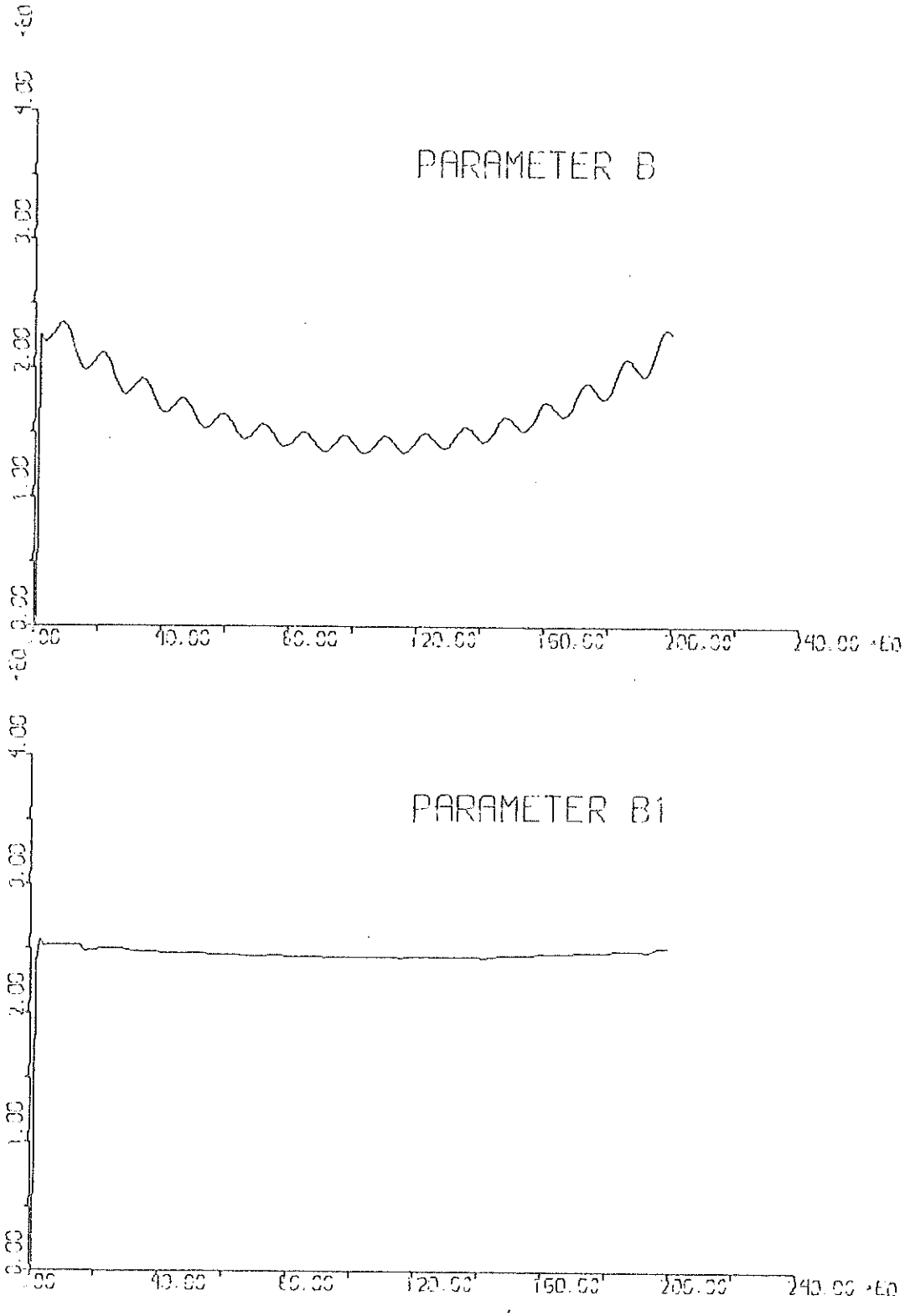
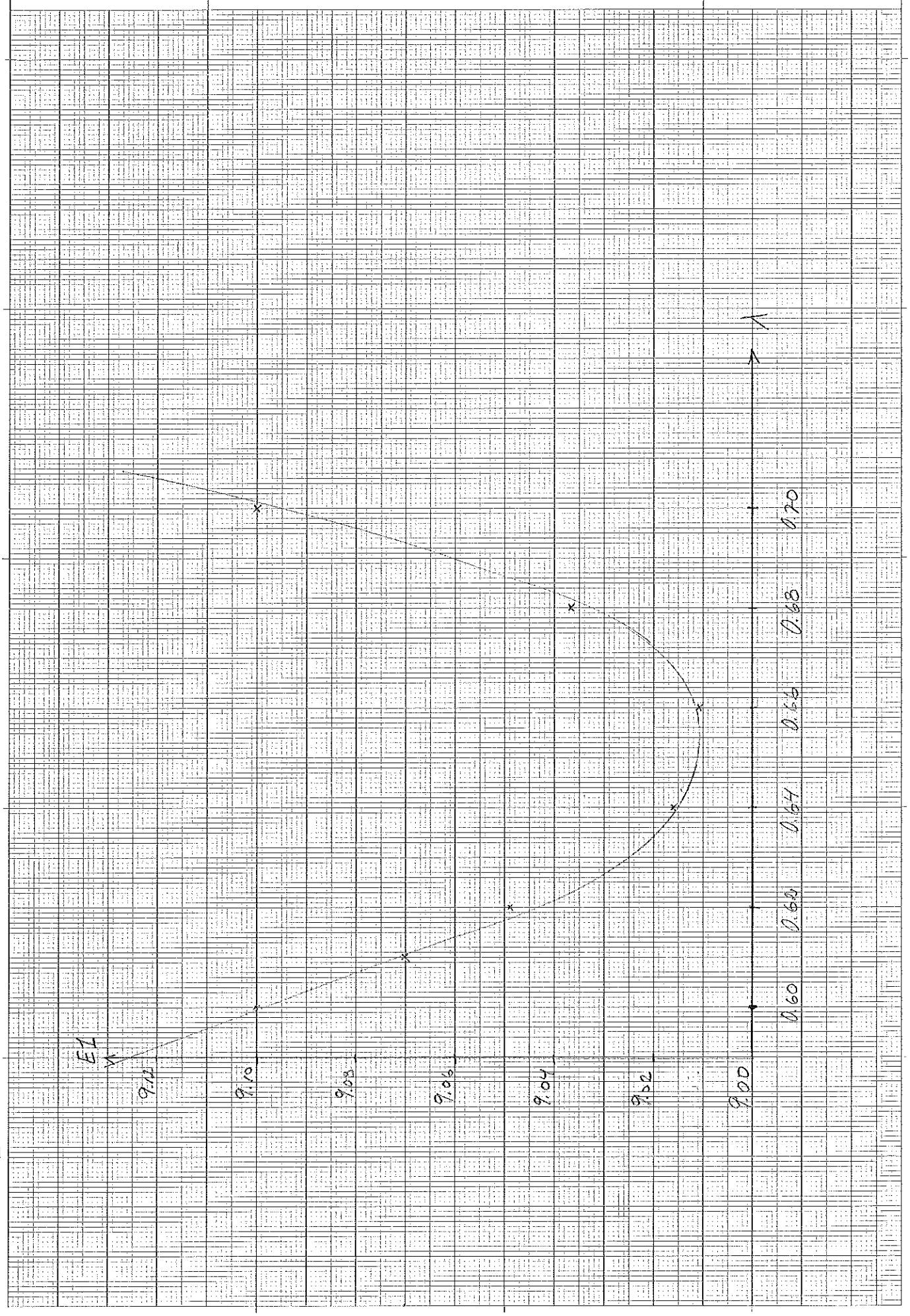


Fig. 17 Skattningen (underst) och det verkliga värdet (överst) av parametern b i exempel 3. R1-matrisen ges av (5.3)

Diagram 1
RTLS
k = 0.1



514 A4
SIS 73 25 01
TULLBERG KLIPPAN
Nr 1624

Diagram 2

RTLS

k = 1.0

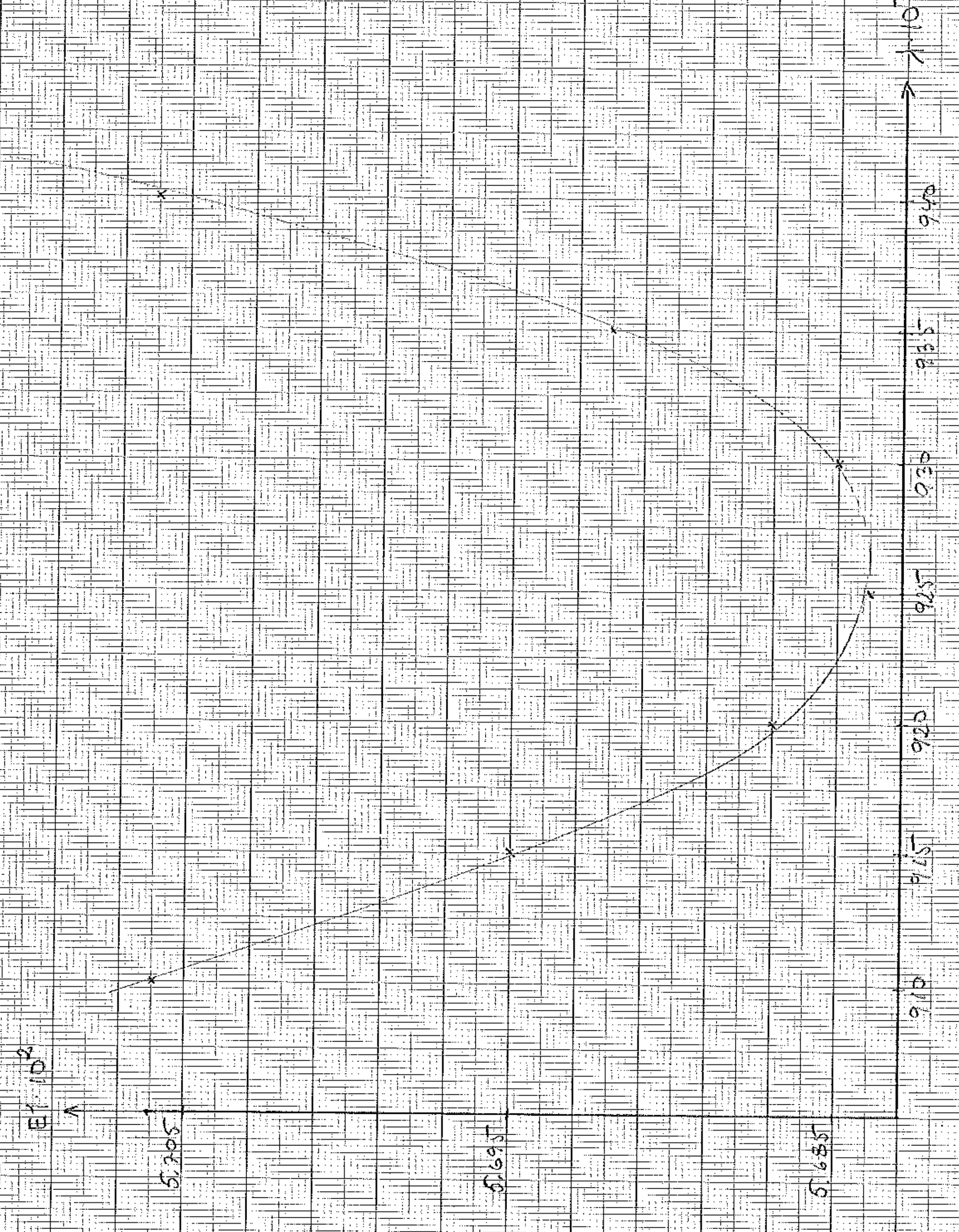


Diagram 3

KALID

k = 0.1

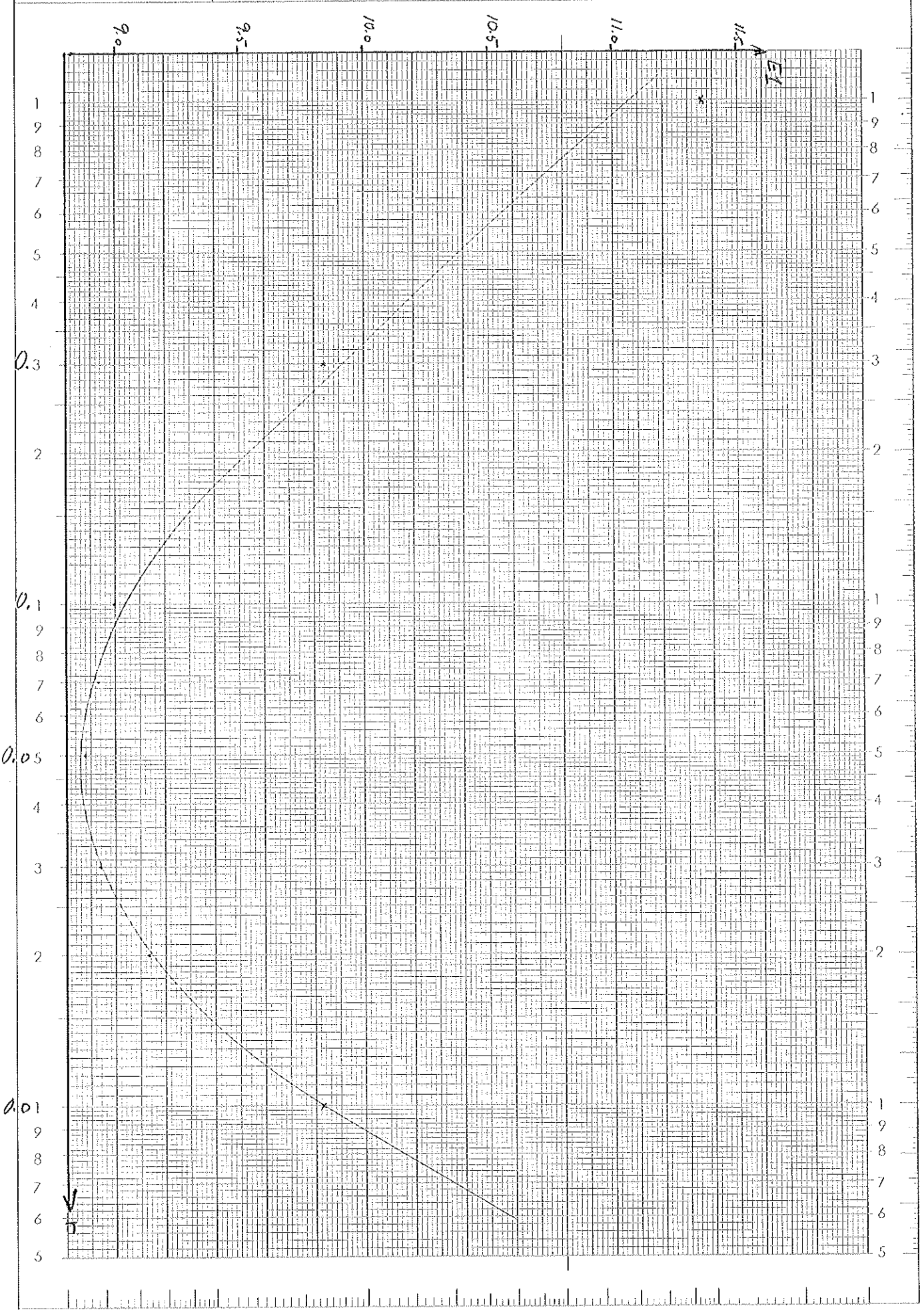
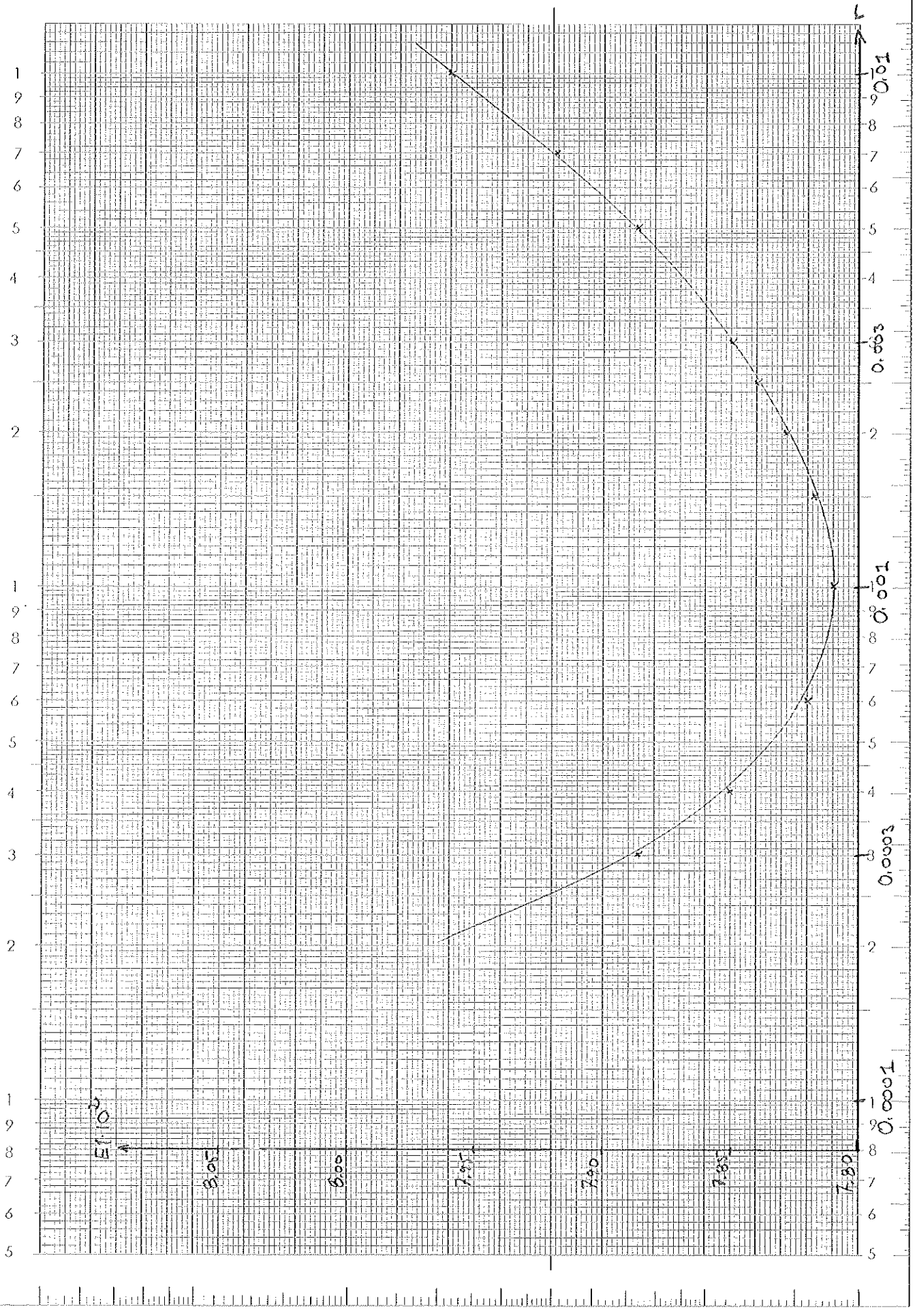


Diagram 4
KALID
k = 1.0

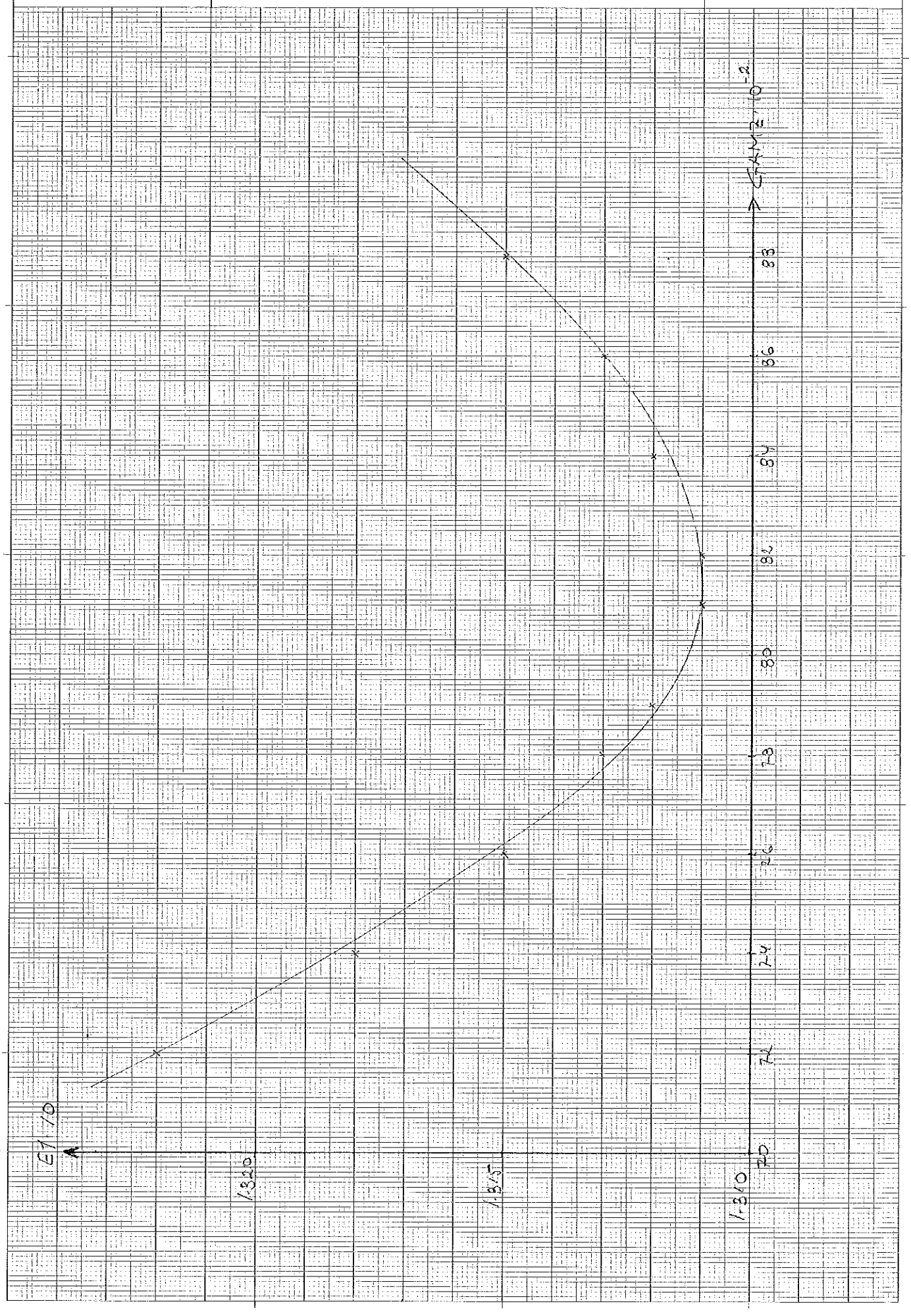


SSELIE
4460

Diagram 5

STAPP

$k = 0.1$



514 A4
SIS 73 25 01
TULLBERG KLIPPAN
Nr 1624

Diagram 6
STAPP
k = 10