

PROGRAM FÖR MAXIMUM-LIKELIHOOD-
IDENTIFIERING PÅ PDP-15.

RUTGER ALMQVIST

RAPPORT RE-103, februari 1972

PROGRAM FÖR MAXIMUM-LIKELIHOOD-
IDENTIFIERING PÅ PDP-15

Examensarbete vid institutionen för Reglerteknik

Rutger Almquist

Handledare: Ivar Gustavsson

ABSTRACT

In this report a program for identification by the Maximum-Likelihood method is presented. The program is written in FORTRAN for a process computer, PDP-15. The application is limited to linear, single output, time-invariant dynamic systems with normal disturbances having rational spectra.

First a resumé of the Maximum-Likelihood method is given. The different strategies which are used in the hill-climbing algorithm are discussed. Then there is an account of the structure of the program. In connection with this, there is a description of how to use the program, the possibilities and the limitations. Finally results from some test examples are given.

SAMMANFATTNING

I denna rapport presenteras ett program för Maximum-Likelihood identifiering. Programmet är skrivet i FORTRAN för PDP-15. Här förutsätts ett lineärt, tidsinvariant system med en utsignal och normalfördelade störningar.

En kort introduktion till teorin bakom Maximum-Likelihood metoden ges. Olika strategier i minimeringsalgoritmen diskuteras. Sedan följer en genomgång av programmets delar. Denna genomgång fungerar också som en handledning för användaren. Till sist ges resultat från några testexempel.

<u>INNEHÅLL</u>	Sida
INLEDNING	1
1. TEORETISK BAKGRUND	2
1.1 Modellen	2
1.2 Likelihoodfunktionen	3
1.3 Minimeringsalgoritmen	4
1.4 Beräkning av gradienten och andraderivatorna till förlustfunktionen	5
2. HILL-CLIMBING	8
2.1 Den normala strategien	8
2.2 Undantagen	10
2.3 Ytterligare tester och åtgärder	11
3. PROGRAMMET	13
3.1 Hierarkien	13
3.2 Programmets delar	13
3.3 COMMON-fälten	19
3.4 Länkstrukturen	19
4. TESTNING AV PROGRAMMET	21
4.1 Testexempel 1	21
4.2 Testexempel 2	24
4.3 Testexempel 3	24
4.4 Exekveringstider	26
APPENDIX A: Referenser	A 1
APPENDIX B: Flödesplaner och programhuvuden	B 1
APPENDIX C: The M-L Identifiers Guide	C 1
APPENDIX D: Data till testexempel	D 1
APPENDIX E: Några förslag till ändringar och utveckling av programmet	E 1

INLEDNING

Utformning av reglersystem för industriella processer, kräver ofta att man har tillgång till bra matematiska modeller för processen. Det går inte alltid att få fram dessa ur fysikaliska grundekvationer och man behöver därför metoder för att bestämma matematiska modeller genom direkt mätning på processen.

Maximum-Likelihood metoden är en processidentifieringsmetod, där man bestämmer parametrar i en matematisk modell med hjälp av mätdata från processen.

Detta examensarbete har gått ut på att skriva ett program för Maximum-Likelihoodidentifiering på PDP-15. Som grund ligger en tidigare version skriven för en CD-3600 av Ivar Gustavsson {3}. Denna version har arbetats om och utvecklats.

Här förutsätts lineära, tidsinvarianta system med en utsignal och normalfördelade störningar.

I kapitel 1 ges en kort introduktion till teorin bakom Maximum-Likelihood metoden och den minimeringsalgoritm som användes. I kapitel 2 diskuteras olika strategier som tillämpas i minimeringsalgoritmen. Kapitel 3 innehåller en redogörelse för programmets uppbyggnad. I kapitel 4, slutligen, finns några testexempel och exempel på exekveringstider.

1. TEORETISK BAKGRUND

I detta kapitel ger jag förutsättningarna och modellen, som ligger till grund för maximum-likelihoodidentifieringen. Dessutom ges huvuddragen i minimeringsalgoritmen. För noggrannare studium hänvisas till {3}.

1.1 Modellen

Vi har en linjär, tidsdiskret och tidsinvariant modell med en utsignal och m insignaler; där m kan vara 0.

$$A^*(q^{-1})y(t) = \sum_{i=1}^m B_i^*(q^{-1})u_i(t) + \lambda C^*(q^{-1})e(t) \quad (1)$$

där $e(t)$ är oberoende och $\in N(0,1)$

q är skiftoperatoren

$$q(t)x(t) = x(t+1) \quad (2)$$

$$\begin{cases} A^*(q) = 1 + a_1q + \dots + a_nq^n \\ B_i^*(q) = b_{i0} + b_{i1}q + \dots + b_{in}q^n \\ C^*(q) = 1 + c_1q + \dots + c_nq^n \end{cases} \quad i = 1, 2, \dots, m \quad (3)$$

Vi antar att funktionerna $A^*(q^{-1})$ och $C^*(q^{-1})$ har alla nollställen innanför enhetscirkeln och att det inte förekommer gemensamma faktorer till alla polynomen A , B och C .

Modellen enligt (1) kan även representera ett system med fördröjningar, τ , om dessa är jämna multiplar av samplingsintervallet T , dvs $k_i T$, $i = 1, 2, \dots, m$. Man klarar av detta helt enkelt genom att skifta in-utsignaldata k_i steg. På samma sätt kan man göra sig av med de konstanta termerna b_{i0} i B -polynomen. Sätt alltså b_{i0} , $i = 1, 2, \dots, m$, till 0 och skifta in-utsignaldata ett steg.

Det följer av (1) att residualerna $\{\varepsilon(t), t = 1, 2, \dots, N\}$ definierade av

$$C^*(q^{-1})\varepsilon(t) = A^*(q^{-1})y(t) - \sum_{i=1}^m B_i^*(q^{-1})u_i(t) \quad (4)$$

är oberoende och $\in N(0, \lambda)$.

1.2 Likelihoodfunktionen

Likelihoodfunktionen för $\epsilon(t)$, $t = 1, 2, \dots, N$ är produkten av frekvensfunktionerna för varje enskilt (t) . Frekvensfunktionen för en stokastisk variabel ξ , som $\epsilon \in N(m, \sigma)$ är:

$$f_{\xi}(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2\sigma^2}(x-m)^2\right\} \quad (5)$$

Likelihoodfunktionen för $\{\epsilon(t), t = 1, 2, \dots, N\}$ blir då:

$$L(\theta, \lambda) = \frac{1}{(\lambda\sqrt{2\pi})^N} \exp\left\{-\frac{1}{2\lambda^2} \sum_{t=1}^N \epsilon^2(t)\right\} \quad (6)$$

där θ är parametervektorn $(a_1, \dots, a_n, b_{11}, \dots, b_{1n}, b_{21}, \dots, b_{mn}, c_1, \dots, c_n)$

Vi skriver $L(\theta, \lambda)$ eftersom ϵ enligt (4) beror av de $n(2 + m)$ parametrarna och λ . Tills vidare har vi bortsett från initialvärden till differensekvation (4).

Då $L > 0$ gäller:

$$\ln L(\theta, \lambda) = -\frac{1}{2\lambda^2} \sum_{t=1}^N \epsilon^2(t) - N \ln \lambda - N \ln \sqrt{2\pi} \quad (7)$$

I fortsättningen kallas

$$V(\theta) = \frac{1}{2} \sum_{t=1}^N \epsilon^2(t) \quad (8)$$

för förlustfunktionen.

Eftersom maximeringen av $L(\theta, \lambda)$ med avseende på λ kan ske oberoende av maximeringen med avseende på θ , får vi maximum av L genom att först söka minimum av $V(\theta)$ med avseende på θ . Med hjälp av minimipunkten $\hat{\theta}$, erhålles sedan maximum av $\ln L$ med avseende på λ , för $\lambda = \hat{\lambda}$, där $\hat{\lambda} = \sqrt{2V(\hat{\theta})/N}$.

Maximum-likelihoodskattningen är konsistent, asymptotiskt normal och effektiv under vissa förutsättningar givna i {7}.

1.3 Minimeringsalgoritmen

För att söka minimum av $V(\theta)$ måste en iterativ teknik användas. Här utnyttjas en kombinerad Gauss-Newton och Newton-Raphson algoritm.

$$\theta^{k+1} = \theta^k - V_{\theta\theta}(\theta^k)^{-1} V_{\theta}(\theta^k) \quad (9)$$

V_{θ} är gradientvektorn av $V(\theta)$

$V_{\theta\theta}$ är matrisen med de partiella andraderivatorna av $V(\theta)$

Differentiering av (8) ger

$$\frac{\partial V}{\partial \theta_i} = \sum_{t=1}^N \epsilon(t) \frac{\partial \epsilon(t)}{\partial \theta_i} \quad (10)$$

$$\frac{\partial^2 V}{\partial \theta_i \partial \theta_j} = \sum_{t=1}^N \frac{\partial \epsilon(t)}{\partial \theta_i} \frac{\partial \epsilon(t)}{\partial \theta_j} + \sum_{t=1}^N \epsilon(t) \frac{\partial^2 \epsilon(t)}{\partial \theta_i \partial \theta_j} \quad (11)$$

Om endast den första termen i (11) utnyttjas, har vi en Gauss-Newton algoritm. Den användes under de första iterationerna. Nära minimum användes hela uttrycket (11). På så sätt fås snabb konvergens nära minimum. Den första termen kallar vi "approximativa andraderivatatan" och hela uttrycket "exakta andraderivatatan", eller $V_{\theta\theta}^x$ resp $V_{\theta\theta}^{ex}$.

Gången i algoritmen blir

1. $\theta^k = \theta^0$ (startvärde på θ)
2. Beräkna $V_{\theta}(\theta^k)$ och $V_{\theta\theta}(\theta^k)$
3. Beräkna θ^{k+1} och upprepa från 2.

Parameternoggrannheten fås med hjälp av inversen till andraderivatmatrisen

$$\sigma_{\theta_i}^2 = \lambda^2 \{V_{\theta\theta}(\hat{\theta})\}_{ii}^{-1} \quad (12)$$

där σ_{θ_i} är standardavvikelsen för parametern θ_i .

$\{V_{\theta\theta}(\theta)\}_{ii}^{-1}$ är elementet i i :te raden och i :te kolonnen i inversen av andraderivatmatrisen.

Standarddeviationen för förlustfunktionen är $\hat{\lambda}^2 \sqrt{N/2}$ och för λ approximativt $\hat{\lambda} / \sqrt{2N}$.

Observera att de statistiska resultaten bara gäller om processen verkligen kan beskrivas av (1). I annat fall kan den uppskattade modellen bara tolkas, som den modell, som minimerar enstegsprediktionsfelet.

1.4 Beräkning av gradienten och andraderivatorna till förlustfunktionen

Differentiering av (4) ger de formler, som behövs för beräkning av första och andraderivatorna av $\epsilon(t)$ med avseende på θ :

$$\begin{cases} C^*(q^{-1}) \frac{\partial \epsilon(t)}{\partial a_j} = q^{-j} y(t) \\ C^*(q^{-1}) \frac{\partial \epsilon(t)}{\partial b_{kj}} = -q^{-j} u_k(t) & k = 1, 2, \dots, m \\ C^*(q^{-1}) \frac{\partial \epsilon(t)}{\partial c_j} = -q^{-j} \epsilon(t) \end{cases} \quad (13)$$

Den sista ekvationen i (13) kan differentieras en gång till:

$$\begin{cases} C^*(q^{-1}) \frac{\partial \epsilon(t)}{\partial a_i \partial c_j} = -q^{-i-j+1} \frac{\partial \epsilon(t)}{\partial a_1} \\ C^*(q^{-1}) \frac{\partial \epsilon(t)}{\partial b_{ki} \partial c_j} = -q^{-i-j+1} \frac{\partial \epsilon(t)}{\partial b_{k1}} \\ C^*(q^{-1}) \frac{\partial \epsilon(t)}{\partial c_i \partial c_j} = -2q^{-i-j+1} \frac{\partial \epsilon(t)}{\partial c_1} \end{cases} \quad k = 1, 2, \dots, m \quad (14)$$

Här har använts

$$\frac{\partial \epsilon(t)}{\partial a_i} = q^{-i+1} \frac{\partial \epsilon(t)}{\partial a_1} = \frac{\partial \epsilon(t-i+1)}{\partial a_1} \quad i \leq t+1 \quad (15)$$

liksom liknande formler för derivatorna med avseende på b- och c-parametrarna. Ekvation (15) härledes från (13).

En utförlig redogörelse för hur dessa formler utnyttjas, programmeringsmässigt sett, finns i {3}. Här skall endast tas upp de åtgärder, som införandet av initialvärden till (4) medför. Om dessa införes på

normalt sätt, dvs genom att införa begynnelsevärden för $\epsilon(1), \epsilon(2), \dots, \epsilon(n)$, så får man svårigheter då derivatorna skall beräknas. Eftersom $\epsilon(1), \epsilon(2), \dots, \epsilon(n)$ är parametrar är alla derivator av dessa med avseende på a_1, b_1 och c_1 lika med 0. För derivator av $\epsilon(1), \epsilon(2), \dots, \epsilon(n)$ med avseende på $\epsilon(1), \epsilon(2), \dots, \epsilon(n)$ gäller:

$$\frac{\partial \epsilon(i)}{\partial \epsilon(j)} = \delta_{ij} \quad \begin{cases} \delta_{ij} = 0 & i \neq j \\ \delta_{ij} = 1 & i = j \end{cases}$$

$$i, j = 1, 2, \dots, n$$

Det är ej möjligt att utnyttja formler av typen (15) för beräkning av dessa derivator. Därigenom kan ej den snabbare beräkningsmetoden enligt {3} användas om initialvärdena införes på detta sätt. Istället använder man parametrarna

$$d_i = - \sum_{k=i}^n a_k y(i-k) \quad i = 1, 2, \dots, n \quad (16)$$

Om vi betraktar t ex ett tredje ordningens system med en insignal har vi

$$\begin{cases} y(1) = -a_1 y(0) - a_2 y(-1) - a_3 y(-2) + b_1 u(0) + b_2 u(-1) + b_3 u(-2) + e_1 + \\ \quad + c_1 e(0) + c_2 e(-1) + c_3 e(-2) \\ y(2) = -a_1 y(1) - a_2 y(0) - a_3 y(-1) \dots \\ y(3) = -a_1 y(2) - a_2 y(1) - a_3 y(0) \dots \end{cases} \quad (17)$$

Låt oss anta $u(t), \epsilon(t) = 0$ om $t \leq 0$. (17) och (4) ger

$$\begin{cases} \epsilon(1) = y(1) + a_1 y(0) + a_2 y(-1) + a_3 y(-2) \\ \epsilon(2) = -c_1 \epsilon(1) + y(2) + a_1 y(1) + a_2 y(0) + a_3 y(-1) + b_1 u(1) \\ \epsilon(3) = -c_1 \epsilon(2) + c_2 \epsilon(1) + y(3) + a_1 y(2) + a_2 y(1) + a_3 y(0) + b_1 u(2) + \\ \quad + b_2 u(1) \end{cases}$$

Sätt

$$\begin{cases} d_1 = -a_1 y(0) - a_2 y(-1) - a_3 y(-2) \\ d_2 = -a_2 y(0) - a_3 y(-1) \\ d_3 = -a_3 y(0) \end{cases}$$

Det gäller nu

$$\frac{\partial \varepsilon(t)}{\partial d_i} = \frac{\partial \varepsilon(t-i+1)}{\partial d_1}$$

Vi kan alltså behandla d_i , $i = 1, 2, \dots, n$ på samma sätt som a-, b- och c-parametrarna.

2. HILL-CLIMBING

I detta kapitel diskuteras en del svårigheter och problemlösningar i samband med sökandet av minimum för förlustfunktionen, "hill-climbing". Denna hill-climbing är baserad på den i kap 1 nämnda Newton-Raphson algoritmen (9)

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

dvs

$$\theta^{k+1} = \theta^k - \Delta\theta \tag{18}$$

Den valda strategien har drag gemensamma med såväl den som använts av Bohlin {1} som den av Gustavsson {3}.

Ett antal modifieringar måste användas i situationer, där algoritmen annars inte skulle fungera. Sådana situationer kan inträffa på grund av lokala ogynnsamma egenskaper hos förlustfunktionen. Tre sådana ogynnsamma egenskaper är:

- a. $V_{\theta\theta}$ singular
- b. $V_{\theta\theta}$ icke-positivt definit
- c. Funktionen $V(\theta)$ växande i stegets riktning

2.1 Den normala strategien

I den följande diskussionen bortses från d-parametrarna. $V(\theta)$ är kvadratisk i dessa och behandlingen av dessa skiljer sig ej från behandlingen av a- och b-parametrarna.

I huvudsak är minimeringen uppdelad i tre faser, i vilka man använder olika strategier. Här är strategi synonymt med val av viss typ av $V_{\theta\theta}$. Valen baseras på följande faktorer:

1. Om c-parametrarna fixeras och endast a- och b-parametrarna får variera, blir förlustfunktionen kvadratisk (i a- och b-parameter-rummet) och algoritmen konvergerar mot ett minimum i ett steg. Vi får alltså en minsta kvadratskattning av parametrarna. Detta kan

speciellt användas som inledande steg för att förbättra startvärdena $\theta^0 = 0$ till algoritmen.

2. Enligt kap 1 använder man $V_{\theta\theta}^x$ i ekvation (9) när man befinner sig långt ifrån minimat. Nära minimat använder man $V_{\theta\theta}^{ex}$ istället. För θ -värden långt ifrån minimat ger varken $V_{\theta\theta}^{ex}$ eller $V_{\theta\theta}^x$ nödvändigtvis en bra uppskattning av vare sig steglängd eller riktning. Genom att använda den positivt (semi)definita $V_{\theta\theta}^x$ garanteras emellertid att man inte går i riktning mot ökande förlustfunktion. Dessutom kräver beräkningen av $V_{\theta\theta}^x$ mindre exekveringstid. Man använder $V_{\theta\theta}^{ex}$ i slutet av iterationsprocessen, endast för att försäkra sig om kvadratisk konvergenshastighet.
3. Då man har en kvadratisk funktion $V(\theta)$, blir reduktionen av denna, om man går från en godtycklig punkt θ till minimat $\hat{\theta}$

$$\Delta V(\theta, \hat{\theta}) = \frac{1}{2}(\theta - \hat{\theta})^T V_{\theta}(\theta) \quad (19)$$

Om θ är den verkliga minimipunkten θ_0 så gäller för stora N

$$\Delta V(\theta_0, \hat{\theta}) = q(\theta_0, \hat{\theta}) V(\theta_0) / N \quad (20)$$

där $q(\theta_0, \hat{\theta})$ är χ^2 -fördelad med $n(2+m)$ frihetsgrader och alltså har medelvärdet $n(2+m) \{1\}$. Vi använder q som ett relevant mått på avståndet mellan förlustfunktionens värden i slutet av iterationsprocessen. (19) och (20) ger

$$q(\theta^k, \theta^{k+1}) = \frac{N}{2}(\theta^k - \theta^{k+1})^T V_{\theta}(\theta^k) / V(\theta^k) \quad (21)$$

Vi låter storleken på q avgöra när vi skall byta från $V_{\theta\theta}^x$ till $V_{\theta\theta}^{ex}$. Storleken på q utnyttjas också som ett mått på konvergensen. Eftersom $\hat{\theta}$ inte avviker signifikant från θ_0 , blir en minskning av förlustfunktionen, som är mycket mindre än $n(2+m) V(\theta) / N$ aldrig signifikant.

Dessa tankegångar utnyttjas på följande sätt. Vi delar in iterationsprocessen i tre faser:

$$\text{Fas 1} \quad V_{\theta\theta} = \begin{bmatrix} V_{aa} & V_{ab} & 0 \\ V_{ba} & V_{bb} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Fas 2} \quad V_{\theta\theta} = V_{\theta\theta}^x$$

$$\text{Fas 3} \quad V_{\theta\theta} = V_{\theta\theta} \text{ ex}$$

Övergång från fas 1 till fas 2 sker efter 1 steg. Övergång från fas 2 till fas 3 sker, då q blir mindre än en konstant, här satt lika med 10. Totala exekveringstiden beror på denna konstant. Väljer vi mindre konstant, får vi fler iterationer. Väljer vi större, går mera tid åt per iteration. Värdet 10 har tidigare [1] befunnits vara en lämplig kompromiss.

Under iterationsprocessens gång ändrar vi alltså typ av $V_{\theta\theta}$ i följande ekvation, som används för att ta fram $\Delta\theta$

$$V_{\theta\theta} \Delta\theta = V_{\theta} \quad (22)$$

Dessa huvudregler gäller så länge inte någon av de i kapitlets inledning nämnda svårigheterna uppträder.

2.2 Undantagen

a. $V_{\theta\theta}$ singular

Problem uppstår då $V_{\theta\theta}$ innehåller linjärt beroende eller nästan beroende kolonner. Vi kan då inte lösa ekvation (22). För att klara detta går vi, om fas = 3, över till fas 2. Dvs man använder $V_{\theta\theta}^x$ istället för $V_{\theta\theta} \text{ ex}$ i ekvation (22). Om vi befinner oss i fas 2, dvs om $V_{\theta\theta}^x$ visar sig vara singular, användes följande:

$$\Delta\theta = V_{\theta\theta}^{\dagger} V_{\theta} \quad (23)$$

där pseudoinversen $V_{\theta\theta}^{\dagger}$ har följande utseende:

$$V_{\theta\theta}^{\dagger} = R_{\Sigma}^{\dagger} R^T$$

R är en matris, vars kolonner är lika med egenvektorerna till $V_{\theta\theta}^x$.
 Σ^\dagger är en diagonalmatris, där diagonalen består av $\sigma_1^\dagger, \sigma_2^\dagger, \dots, \sigma_n^\dagger$.

$$\sigma_i^\dagger = \begin{cases} 1/\sigma_i & \sigma_i \geq 10^{-5} \\ 0 & \sigma_i < 10^{-5} \end{cases}$$

σ_i , ($i=1,2, \dots, n$) är egenvärdena till $V_{\theta\theta}^x$.

b. $V_{\theta\theta}$ icke-positivt definit

Detta problem kan i princip endast uppstå i fas 3. Avrundningsfel etc kan dock även ge upphov till detta i fas 2. Samtidigt som man testar $V_{\theta\theta}$ angående singularitet (alltså strax innan (22) skall lösas), testas om $V_{\theta\theta}$ är icke-positivt definit. Om så är fallet ersättes $V_{\theta\theta}^x$ med $V_{\theta\theta}^\dagger$ och om problemet uppstår i fas 2 användes $V_{\theta\theta}^\dagger$.

c. Förlustfunktionen växande i stegets riktning

Då $\Delta\theta$ har erhållits, testas om ändringen i förlustfunktionen kommer att gå åt rätt håll. Riktningen är "down-hill", om och endast om $V_{\theta\theta}^T(\theta^k)\Delta\theta$ är positiv. (Observera att $\Delta\theta = \theta^k - \theta^{k+1}$ enligt (18)). Om $V_{\theta\theta}^T(\theta^k)\Delta\theta \leq 0$ beräknar vi ett nytt $\Delta\theta$ med hjälp av ett annat $V_{\theta\theta}$ än det som användes föregående gång. Antingen $V_{\theta\theta}^x$ eller $V_{\theta\theta}^\dagger$ beroende på vilket $V_{\theta\theta}$ som tidigare användes. Om inte ens $V_{\theta\theta}^\dagger$ hjälper, sätts $\Delta\theta = V_{\theta\theta}(\theta^k)$, "method of steepest descent".

2.3 Ytterligare tester och åtgärder

Instabilt C-polynom

Om rötterna till $C^*(q^{-1})$ ligger utanför enhetscirkeln är ekvationen (4) instabil. Detta testas då $\theta^{k+1} = \theta^k - \Delta\theta$ har erhållits, Skulle $C^*(q^{-1})$ visa sig ha någon rot utanför enhetscirkeln halveras $\Delta\theta$ och θ^{k+1} beräknas igen. Därefter åter test av stabiliteten. Om maximala relativa ändringen $\max_i \left| \frac{\Delta\theta_i}{\theta_i} \right|$ blir mindre eller lika med ett tal eps (eps här satt 10^{-4}) utan att man uppnår stabilitet avbryts identifieringen. Om

$$|\theta_j| \leq 10^{-4} \text{ jämförs } \max_i \left| \frac{\Delta\theta_i}{\theta_i} \right| \text{ och } \max_j |\Delta\theta_j| \quad i \neq j.$$

Den nya förlustfunktionen $V(\theta^{k+1})$ trots allt större än $V(\theta^k)$

Då man med hjälp av θ^{k+1} har beräknat $V(\theta^{k+1})$, undersöks om förlustfunktionen verkligen har minskat. Annars halveras $\Delta\theta$ och nytt $V(\theta^{k+1})$ beräknas. Vi håller på så tills antingen $V(\theta^{k+1}) \leq V(\theta^k)$ eller tills maximala relativa ändringen blir mindre eller lika med eps. Inträffar det senare avbryts identifieringen. Detta avbrott kan i princip bara ske då vi ligger mycket nära minimat. Det medför att den vid avbrottet erhållna förlustfunktionen kan vara större än den, som erhöles i föregående iteration. I detta fall skrivs alltid föregående förlustfunktionsvärdet ut.

Konvergenstester

I slutet på varje iteration testas om konvergens har uppnåtts. Här används tre indikationer på konvergens:

$$q \leq \text{eps1}$$

$$\max_i \left| \frac{\Delta\theta_i}{\theta_i} \right| \leq \text{eps2} \quad (24)$$

$$\left| \frac{\Delta V}{V} \right| \leq \text{eps3}$$

Är något av dessa villkor uppfyllda anser vi oss ha konvergens och identifieringen avbryts.

eps1 har valts till 10^{-3} . Se (21).

eps3 har, på grund av maskinnoggrannheten, valts till 10^{-6} .

Då $V(\theta)$ beror på (approximativt) θ^2 borde eps2 = 10^{-3} överensstämma med det värde vi valde på eps3. För säkerhets skull sätter vi eps2 = 10^{-4} .

3. PROGRAMMET

I detta kapitel redogörs för hierarkien i programmet, de ingående subrutinerna och kommunikationen dem emellan. Dessutom behandlas uppbyggnaden av länkstrukturen med CHAIN. Flödesplaner och programhuvuden finns i appendix B.

3.1 Hierarkien

I fig B 1 åskådliggöres rangordningen mellan de ingående rutinerna. Rutan märkt X anger, att det finns möjlighet att lägga in ytterligare rutiner under MLADM.

De i kap 2 diskuterade strategierna och problemlösningarna tillämpas i de 4 subrutinerna MLINI, MLSOL, MLCOR och MLCON. Denna uppdelning i subrutiner är nödvändig pga det begränsade minnesutrymme, som står till förfogande.

3.2 Programmets delar

MLADM, Maximum Likelihood Administration (se fig B 2)

Detta är huvudprogrammet, som administrerar identifieringen. Programmet ser till att frågor skrivs ut på teletypen. Användarens svar på dessa frågor styr sedan identifieringen. Här följer frågorna med kommentarer:

DO YOU WANT THE ML-IDENTIFIERS GUIDE?

Svar kan vara YES, Y, NO eller N. En handledning för användaren (finns i appendix C) skrivs ut på radskrivaren om svar YES (Y). En subroutine MLGID sköter denna utskrift.

ARTIFICIAL DATA?

En särskild subroutine CREAMFI (se nedan) genererar artificiella in-utsignaldata.

Rutan "Read input data" i flödesschemat motsvarar de följande 7 indata:

NO=	Modellens ordningstal
NI=	Antal insignaler (min 0, max 8)
NP=	Antal mätpunkter
INIT=	0 Ingen skattning av initialvärden
	1 Skattning av initialvärden

- LS= 0 Ingen minsta kvadratskattning i första iterationen.
Användaren måste ge egna startvärden
- 1 Minsta kvadratskattning i första iterationen
- 2 Minsta kvadratskattning i första iterationen
Användaren måste ge egna startvärden på c-parametrarna
- IT= 0 Iterationsprocessen fortgår tills konvergens eller avbrott
- 1 En (1) iteration
- IPRINT= 0 Ingen utskrift mellan iterationerna. Startvärdena θ^0 , $V(\theta^0)$, $\lambda_{\pm\sigma}(\lambda)$ skrivs ut. Från sista iterationen fås utskrift av $\theta_{\pm\sigma}(\theta)$, $V(\theta)$, $\lambda_{\pm\sigma}(\lambda)$, $V_{\theta}(\theta)$, $V_{\theta\theta}(\theta)$, $V_{\theta\theta}^{-1}(\theta)$ och egenvärdena för $V_{\theta\theta}(\theta)$. Dessutom fås utskrift från första iterationen av $\theta_{\pm\sigma}(\theta)$, $V(\theta)$, $\lambda_{\pm\sigma}(\lambda)$. Från eventuell minsta kvadratskattning fås inte $\sigma(\theta)$ för c-parametrarna eftersom dessa då är fixerade.
- 1 Ger samma som IPRINT=0 men även utskrift mellan iterationerna av $\theta_{\pm\sigma}(\theta)$, $V(\theta)$, $\lambda_{\pm\sigma}(\lambda)$.
- 2 Samma som IPRINT=1. Dessutom fås $V_{\theta}(\theta)$, $V_{\theta\theta}(\theta)$, $V_{\theta\theta}^{-1}(\theta)$ och egenvärdena till $V_{\theta\theta}$ i varje iteration.

LIMITED RESIDUALS?

Om man inte vill att residualer $> 3\lambda$ skall påverka identifieringen kan man begränsa dessa till 3λ .

FIXED PARAMETERS?

Vill man ha vissa parametrar fixerade genom hela iterationsprocessen svarar man YES och får följande frågor: HOW MANY? och WHICH?. På WHICH? svarar man exempelvis:

A 1

B11

B23

C 3

Om NI=1 behöver man inte skriva B11 utan B 1 räcker.

När man har hunnit så långt övertar MLCAL identifieringen. Då iterationsprocessen har avbrutits följer:

CHANGE ORDER?

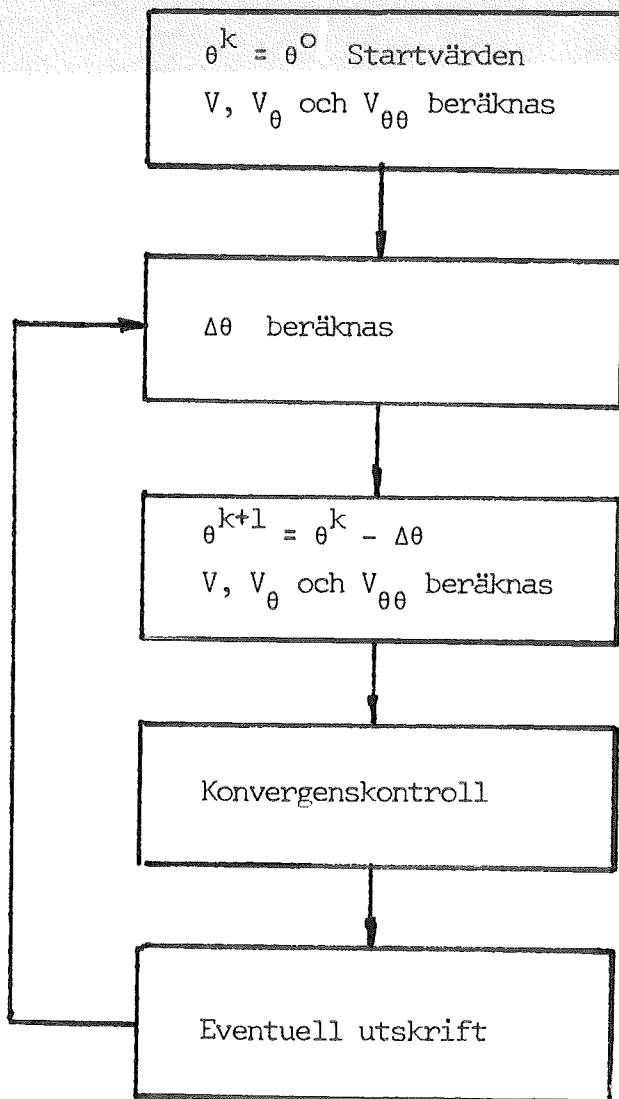
En ny identifiering med endast ordningstalet NO och eventuella fixerade parametrar ändrade, kan fås.

NEW IDENTIFICATION?

Om svar YES så hoppar man till början av programmet. AA i flödes-schemat.

SUBROUTINE MLCAL, M-L CALLs (se fig B 3)

Denna rutin ser till att gången i algoritmen blir den riktiga. I kap 1 gavs gången i några huvudpunkter. Följande flödesschema åskådliggör hur algoritmen genomlöpes i programmet:



De fem rutorna motsvaras uppifrån och ner subrutinerna MLINI, MLSOL, MLCOR, MLCON och MLPRI.

Detta flödesschema är i princip samma som i fig B 3.

En klocka, subrutinen TIME från programbiblioteket, startar i början av varje iteration. I slutet av varje iteration adderas och kontrolleras exekveringstiden. Om tiden överskrider fem minuter fås följande fråga:

CONTINUE?

Om svar NO så avbryts iterationsprocessen, om svar YES kommer nästa fråga:

NEW START-VALUES?

Om svar NO, fortsätter man som om ingenting hade hänt, om svar YES, får man tillfälle att ge nya startvärden och iterationsprocessen börjar på nytt.

Om avbrottsindikatorn IUT blivit > 0 i någon av underrutinerna till MLCAL, fås ett sista anrop av MLPRI och därefter återhopp till MLADM. En sak som inte framgår av flödesschemat i fig B 3, är att om IPRINT = 0 så anropas inte MLPRI mellan iterationerna.

SUBROUTINE MLINI, M-L INITiation (se fig B 4)

Beroende på vad man har satt LS till, läses eventuella startvärden till algoritmen ifrån denna rutin.

Genom anrop till FUNC fås $V(\theta^0)$, $V_\theta(\theta^0)$ och $V_{\theta\theta}(\theta^0)$. Om man har satt LS = 0 och IT = 1, vill man inte ha θ^{k+1} . Därför sätter rutinen IUT = 1 och uthopp till MLADM via MLCAL sker redan här.

Om LS = 1 eller 2, justerar $V_{\theta\theta}$ till

$$\begin{bmatrix} V_{aa} & V_{ab} & 0 & (V_{ad}) \\ V_{ba} & V_{bb} & 0 & (V_{bd}) \\ 0 & 0 & 1 & 0 \\ (V_{da}) & (V_{db}) & 0 & (V_{dd}) \end{bmatrix}$$

och V_θ till $(V_a, V_b, 0, (V_d))$. Alltså det som i kap 2 benämnes fas 1.

SUBROUTINE MLSOL, M-L SOLution (se fig B 5)

I första rutan i flödesschemat stöter man på fasindikatorn IVTT. IVTT = 1 anger, att exakta andraderivatematrisen användes. IVTT = 2 anger, att approximativa andraderivatematrisen användes. Här, i början av MLSOL, sker också en justering av $V_{\theta}(\theta)$ och $V_{\theta\theta}(\theta)$, om det finns fixerade parametrar.

Med hjälp av DESYM från programbiblioteket fås bl a rangen på $V_{\theta\theta}$. DESYM anger även, om $V_{\theta\theta}$ är icke-positivt definit. Därefter tillämpas de strategier beträffande singularärt och icke-positivt definit $V_{\theta\theta}$, som diskuterades i kap 2. Subrutinen MLPSE användes, om pseudo-inversen måste beräknas. I MLSOL ligger även testet på $\Delta\theta V_{\theta}^T$.

Om NSP = 1 fås ett meddelande via MLPRI:

PSEUDO-INVERSE WAS USED IN THIS ITERATION

Om NSI = 1 fås pss

DT = VT WAS USED IN THIS ITERATION

DT är programbeteckningen för $\Delta\theta$.

VT är programbeteckningen för $V_{\theta}(\theta)$.

SUBROUTINE MLCOR, M-L CORrection (se fig B 6)

Här korrigeras θ enligt (18). Vi testar sedan med hjälp av INPOL från programbiblioteket, om C-polynomet är stabilt. Därefter avgörs vilken fas algoritmen skall fortsätta i. IVTT får värdet 1 eller 2.

Om IVTT = 1 fås senare ett meddelande via MLPRI:

EXACT SEC. DER. WILL BE USED IN NEXT ITERATION

FUNC anropas för att få nya $V(\theta)$, $V_{\theta}(\theta)$ och $V_{\theta\theta}(\theta)$.

Sist ligger testet om förlustfunktionen verkligen har minskat.

Om NSV = 1 eller NSH = 1, dvs om $\Delta\theta$ har halverats minst en gång, fås senare ett meddelande via MLPRI:

COEFF. CORRECTION WAS HALVED IN THIS ITERATION

Om användaren har satt IT = 1 önskar han inte fler iterationer. Därför sätts IUT = 1 och uthopp till MLADM via MLCAL sker här redan efter första iterationen.

Om vid halvering av ändringen $\Delta\theta$ $\max_i \left| \frac{\Delta\theta_i}{\theta_i} \right|$ blir $\leq 10^{-4}$ sätts IUT = 2

för att iterationsprocessen skall brytas. Samtidigt skrivs något av dessa meddelanden ut:

NO SMALLER LOSS FUNCTION FOUND EVEN IF MAX RELATIVE COEFF: CORRECTION = 1.E -4

C - POLYNOMIAL NOT STABLE EVEN IF MAX RELATIVE COEFF. CORRECTION = 1.E -4

SUBROUTINE MLCON, M-L CONvergence (se fig B 7)

Här finns konvergenstesterna. Om konvergens, så sätts IUT = 1 för att iterationsprocessen skall avbrytas. Meddelandet CONVERGENCE fås på radskrivaren.

SUBROUTINE MLPRI, M-L PRInting (se fig B 8)

Denna subrutin sköter utskriften av den information man vill ha från identifieringen.

I MLPRI beräknas inversen av $V_{\theta\theta}$ på samma sätt som i MLPSE. I MLPSE beräknas pseudoinversen $V_{\theta\theta}^+$. Det gäller ju att $V_{\theta\theta}^+ = V_{\theta\theta}^{-1}$ om $V_{\theta\theta}^{-1}$ existerar. Vi undersöker egenvärden till $V_{\theta\theta}$. Om något av dem $\leq 10^{-6}$ är $V_{\theta\theta}$ singulär och vi får meddelandet:

INVERSE OF SEC. DER. MATRIX CANT BE COMPUTED

$V_{\theta\theta}^{-1}$ skrivs då alltså ej ut, ej heller $\sigma(\theta)$ enligt (12).

SUBROUTINE FUNC (se fig B 9)

Detta är i stort sett samma rutin (VV1V2), som användes av Gustavsson i {3}, för att evaluera $V(\theta)$, $V_{\theta}(\theta)$ och $V_{\theta\theta}(\theta)$. En del smärre ändringar har gjorts för att anpassa den för PDP-15. Dessutom har den i kap 1 nämnda möjligheten att ta hänsyn till initialvärden införts. Nu finns också möjlighet att begränsa residualerna. FUNC kräver att in- utsignaldata är lagrade i en fil IODATA EXT på skivan i följande ordning:

$u_1(1), u_2(1), \dots, u_m(1), y(1), \dots, u_1(N), u_2(N), \dots, u_m(N), y(N)$ $0 \leq m \leq 8$

Residualerna $\{\epsilon(t), t=1, 2, \dots, N\}$, som evalueras i FUNC, lagras i en fil RESIDU EXT på skivan.

SUBROUTINE CREAFI

Denna rutin används om man vill generera artificiella in- utsignaldata. Det finns möjlighet att generera data med en eller två insignaler, Den

första signalen är en PRBS (Pseudo-Random Binary Signal) med period 63 och amplitud 1.0. Insignal 2 är en PRBS med period 127 och amplitud 1.0. Utsignalen fås med hjälp av en modell som har samma utseende som (1). Störningarna $e(t)$ är oberoende, normalfördelade (0,1) och fås från MCNODI från programbiblioteket. Data lagras i filen IODATA EXT i den ordning som FUNC kräver. Man får även utskrift av data på radskrivaren. Input till CREAMI är NO, NI, NP, INIT, koefficienterna till polynomen i (1) och λ . NO och NP har samma betydelse som i MLADM. NI anger antal insignaler, som här måste vara 1 eller 2. Om INIT=0 fås initialvärdena 0 på $y(t)$, $t=1,2, \dots, NO$. Om INIT=1 får man ge egna initialvärden.

3.3 COMMON-fälten

Kommunikationen mellan rutinerna sker via COMMON/SLASK/, COMMON/A/, COMMON/B/ och COMMON/BB/. Alla delar i /SLASK/ användes. Subrutinen FUNC är konstruerad så, att den skall kunna användas i andra program. Därför kommunicerar den inte via /SLASK/ utan informationen överföres genom anropsargumenten. Utskriftsindikatorn IPRINT överföres i anropen av MLCAL och MLPRI. Redogörelse för innehållet i COMMON-fälten finns i programhuvudena i appendix B. Där framgår att parametervektorn har dimensionen 20. Detta betyder att antalet insignaler och ordningstalet på modellen begränsas enligt $NO(2+NI+INIT) \leq 20$.

3.4 Länkstrukturen

Programmet är så stort, att det inte samtidigt ryms i kärnminnet utan tekniken med CHAIN AND EXECUTE måste användas. Uppbyggnaden av länkstrukturen sker enligt följande:

```
$A DKAØ -5
```

```
$CHAIN
```

```
CHAIN V5A
```

```
NAME XCT FILE
```

```
>MLID
```

```
LIST OPTIONS & PARAMETERS
```

```
>NM
```

```
DEFINE RESIDENT CODE
```

```
>MLADM,MLCAL,REANS
```

```
DESCRIBE LINKS & STRUCTURE
```

```
>LK1=MLINI/FUNC
```

```
>LK2=MLSOL,MLPSE
```

```
>LK3=MLCOR,MLCON/FUNC
```

```
>LK4=MLPRI
```

```
>LK5=CREAFI
```

```
>LK6=MLGID
```

```
>LK1:LK2:LK3:LK4:LK5:LK6
```

```
>
```


Erforderligt minnesutrymme är 11.3 k och därav upptar resident kod
6.3 k.

LINK TABLE

37525-37636 00132

RESIDENT CODE

MLADM	36267-37504	01215
MLCAL	35414-36266	00653
REANS	35250-35413	00144
ATLTP6	35145-35247	00103
LPA	34301-35144	00644
BCDIO	31243-34300	03036
FIOPS	30503-31242	00542
TIME	30430-30502	00053
RELEASE	26736-27777	01042
INTEAE	30314-30427	00114
STOP	30301-30313	00013
SPMSG	30206-30300	00073
OTSER	30112-30205	00074
MOVE	30006-30111	00031
.DA	26667-26735	00047
.CB	30041-30062	00020
CNTRLP	30033-30040	00006
SLASK	24667-26666	02000
B	30022-30032	00011
EB	30016-30021	00004

LINK -- LK1

MLINI	23137-24666	01530
FUNC	14027-17777	03751
STDAI	21765-23136	01152
RDDAT	20621-21764	01144
RTIFF	20226-20620	00373
BINIO	13553-14226	00254
FILE	13220-13552	00333
SCAPRO	12607-13217	00411
BIO	20121-20225	00105
ABS	20103-20120	00016
FLOAT	20072-20102	00011
SIGN	20042-20071	00030
.BC	12533-12606	00054
SQRT	12441-12532	00072
.EE	12350-12440	00071
.EF	12232-12347	00116
.EC	12166-12231	00044
DOTSS	12054-12165	00112
A	11734-12053	00120

LINK -- LK2

MLSOL	22231-24666	02436
MLPSE	20120-22230	02111
EIGS	16156-17777	01622
DESYM	15175-16155	00761
SOLVS	14524-15174	00451
SCAPRO	14113-14523	00411
ABS	20102-20117	00016
FLOAT	20071-20101	00011
SQRT	14021-14112	00072
DOTSS	13707-14020	00112
A	13567-13706	00120

LINK -- LK3

MLCOR	23542-24666	01125
MLCON	23212-23541	00530
FUNC	14027-17777	03751
STDAI	21640-23211	01152
RDDAT	20474-21637	01144
INPOL	20060-20473	00414
BINIO	13553-14026	00254
FILE	13220-13552	00333
SCAPRO	12607-13217	00411
BIO	12502-12606	00105
ABS	20042-20057	00016
FLOAT	20031-20041	00011
SIGN	12452-12501	00030
SQRT	12360-12451	00072
DOTSS	12246-12357	00112
A	12126-12245	00120

LINK -- LK4

MLPRI	12142-17777	05636
EIGS	23045-24666	01622
SCAPRO	22434-23044	00411
ABS	22415-22433	00016
FLOAT	22425-22415	00011
SQRT	22313-22404	00072
DOTSS	22201-22312	00112

LINK -- LK5

CREAFI	22042-24666	02625
STDAI	20670-22041	01152
RDDAT	16634-17777	01144
RTIFF	20275-20667	00373
PRBSTA	20121-20274	00154
MCHODI	16527-16633	00125
PRB	20031-20120	00070
BINIO	15253-16526	00254
FILE	15720-16252	00333
BIO	15613-15717	00105
.BC	15537-15612	00054
.EE	15446-15536	00071
.EF	15330-15445	00116
.EC	15264-15327	00044
DOTSS	15152-15263	00112

LINK -- LK6

MLGID	24362-24666	00305
FILE	24027-24361	00333
DOTSS	23715-24026	00112

CORE REQ'D

11734-37636 25703

4. TESTING AV PROGRAMMET

Nedan följer tre testexempel. De två första är identifieringar på artificiellt genererade data. Det sista använder data från Ågestareaktorn. I slutet på kapitlet redovisas exekveringstider. In- utsignalsekvenserna till testexemplen finns i appendix D.

4.1 Testexempel 1

Detta är en identifiering på de 100 första in- utsignalsparen av de data som användes i "Test example 1" i {3}. Dessa data har medelvärdeskorri- gerats.

Systemet man genererade data ifrån var av andra ordningen med en insig- nal:

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

{u(t)} var en PRBS med period 263 och amplitud 1.0.

{e(t)} en sekvens av slumpstal, oberoende och $\epsilon N(0,1)$, från SUBROUTINE RANSS.

Programmet testades med ordningstalen 1, 2 och 3 på modellen. Slutresul- tat från identifieringarna visas i tabellerna 1, 2 och 3 nedan.

Antalet iterationer var i de olika fallen 4, 7 och 7.

Vid jämförelse med identifieringar utförda på dessa data med programmet i {3}, finner man att för $n = 2$ och $n = 3$ överensstämmer slutresultaten på alla punkter med en noggrannhet på 3 decimaler. Antalet iterationer är här en mindre för varje fall. Detta beror på att q (21) används. q avgör ju både när övergång från $V_{\theta\theta}^x$ till $V_{\theta\theta}$ sker och när konvergens uppnås.

Om man jämför slutresultaten för $n = 1$ finner man skillnader. Se tabell 4 nedan. Man kan förklara dessa med att q har brutit iterationsprocessen för tidigt. Här kan påpekas att då $N < 500$ borde man använda $q \leq 10^{-4}$, som konvergenstest istället för $q \leq 10^{-3}$.

CONVERGENCE ($0 < 1.E-03$)

FINAL ESTIMATION

A 1 -8.186404E-01 +- 6.135835E-02

B 1 1.821813E-01 +- 2.239707E-01

C 1 5.584842E-01 +- 7.194467E-02

LOSS FUNCTION: 3.012704E+02

LAMBDA: 2.454671E+00 +- 1.735714E-01

GRADIENT OF V

-1.104105E+01 1.026858E+01 6.868911E+00

SECOND DERIVATIVE MATRIX

1.737263E+03 2.544142E+01 -3.754928E+02

2.544142E+01 1.478661E+02 1.923206E+02

-3.754928E+02 1.923206E+02 1.510573E+03

INVERSE OF SECOND DERIVATIVE MATRIX

6.248286E-04 -3.709442E-04 2.025448E-04

-3.709442E-04 8.325224E-03 -1.152144E-03

2.025448E-04 -1.152144E-03 8.590350E-04

EIGENVALUES OF VTT

2.020981E+03 1.257336E+03 1.173854E+02

COEFF. CORRECTION WAS HALVED IN THIS ITERATION

THIS WAS ITERATION NR: 4 EXECUTION TIME: 0 MINUTES 21 SECONDS

Tabell 1 - All utskrift från sista iterationen. n = 1.

A 1	-1.488749E+00	+-	1.899989E-02
A 2	6.974376E-01	+-	1.676320E-02
B 1	8.954328E-01	+-	9.854494E-02
B 2	1.244611E+00	+-	1.229465E-01
C 1	-1.005956E+00	+-	1.030951E-01
C 2	2.048739E-01	+-	9.494968E-02

LOSS FUNCTION: 4.397358E+01
 LAMBDA: 9.378014E-01 +- 6.631257E-02

Tabell 2 - Parametrarna med standardavvikelser, förlustfunktionen och $\lambda \pm \sigma(\lambda)$. Sista iterationen. n = 2.

A 1	-5.784770E-01	+-	1.221819E-01
A 2	-6.539686E-01	+-	1.804749E-01
A 3	6.316951E-01	+-	8.626064E-02
B 1	8.956509E-01	+-	9.840033E-02
B 2	2.050452E+00	+-	1.447412E-01
B 3	1.151809E+00	+-	1.903841E-01
C 1	-9.247017E-02	+-	1.607575E-01
C 2	-6.717913E-01	+-	1.416230E-01
C 3	1.426973E-01	+-	1.046968E-01

LOSS FUNCTION: 4.365821E+01
 LAMBDA: 9.344326E-01 +- 6.607436E-02

Tabell 3 - Parametrarna med standardavvikelser, förlustfunktionen och $\lambda \pm \sigma(\lambda)$. Sista iterationen. n = 3.

A 1	-8.094498-01	+-	6.163385-02
B 1	1.024885-01	+-	2.176687-01
C 1	5.659066-01	+-	7.020438-02

Loss function: 3.008332+02
 Lambda: 2.452889+00

Tabell 4 - Slutresultat från en identifiering med programmet i {3}.n = 1.

4.2 Testexempel 2

Här har data genererats med CREAMI från ett andra ordningens system med två insignaler

$$y(t) - 1.5y(t-1) + 0.7y(t-2) = u_1(t-1) + 0.5u_1(t-2) + 0.7u_2(t-1) - \\ - 0.3u_2(t-2) + 1.5(e(t) - 1.0e(t-1) + 0.2e(t-2))$$

Längden på den undersökta sekvensen (N) var 500.

Beträffande $u_1(t)$, $u_2(t)$ och $e(t)$ se SUBROUTINE CREAMI i kap 3.

Slutresultat från en identifiering med $n = 2$ finns i tabell 5 nedan.

A 1	-1.490884E+00	+-	1.899624E-02
A 2	6.963925E-01	+-	1.563509E-02
B11	1.040509E+00	+-	6.358114E-02
B12	5.202218E-01	+-	8.688238E-02
B21	7.580590E-01	+-	6.310786E-02
B22	-2.573616E-01	+-	6.450422E-02
C 1	-1.043990E+00	+-	4.802297E-02
C 2	2.282428E-01	+-	4.556779E-02
LOSS FUNCTION:	5.156240E+02		
LAMBDA:	1.436139E+00	+-	4.541471E-02

Tabell 5 - Parametrarna med standardavvikelser, förlustfunktionen och $\lambda \pm \sigma(\lambda)$. Sista iterationen. $n = 2$ och $m = 2$.

4.3 Testexempel 3

Här har in- utsignaldata från Ågestareaktorn använts. Insignal var kontrollstavarnas läge och utsignal nukleär effekt. Samplingsintervall 1 sekund. Se mera härom i {5}. Längden på den undersökta sekvensen (N) var 513. (AR60A input 2 i {5}). Data har medelvärdeskorrigerats och skiftats ett steg.

I tabell 6 finns slutresultat från en identifiering med $n = 3$.

Startvärde $\theta^0 = 0$.

A 1 -1.714545E-01
 A 2 -8.433026E-01
 A 3 1.446676E-01

B 1 7.623992E-01
 B 2 3.270277E-02
 B 3 -5.956883E-01

C 1 4.411664E-01
 C 2 -2.586563E-01
 C 3 1.800701E-01

LOSS FUNCTION: 1.351275E+04

LAMBDA: 7.258189E+00 +- 2.265972E-01

Tabell 6 - Parametrarna utan standardavvikelser ($V_{\theta\theta}$ singular), förlustfunktionen och $\lambda \pm \sigma(\lambda)$. $n = 3$. Sista iterationen.

Om man använder följande startvärden får man en minskning av förlustfunktionen. Se tabellerna 7 och 8.

STARTING VALUES

A 1 -2.062000E+00
 A 2 1.253000E+00
 A 3 -1.880000E-01

B 1 7.600000E-01
 B 2 -1.352000E+00
 B 3 5.860000E-01

C 1 -2.009000E+00
 C 2 1.216000E+00
 C 3 -1.960000E-01

Tabell 7 - Startvärden till identifiering på Ågestadata.

```

A 1   -2.019734E+00 +- 1.518408E-02
A 2    1.152182E+00 +- 2.056254E-02
A 3   -1.292948E-01 +- 5.930528E-03

B 1    7.593616E-01 +- 2.874166E-03
B 2   -1.369049E+00 +- 1.104402E-02
B 3    6.112370E-01 +- 1.035407E-02

C 1   -1.591128E+00 +- 4.686959E-02
C 2    7.020185E-01 +- 8.033981E-02
C 3   -6.992095E-02 +- 4.544728E-02

LOSS FUNCTION: 1.102343E+04
LAMBDA:        6.555635E+00 +- 2.046638E-01

```

Tabell 8 - Parametrarna med standardavvikelser, förlustfunktionen och $\lambda \pm \sigma(\lambda)$. Slutresultat från identifiering med startvärden redovisade i tabell 7.

Detta exempel visar att om förlustfunktionen har ett besvärligt utseende så finns det risk att man hamnar i ett lokalt minimum då man startar från $\theta^0 = 0$.

4.4 Exekveringstider

Den tid det tar att göra en identifiering beror på antalet iterationer, ordningstalet, samplingssekvensens längd och hur ofta de olika undantagen (kap 2) inträffar. Endast en subrutine, FUNC, beror på N. I tabell 9 finns exempel på FUNC:s exekveringstider för olika n och N.

n \ N	100	500	1000
1	1.5 (2.1)	6.7 (9.7)	13.2 (19.0)
2	2.6 (3.5)	11.9 (16.6)	23.8 (33.1)
3	3.6 (5.0)	17.0 (23.6)	33.9 (47.0)

Tabell 9 - FUNC:s exekveringstid, för olika n och N, i sekunder.

Tider inom parentes gäller då $V_{\theta\theta}$ ex beräknas.

Tidsåtgången i utskriftsrutinen MLPRI finns redovisad i tabell 10. I tabell 11, slutligen, visas exekveringstiden för en iteration för olika n och N .

n	Tid (sek)
1	0.8 (2.7)
2	3.0 (5.4)
3	8.7 (14.5)

Tabell 10 - MLPRI:s exekveringstider för olika n . IPRINT = 1. Tider inom parentes gäller för IPRINT = 2.

n \ N	100	500	1000
1	7.4	11.9	18.3
2	10.1	19.4	31.1
3	18.4	30.1	46.8

Tabell 11 - Tidsåtgång för en iteration då approximativ andra-derivata användes. Tid i sekunder. IPRINT = 1. Inget av undantagen (kap 2) har inträffat.

APPENDIX A

REFERENSER

1. Bohlin T.: Analysis of stationary EEG-signals by the Maximum-Likelihood and Generalized Least-Squares Methods.
TP 18.200 IBM Nordic Laboratory, mars 1971.
2. Bohlin T.: On the Maximum-Likelihood method of identification.
IBM J. Res. Develop., januari 1970.
3. Gustavsson I.: Parametric identification on multiple input, single output linear dynamic systems.
Rapport 6907, Institutionen för reglerteknik, Lunds Tekniska Högskola, juli 1969.
4. Gustavsson I.: Processidentifiering.
STF-TLI kursverksamhet, januari 1970.
5. Gustavsson I.: Maximum-Likelihood identification of dynamics of the Ågesta reactor and comparison with results of spectral analysis.
Rapport 6903, Institutionen för reglerteknik, Lunds Tekniska Högskola, februari 1969.
6. Gustavsson S. och Ewing G.: Davidons minimeringsalgoritm vid Maximum-Likelihoodidentifiering.
Examensarbete, E71-05 Institutionen för regleringsteknik, Chalmers Tekniska Högskola, Göteborg mars 1971.
7. Åström K.J., Bohlin T. och Wensmark S.: Automatic construction of linear stochastic dynamic models for stationary processes with random disturbances using operating records.
TP 18.150 IBM Nordic Laboratory, 1965.

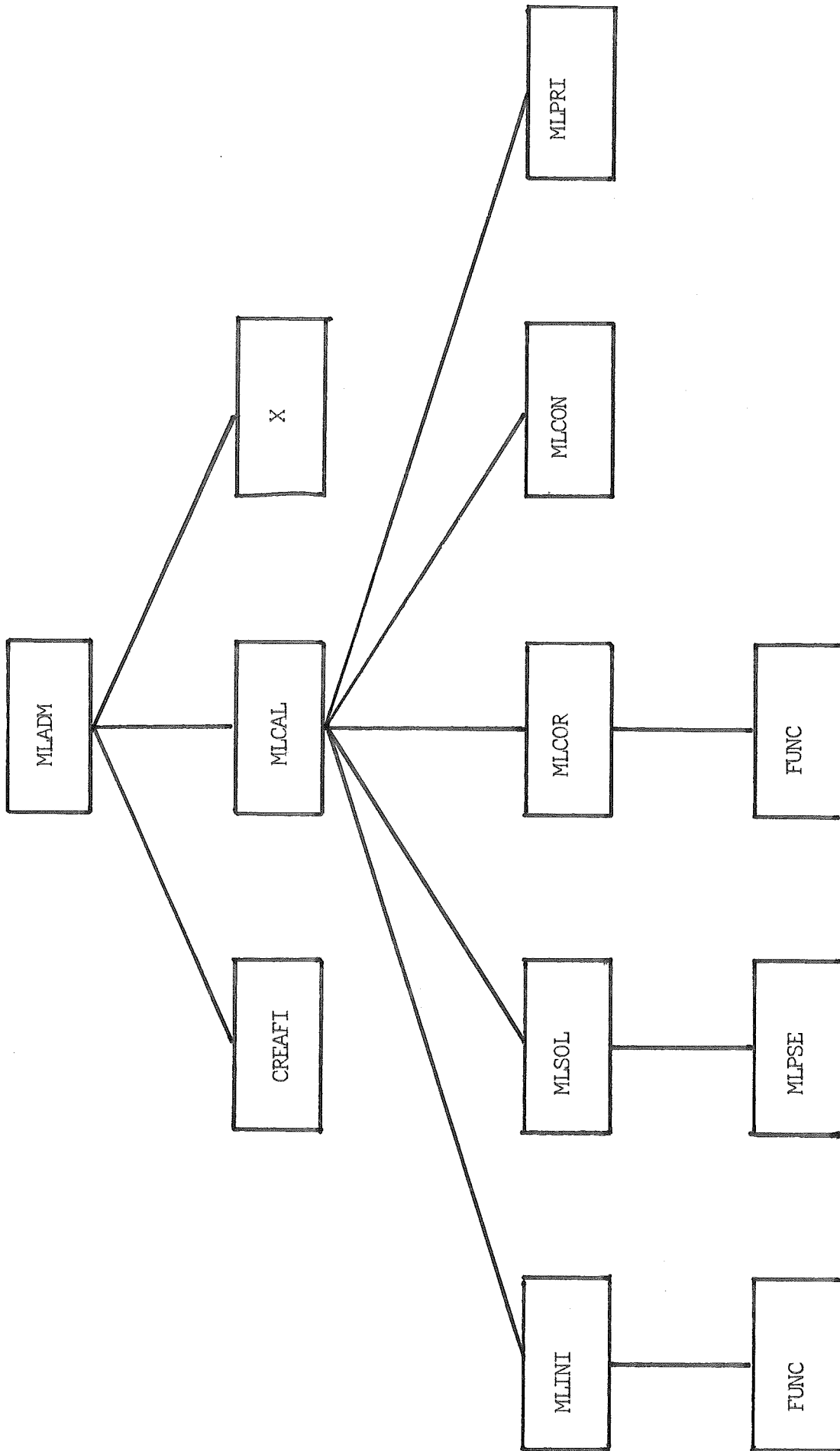


Fig B 1 - Hierarkien i programmet.

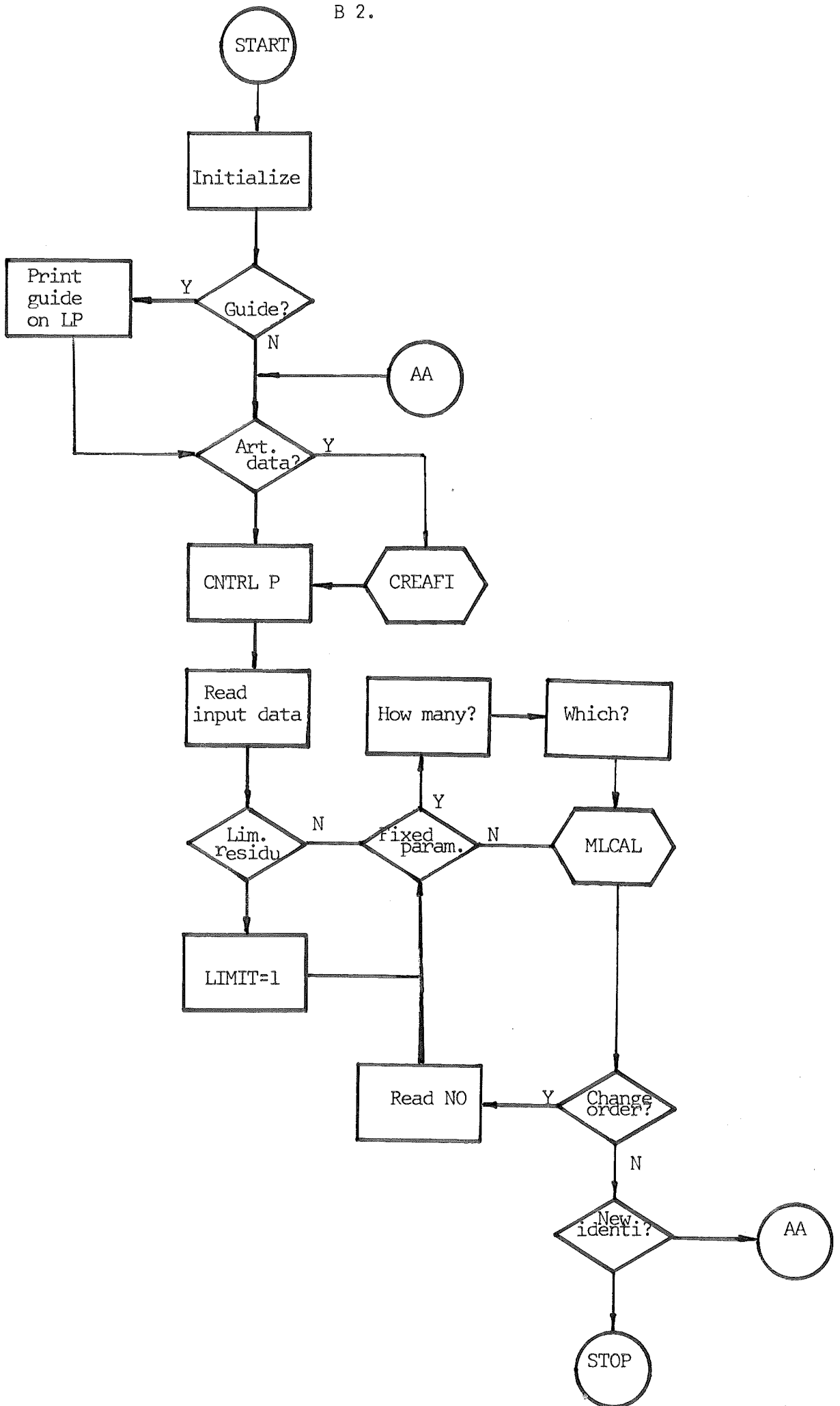


Fig B 2 - MLADM

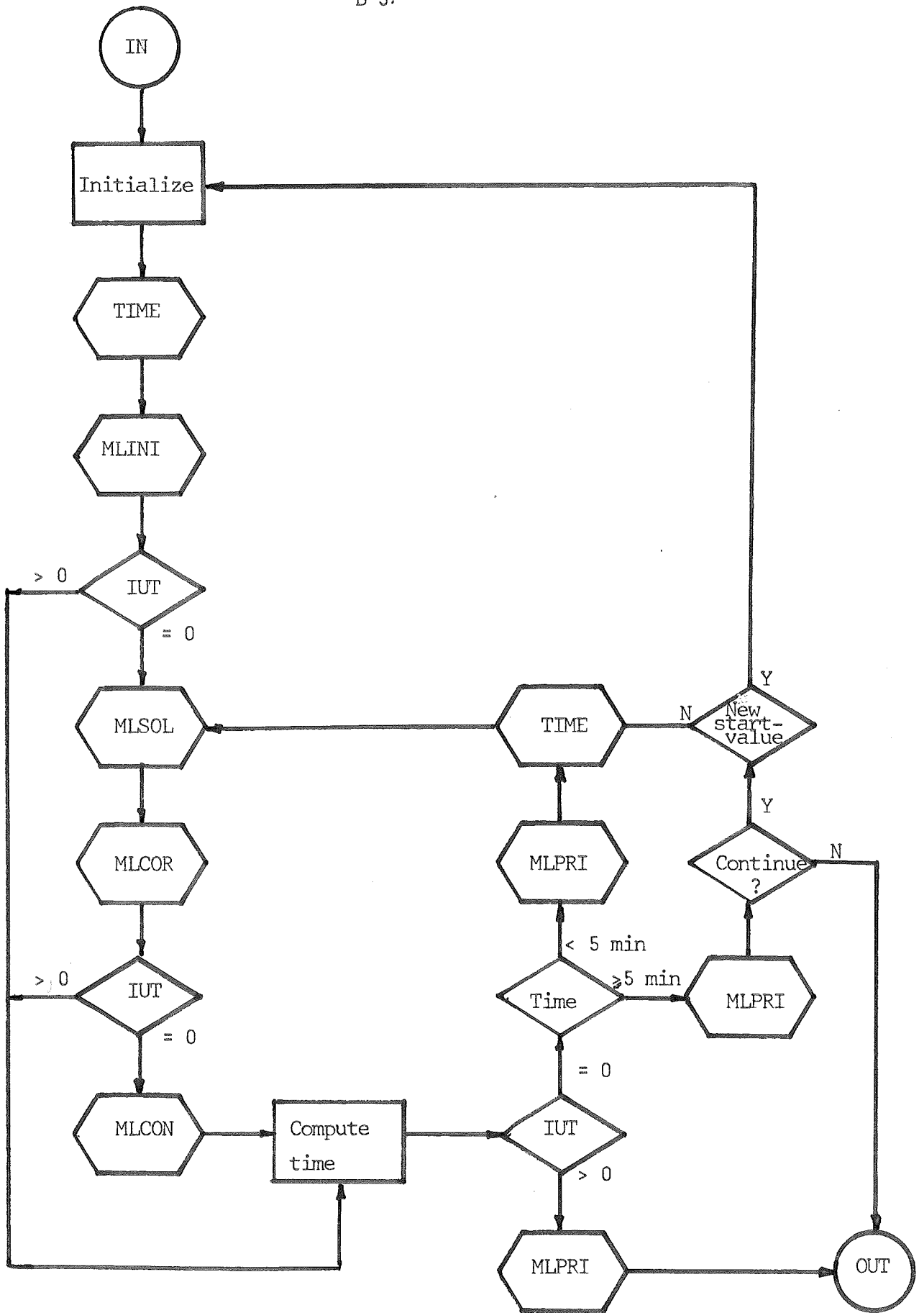


Fig B 3 - SUBROUTINE MLCAL

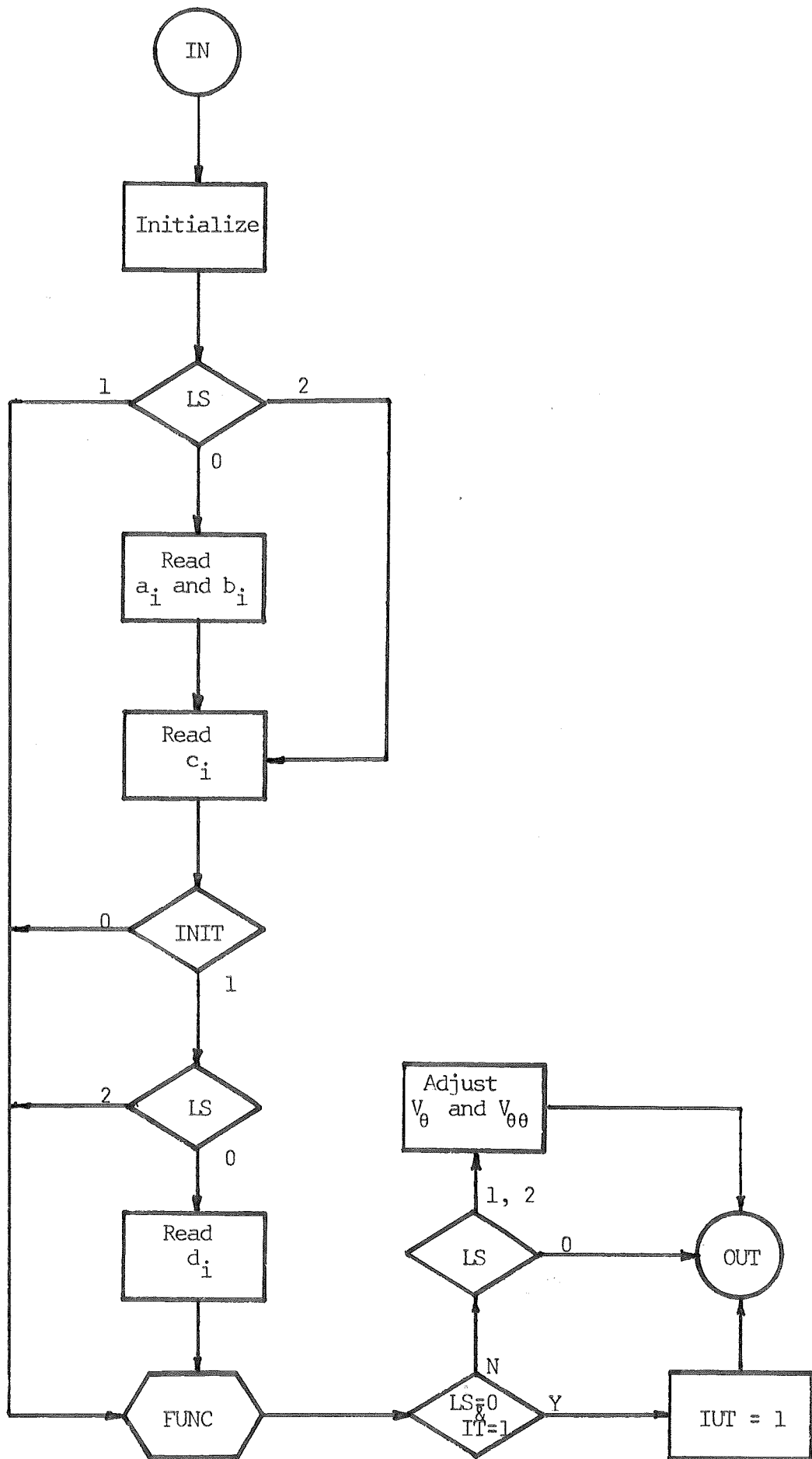


Fig B 4 - SUBROUTINE MLINI

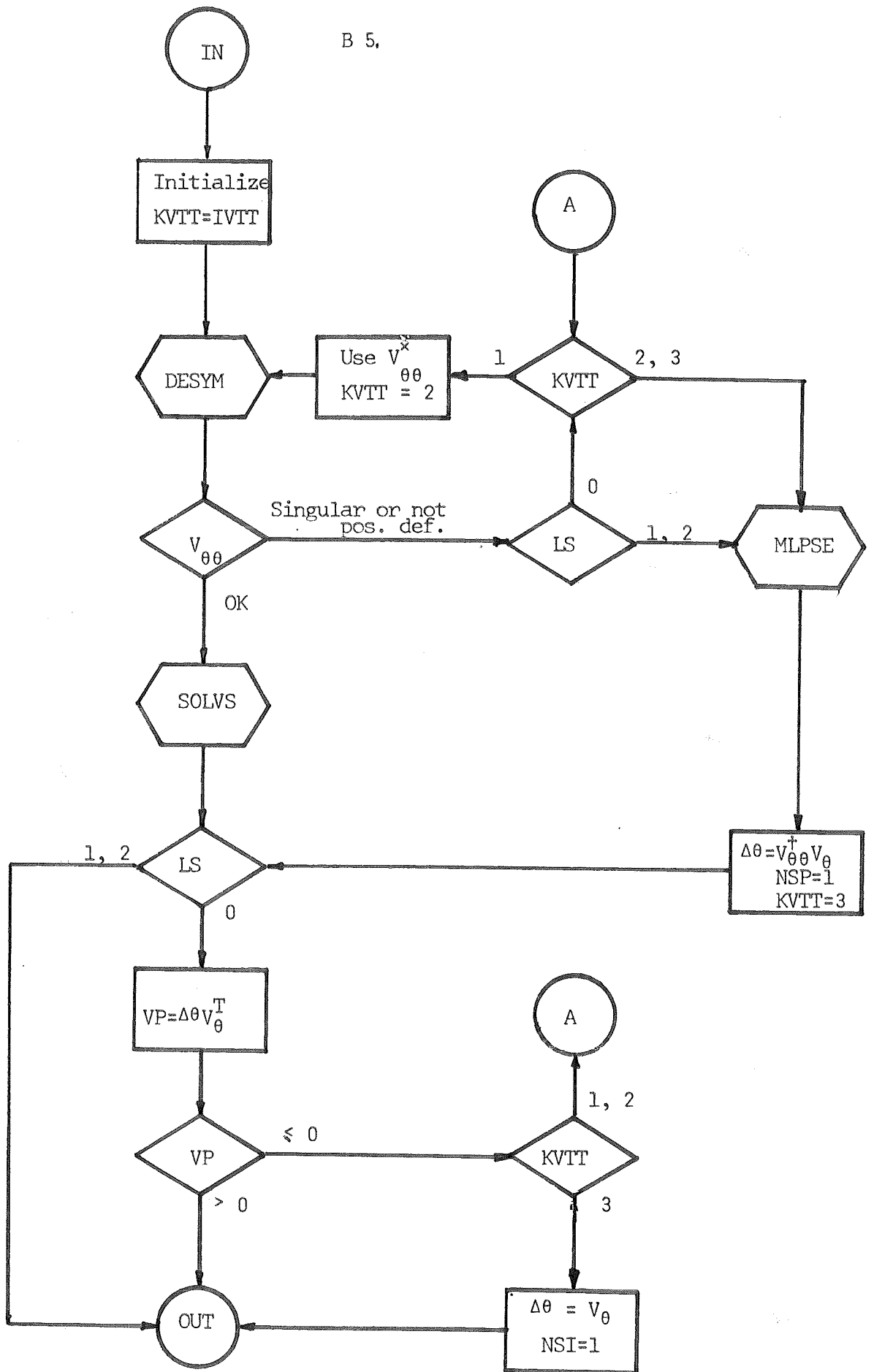


Fig B 5 - SUBROUTINE MLSOL

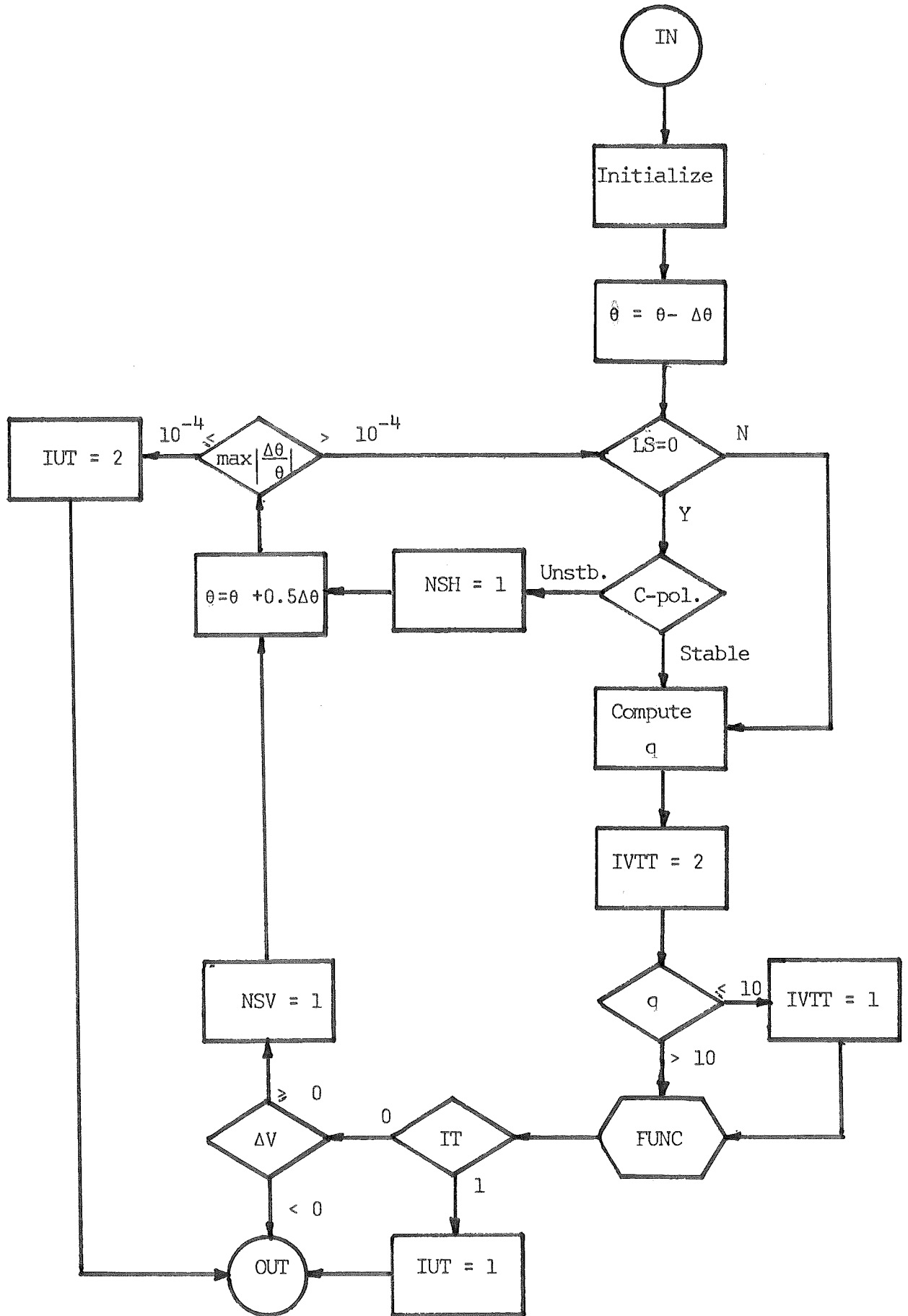


Fig B 6 - SUBROUTINE MLCOR

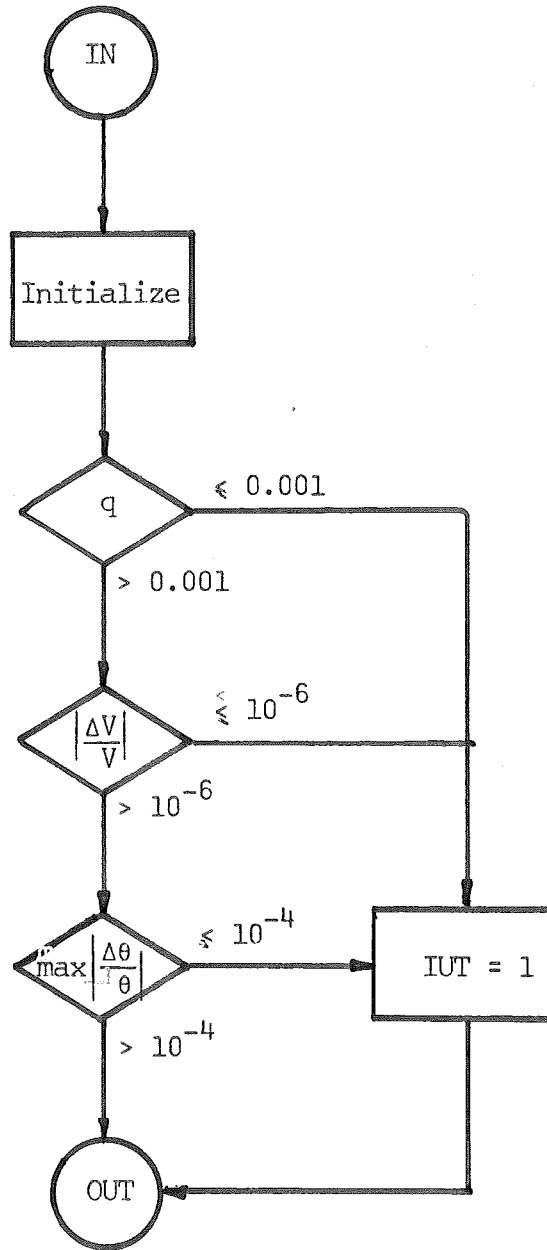


Fig B 7 - SUBROUTINE MLCON

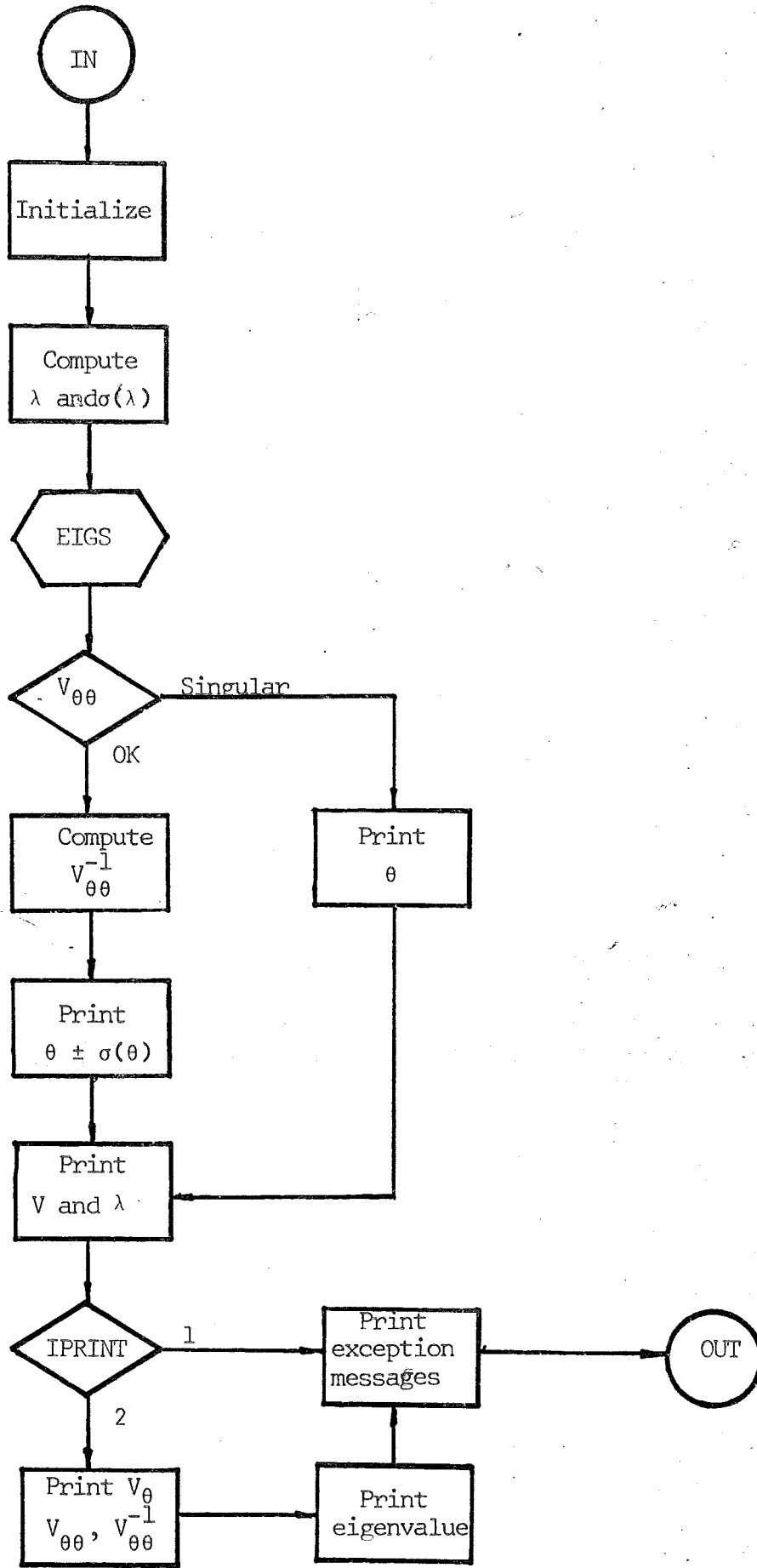


Fig B 8 - SUBROUTINE MLPRI

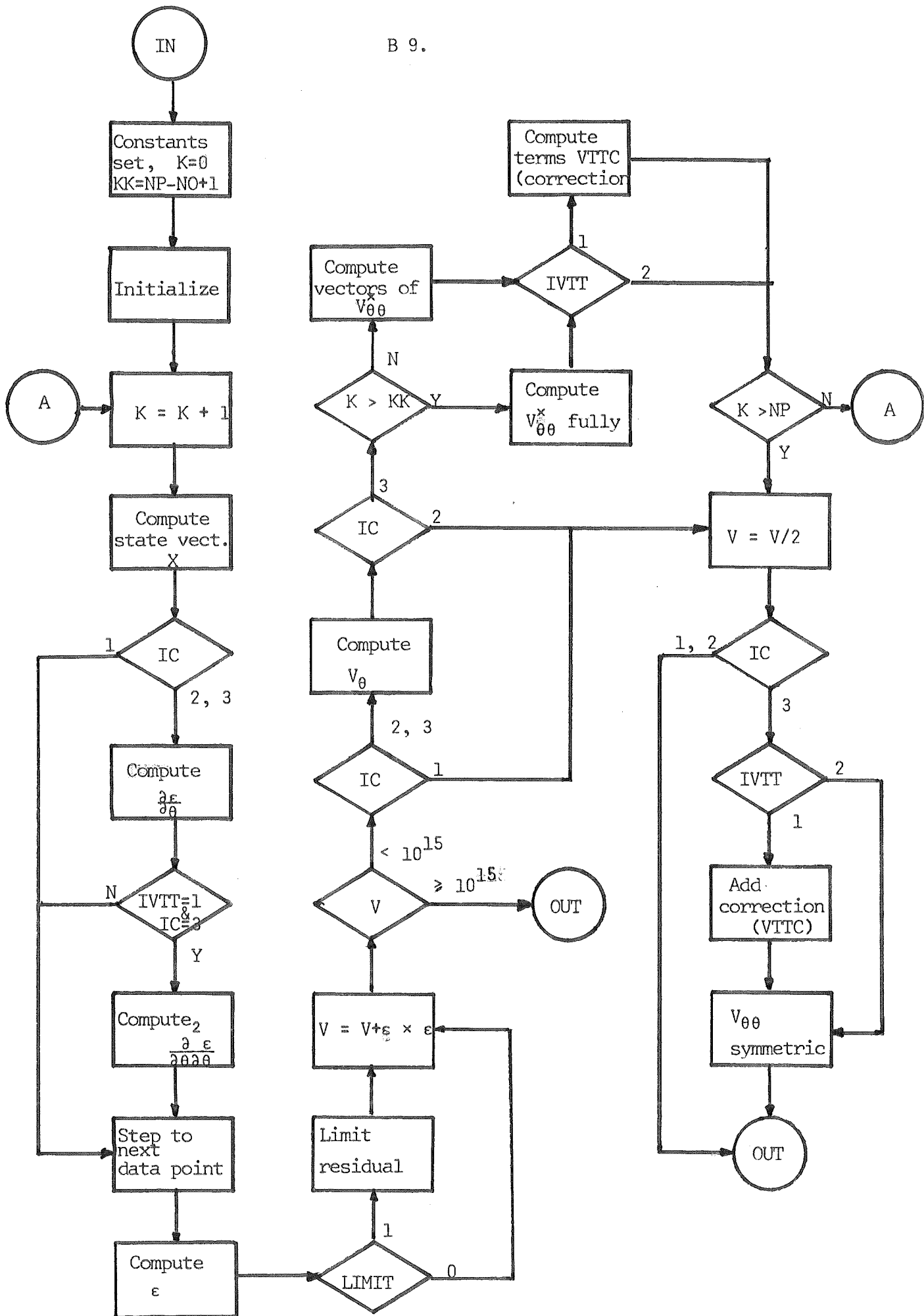


Fig B 9 - SUBROUTINE FUNC

```

C PROGRAM MLADM
C
C MLADM STANDS FOR MAXIMUM-LIKELIHOOD ADMINISTRATION,
C I.E. THIS PROGRAM ADMINISTRATES THE M-L IDENTIFICATION.
C IT CONTAINS A NUMBER OF QUESTIONS AND COMMANDS WHICH
C ARE WRITTEN ON THE TELETYPE.
C THE ANSWERS ARE THEN TAKEN AS PARAMETERS WHICH
C CONTROL THE EXECUTION OF THE IDENTIFICATION.
C THE HILL-CLIMBING ROUTINE STARTS BY CALL MLCAL.
C REFERENCE ALMQUIST R. PROGRAM FOR MAXIMUM LIKELIHOOD
C IDENTIFIERING PA PDP-15
C AUTHOR R. ALMQUIST 1971-09-20
C
C NO -ORDER OF MODEL
C NI -NUMBER OF INPUT SIGNALS (MIN 0, MAX 8)
C NP -NUMBER OF DATA POINTS
C INIT -INIT-VALUE CODE
C 0 NO ESTIMATION OF INIT-VALUES WANTED
C 1 ESTIMATION OF INIT-VALUES WANTED
C IVTT -PHASE CODE
C 1 EXACT SECOND DERIVATIVES WILL BE USED
C 2 APPROXIMATIVE SEC. DERIVATIVES WILL BE USED
C LS -LEAST-SQUARES ESTIMATION INDICATOR
C 0 NO LS EASTIMATION
C 1 LS ESTIMATION IN THE FIRST ITERATION
C 2 LS ESTIMATION. STARTING VALUES FOR C-PARAMETERS
C WANTED
C IT -INDICATES NUMBER OF ITERATIONS
C NOTE: ONLY THESE TWO ALTERNATIVES
C 0 THE ALGORITHM RUNS UNTIL CONVERGENCE
C 1 ONE ITERATION
C IPRINT -PARAMETER DETERMINING WHICH KIND OF
C PRINTING BETWEEN THE ITERATIONS.
C 0 NO PRINTING
C 1 COEFFECIENT AND STANDARD DEVIATION
C ESTIMATES, LOSS FUNCTION, ESTIMATION
C OF PREDICTION ERROR AND ITS STANDARD DEV.
C 2 SAME AS IPRINT=1 BUT ALSO GRADIENT OF LOSS
C FUNCTION, THE MATRIX WITH THE SECOND DERIVATIVES
C OF THE LOSS FUNCTION AND THE INVERSE OF THIS
C MATRIX.
C NOTE: STARTING VALUES,LS-ESTIMATION AND
C FINAL ESTIMATION ARE ALWAYS PRINTED.
C IFIX -FIX CODE
C 0 NO FIXED PARAMETERS
C 1 FIXED PARAMETERS
C LIMIT -LIMIT CODE
C 0 RESIDUALS NOT LIMITED
C 1 LIMITED RESIDUALS
C
C IF CNTRL P IS TYPED AT ANY POINT, AFTER CALL CNTRLP,
C THE EXECUTION RESTARTS FROM "DATA FOR IDENTIFICATION
C ROUTINES".
C
C SUBROUTINES REQUIRED
C MLCAL
C CREAFI
C MLCAL
C REANS
C
C DIMENSION NRA(20), IDUM(36), IDUMY(968)
C
C COMMON/SLASK/IDUM, NRA, IDUMY
C
C COMMON/B/NO, NI, NP, INIT, IVTT, LS, IT, IFIX, LIMIT

```

SUBROUTINE MLCAL(IPRINT)

C
 C MLCAL STANDS FOR MAXIMUM-LIKELIHOOD CALLS.
 C IT DETERMINES THE ORDER OF THE CALLS OF THE
 C SUBROUTINES MLINI,MLSOL,MLCOR,MLCON,MLPRI. I.E.,
 C WHEN THE USER HAS GIVEN HIS COMMANDS, MLCAL IS
 C SUPERVISING THE ML-IDENTIFICATION.
 C REFERENCE ALMQUIST R. PROGRAM FOR MAXIMUM LIKELIHOOD
 C IDENTIFYING PA PDP-15
 C AUTHOR R.ALMQUIST 1971-09-16
 C
 C IPRINT -SEE PROGRAM HEAD OF MLADM.
 C
 C CONTENTS OF COMMON/SLASK/
 C ALL BLOCKS ARE USED.
 C T -COEFFICIENT ARRAY
 C V -LOSS FUNCTION
 C VT -GRADIENT OF LOSS FUNCTION
 C VTT -THE SECOND DERIVATIVE MATRIX
 C DT -THE COEFF. CORRECTION ARRAY
 C VOLD -THE OLD V-VALUE
 C NRA -ARRAY CONTAINING INFORMATION ABOUT WHICH
 C PARAMETERS THAT ARE TO BE FIXED
 C
 C CONTENTS OF COMMON/B/
 C -SEE PROGRAM HEAD OF MLADM
 C
 C CONTENTS OF COMMON/BB/
 C IUT -IF IUT=1 RETURN TO MLADM
 C NSH -IF NSH=1 THE COEFF. CORRECTION WAS HALVED
 C IN THE LAST ITERATION
 C NSP -IF NSP=1 THE PSEUDO-INVERSE OF VTT WAS USED
 C IN THE LAST ITERATION
 C NSI -IF NSI=1 THEN DT=VT WAS USED IN THE LAST
 C ITERATION
 C
 C SUBROUTINES REQUIRED
 C MLINI
 C MLSOL
 C MLCOR
 C MLCON
 C MLPRI
 C
 C DIMENSION VTT(20,20),VT(20),DT(20),T(20),IDUM(36)
 C DIMENSION IDUMY(44),NRA(20)
 C
 C COMMON/B/NO,NI,NP,INIT,IVTT,LS,IT,IFIX,LIMIT
 C COMMON/BB/IUT,NSH,NSI,NSP
 C COMMON/SLASK/IDUM,NRA,VT,VOLD,DT,T,V,VTT,IDUMY
 C

SUBROUTINE MLINI

C
C
C INITIATES THE MAXIMUM-LIKELIHOOD IDENTIFICATION, BY PUTTING
C WANTED STARTING VALUES TO THE PARAMETERS.
C
C MAKES THE FIRST CALL TO FUNC.
C
C ADJUSTS, IF LS-ESTIMATION IS WANTED, THE SEC. DER. MATRIX
C AND THE GRADIENT OF V.
C
C AUTHOR, R.ALMQUIST 1971-09-15

C
C CONTENTS OF COMMON FIELDS: SEE MLADM AND MLCAL

C
C SUBROUTINE REQUIRED

C
C FUNC

C
C DIMENSION VTT(20,20),VT(20),T(20),D(4),IDUM(56),DT(20)
C DIMENSION IDUMY(44)

C
C COMMON/B/NO,NI,NP,INIT,IVTT,LS,IT,IFIX,LIMIT
C COMMON/BR/IUT,NSH,NSI,NSP
C COMMON/SLASK/IDUM,VT,VOLD,DT,T,V,VTT,IDUMY

SUBROUTINE MLSOL

C
C
C MLSOL STANDS FOR MAXIMUM-LIKELIHOOD SOLUTION.
C
C COMPUTES THE CORRECTION DT BY SOLVING EITHER
C
C $(VTT-APPR.)*DT=VT$ OR $(VTT-EX.)*DT=VT$ OR $DT=(VTT-PSEUDO)*VT$
C OR $DT=VT$ DEPENDING ON THE DIFFERENT LOCAL PROPERTIES OF THE
C
C LOSS FUNCTION
C
C AUTHOR, R.ALMQUIST 1971-09-15

C
C CONTENT OF COMMON/A/

C
C VTTC -AN ARRAY CONSISTING OF THE NON-QUADRATIC
C
C ELEMENTS OF THE SECOND DERIVATIVE MATRIX
C
C I.E. THESE ELEMENTS MAKE THE DIFFERENCE
C
C BETWEEN THE EXACT AND THE APPROXIMATIVE SEC.
C
C DERIVATIVE MATRIX

C
C CONTENTS OF OTHER COMMON FIELDS

C
C -SEE PROGRAM HEADS OF MLADM AND MLCAL

C
C SUBROUTINE REQUIRED

C
C MLPSE

C
C DIMENSION VTT(20,20),VT(20),T(20),DT(20),G(20,20)
C DIMENSION SDT(20,1),IDUM(36),SVT(20,1),NRA(20),IDUMY(44)

C
C COMMON/A/VTTC(40)

C
C COMMON/B/NO,NI,NP,INIT,IVTT,LS,IT,IFIX,LIMIT
C COMMON/BR/IUT,NSH,NSI,NSP
C COMMON/SLASK/IDUM,NRA,VT,VOLD,DT,T,V,VTT,IDUMY

SUBROUTINE MLCOR

```

C
C   MLCOR STANDS FOR MAXIMUM-LIKELIHOOD CORRECTION.
C   COMPUTES CORRECTED PARAMETERS, T(I)=T(I)-DT(I).
C   TESTS IF C-POLYNOMIAL STABLE.
C   DECIDES WHICH PHASE. (VTT.APPR. OR VTT-EX.)
C   CALLS FUNC TO GET NEW V,VT AND VTT
C   TESTS IF THE LOSS FUNCTION DECREASES.
C   AUTHOR, R.ALMQUIST 1971-09-15
C
C   CONTENTS OF COMMON FIELDS
C       -SEE PROGRAM HEADS OF MLADM AND MLCAL
C
C   SUBROUTINE REQUIRED
C       FUNC
C
C   DIMENSION VTT(20,20),T(20),DT(20),C(6),VT(20),IDUM(34)
C   DIMENSION IDUMY(44)
C   DIMENSION NRA(20)
C
C   COMMON/B/NO,NI,NP,INIT,IVTT,LS,IT,IFIX,LIMIT
C   COMMON/BB/IUT,NSH,NSI,NSP
C   COMMON/SLASK/IDUM,Q,NRA,VT,VOLD,DT,T,V,VTT,IDUMY
C

```

SUBROUTINE MLCON

```

C
C   MLCON STANDS FOR MAXIMUM-LIKELIHOOD CONVERGENCE
C   COMPUTES TESTS OF CONVERGENCE IN THE MAXIMUM-LIKELIHOOD
C   IDENTIFICATION ALGORITHM.
C   AUTHOR, R. ALMQUIST 1971-15-09.
C
C   CONTENTS OF COMMON FIELDS
C       -SEE PROGRAM HEADS OF MLADM AND MLCAL
C
C   SUBROUTINE REQUIRED
C       NONE
C
C   DIMENSION VTT(20,20),VT(20),T(20),DT(20),IDUM(34),NRA(20)
C   DIMENSION IDUMY(44)
C
C   COMMON/B/NO,NI,NP,INIT,IVTT,LS,IT,IFIX,LIMIT
C   COMMON/BB/IUT,NSH,NSI,NSP
C   COMMON/SLASK/IDUM,Q,NRA,VT,VOLD,DT,T,V,VTT,IDUMY
C

```

SUBROUTINE MLPRI(IPRINT)

MLPRI STANDS FOR MAXIMUM-LIKELIHOOD PRINTING.
IT HANDLES THE PRINTING ON LP BETWEEN THE ITERATIONS
IN THE M-L IDENTIFICATION.
AUTHOR, R. ALMQUIST 1971-08-31

IPRINT- PARAMETER DETERMINING WHICH KIND OF PRINTING
 BETWEEN THE ITERATIONS.
IPRINT=0 NO PRINTING BETWEEN THE ITERATIONS
IPRINT=1 T(1) +- ST. DEV. AND V, LAMBDA +- ST. DEV.
 FOR EACH ITERATION.
IPRINT=2 T(1) +- ST. DEV. AND V, LAMBDA, VT, VTT, VTTI
 FOR EACH ITERATION.

CONTENTS OF COMMON FIELDS

-SEE PROGRAM HEADS OF MLADM AND MLCAL

SUBROUTINES REQUIRED

NONE

DIMENSION VTT(20,20),VTTI(20,20),VT(20),T(20),EV(20),R(20,20)
DIMENSION IDUM(36),D(4),DT(20),NRA(20),IDUMY(44),BN(5)

COMMON/B/NO,NI,NP,INIT,IVTT,LS,IT,IFIX,LIMIT
COMMON/BR/IUT,NSH,NSI,NSP
COMMON/SLASK/IDUM,NRA,VT,VOLD,DT,T,V,VTT,IDUMY

SUBROUTINE CREATI

CREATES A FILE IODATA EXT, CONTAINING INPUT AND OUTPUT
DATA, STORED U1(1),U2(1),...,UNI(1),Y(1),...,UNI(NP),Y(NP).
INPUT 1 IS A PRBS WITH PERIOD=63 AND AMPLITUDE 1.0.
INPUT 2 IS PRBS WITH PERIOD 127 AND AMP.=1.0
THE OUTPUT SIGNAL IS GENERATED FROM:
$$Y(T)+A(1)*Y(T-1)+,\dots,+A(NO)*Y(T-NO)=B1(1)*U1(T-1)+,\dots,+$$

$$BNI(NO)*UNI(T-NO)+AL*(E(T)+C(1)*E(T-1)+,\dots,+C(NO)*E(T-NO))$$

WHERE E(T) IS NOISE DISTRIBUTED ACCORDING TO THE NORMAL (0,1)
DISTRIBUTION.

AUTHOR R. ALMQUIST 1971-07-29.

NO -ORDER OF SYSTEM
NI -NUMBER OF INPUT SIGNALS (MAX 2)
NP -NUMBER OF DATA POINTS
T -COEFF. VECTOR CONSISTING OF THE A,B1,...,BNI AND C
 COEFFECIENTS
AL -LAMBDA
INIT -INITIAL VALUE INDICATOR
 0 NO INIT-VALUES FOR Y
 1 INIT-VALUES WANTED

SUBROUTINE REQUIRED

NONE

DIMENSION FILE(2),Y(6),U(30),E(6),T(20),OT(12),BN(2),D(3)

DATA FILE(1),FILE(2)/5HIODAT,4HAEXT/

SUBROUTINE FUNC(T,NPAR,V,VT,VTT,IA,IC)

COMPUTES LOSS FUNCTION, GRADIENT AND SECOND
DERIVATIVES OF LOSS FUNCTION.

REFERENCE GUSTAVSSON J. PARAMETRIC IDENTIFICATION OF
MULTIPLE INPUT SINGLE OUTPUT DYNAMIC SYSTEMS,
AUTHOR, J. GUSTAVSSON,
REVISED, R. ALMQUIST 71-10-10

T -PARAMETER ARRAY
NPAR -DIMENSION PARAMETER OF T
V -LOSS FUNCTION
VT -VECTOR OF GRADIENT OF LOSS FUNCTION.
VTT -MATRIX OF SECOND DERIVATIVES OF LOSS FUNCTION
IA -DIMENSION PARAMETER OF VTT,
IC -INPUT CODE
1 COMPUTE V ONLY
2 COMPUTE V AND VT
3 COMPUTE V,VT AND VTT

CONTENTS OF COMMON/B/

NO -ORDER OF MODEL
NI -NUMBER OF INPUT SIGNALS (MIN 0, MAX 8)
INIT -INIT-VALUE CODE
0 NO ESTIMATION OF INIT-VALUES WANTED
1 ESTIMATION OF INIT-VALUES WANTED
IVTT -INDICATOR DETERMINING WHICH KIND OF SEC. DER.
THAT IS WANTED
1 EXACT SEC. DER.
2 APPROXIMATIVE SEC. DER.
ID1 -DUMMY
ID2 -DUMMY
ID3 -DUMMY
LIMIT -INDICATES IF THE RESIDUALS ARE TO BE LIMITED
0 RESIDUALS NOT LIMITED
1 LIMITED RESIDUALS

CONTENT OF COMMON/A/

VTTG -AN ARRAY CONSISTING OF THE NON-QUADRATIC ELEMENTS
OF THE SECOND DERIVATIVE MATRIX, I.E. THESE
ELEMENTS MAKES THE DIFFERENCE BETWEEN THE EXACT
AND THE APPROXIMATIVE SEC. DER.

REQUIRE DATA IN FILE IODATA EXT IN ORDER

U1(1),U2(1),...,UNI(1),Y(1),U1(2),
U2(2),...,UNI(2),Y(2),...,U1(NP),U2(NP),...,UNI(NP),Y(NP)

SUBROUTINE REQUIRED

NONE

DIMENSION X(5),XA(11),XB(11),XC(6),T(1),VT(1),VTT(1,1),U(8)
DIMENSION ET(20),ETT(40),FILE(2),FIL(2)

COMMON/A/VTTG(40)

COMMON/B/NO,NI,NP,INIT,IVTT,ID1,ID2,ID3,LIMIT

DATA FILE(1),FILE(2)/5HIODAT,4HAEXT/,FIL(1),FIL(2)/5HRESID,
14HUEXT/

SUBROUTINE MLPSE

C
 C COMPUTES THE PSEUDO-INVERSE OF A SYMMETRIC MATRIX VTT AS
 C $VTT-PSE=R*S*RT$, R IS A MATRIX CONTAINING THE COLUMNS
 C OF EIGENVECTORS OF VTT, RT IS R TRANSPOSE, S IS A
 C DIAGONAL MATRIX CONTAINING THE INVERSES OF THE
 C EIGENVALUES $> EPS$, AND THE OTHER ELEMENTS OF THE DIAGONAL
 C ARE EQUAL TO ZERO.
 C VTT IS DESTROYED AND REPLACED BY ITS PSEUDO-INVERSE.
 C
 C AUTHOR,R.ALMQUIST 1971-07-18
 C
 C CONTENTS OF COMMON FIELDS: SEE MLADM AND MLCAL
 C
 C SUBROUTINE REQUIRED
 C NONE
 C
 C DIMENSION R(20,20),EV(20),VTT(20,20),DT(20),VT(20),IDUM(56)
 C DIMENSION IDUMY(44),T(20)
 C
 C COMMON/B/NO,NI,NP,INIT,IVTT,LS,IT,IFIX,LIMIT
 C COMMON/SLASK/IDUM,VT,VOLD,DT,T,V,VTT,IDUMY

SUBROUTINE MLGID

C
 C
 C WRITES THE ML-IDENTIFIERS GUIDE ON LP
 C THE GUIDE IS READ FROM MLGUID EXT
 C
 C AUTHOR R. ALMQUIST 1972-01-08
 C
 C DIMENSION FIML(2),OT(16)
 C DATA FIML(1),FIML(2)/5HMLGUI,4HDEXT/
 C

APPENDIX C

```

*****
*THE ML-IDENTIFIERS GUIDE*
*****

```

DATA FOR THE IDENTIFICATION CAN BE SIMULATED IN THE PROGRAM OR SHOULD BE GIVEN IN THE DATA FILE IODATA EXT IN ORDER U1(1),U2(1),...,UN1(1),Y(1),...,UN1(NP),Y(NP). FOR THE GENERATION OF THIS FILE USE STDAT. THE RESIDUALS ARE STORED IN THE DATA FILE RESIDU EXT.

DATA FOR THE SIMULATION PROGRAM:

```

NO      -SYSTEM ORDER
NI      -NUMBER OF INPUTS (1 OR 2)
NP      -NUMBER OF SAMPLES
INIT    -INITIAL VALUES INDICATOR.
         0  INITIAL VALUES OF Y(T) = 0
         1  TYPE OTHER INITIAL VALUES

```

MODEL: $Y(T)+A(1)*Y(T-1)+\dots+A(NO)*Y(T-NO) = B1(1)*U1(T-1) + \dots + B2(NO)*U2(T-NO) + LAMBDA*(E(T)+C(1)*E(T-1)+\dots+C(NO)*E(T-NO))$

E(T) INDEPENDENT NORMAL (0,1)
 U1(T) PRBS WITH PERIOD 63 AND AMPLITUDE 1.0
 U2(T) PRBS WITH PERIOD 127 AND AMPLITUDE 1.0

DATA FOR IDENTIFICATION ROUTINES:

```

NO      -ORDER OF MODEL
NI      -NUMBER OF INPUT SIGNALS (MIN 0, MAX 8)
NP      -NUMBER OF SAMPLES
INIT    -INDICATES ESTIMATION OF INITIAL VALUES
         0  NO ESTIMATION OF INITIAL VALUES
         1  ESTIMATION OF INITIAL VALUES
NOTE:   NO*(2+NI)+INIT)<21
LS      -LEAST-SQUARES ESTIMATION INDICATOR
         0  NO LS-ESTIMATION
         1  LS-ESTIMATION IN THE FIRST ITERATION
         2  LS-ESTIMATION IN THE FIRST ITERATION WITH STARTING
            VALUES FOR THE C-PARAMETERS
IT      -INDICATES NUMBER OF ITERATIONS
         0  THE PROGRAM RUNS UNTIL CONVERGENCE OR MAX 5 MINUTES
            (CAN BE RESTARTED)
         1  ONE ITERATION ONLY
IPRINT  -PARAMETER DETERMINING OUTPUT OF THE PROGRAM
         0  NO PRINTING BETWEEN THE ITERATIONS. STARTING VALUES
            LS ESTIMATION AND FINAL ESTIMATION ARE PRINTED
         1  COEFFICIENT AND STANDARD DEVIATION ESTIMATES,
            LOSS FUNCTION, ESTIMATION OF PREDICTION ERROR
            PRINTED FOR EACH ITERATION
         2  AS 1 BUT GRADIENT, SECOND DERIVATIVES MATRIX,
            INVERSE OF SEC. DER. MATRIX AND THE EIGENVALUES OF
            THE SEC. DER. MATRIX ARE ALSO PRINTED

```

THE VALUES OF THE RESIDUALS MAY BE LIMITED TO $3 \cdot \text{LAMBDA}$, WHERE LAMPDA IS THE ESTIMATION OF THE PREDICTION ERROR IN THE PREVIOUS ITERATION.

SOME PARAMETERS MAY BE FIXED TO PREDETERMINED VALUES. TYPE E.G.

A 2
B12
C 3

IF $N1=1$ B 1 CAN BE TYPED INSTEAD OF B11.

AFTER AN IDENTIFICATION OF A DATA SET WITH A MODEL OF ORDER $N1$ IT IS POSSIBLE TO IDENTIFY THE SAME DATA SET WITH A MODEL OF ORDER $N2$. IT IS ALSO POSSIBLE TO JUMP TO THE BEGINNING OF THE PROGRAM OR TO THE MONITOR.

IF CNTRL P IS TYPED THE EXECUTION RESTARTS FROM "DATA FOR IDENTIFICATION ROUTINES.

INPUT-OUTPUT DATA

1.000	1.000	0.000	1.000	0.000	1.000	1.000	1.000	3.629	1.000	1.000	-1.000	5.817
1.000	-1.000	7.170	-1.000	7.827	1.000	1.000	1.000	5.164	-1.000	-1.000	-1.000	5.933
1.000	1.000	1.390	-1.000	1.273	1.000	1.000	1.000	-0.494	1.000	1.000	1.000	-3.061
-1.000	1.000	1.791	-1.000	1.882	1.000	1.000	1.000	2.177	1.000	1.000	-1.000	-0.164
-1.000	1.000	-1.288	1.000	-1.241	1.000	1.000	1.000	0.099	1.000	1.000	-1.000	4.235
-1.000	-1.000	5.723	-1.000	2.431	1.000	1.000	1.000	2.664	-1.000	-1.000	1.000	1.654
1.000	1.000	1.225	-1.000	2.140	1.000	1.000	1.000	4.452	1.000	1.000	1.000	2.994
-1.000	1.000	2.167	-1.000	0.675	1.000	1.000	1.000	-1.216	1.000	1.000	-1.000	-6.715
1.000	-1.000	-3.235	-1.000	-3.257	1.000	1.000	1.000	-0.732	-1.000	-1.000	1.000	1.843
1.000	-1.000	3.171	-1.000	1.347	1.000	1.000	1.000	-0.196	1.000	1.000	-1.000	0.615
1.000	-1.000	-0.925	1.000	2.161	-1.000	1.000	1.000	3.018	-1.000	-1.000	-1.000	4.671
1.000	1.000	2.242	-1.000	0.082	-1.000	1.000	1.000	-0.578	-1.000	-1.000	1.000	1.550
-1.000	-1.000	1.322	-1.000	-3.790	1.000	1.000	1.000	-6.590	1.000	1.000	1.000	-3.829
-1.000	1.000	0.820	-1.000	1.358	-1.000	1.000	1.000	3.347	-1.000	-1.000	1.000	-3.185
1.000	-1.000	-3.099	-1.000	-6.124	1.000	1.000	1.000	-5.756	-1.000	-1.000	1.000	-4.047
-1.000	-1.000	-3.639	-1.000	-7.168	1.000	1.000	1.000	-6.956	1.000	1.000	-1.000	-8.349
1.000	1.000	-3.257	1.000	-0.610	-1.000	1.000	1.000	6.015	1.000	1.000	1.000	10.396
-1.000	-1.000	12.989	1.000	11.622	-1.000	1.000	1.000	8.506	1.000	1.000	1.000	8.161
-1.000	1.000	5.814	-1.000	4.422	1.000	1.000	1.000	-1.046	-1.000	-1.000	1.000	-0.921
-1.000	1.000	-3.691	1.000	-3.167	1.000	1.000	1.000	-1.237	-1.000	-1.000	-1.000	-1.279
1.000	1.000	-1.249	-1.000	1.470	1.000	1.000	1.000	1.761	-1.000	-1.000	-1.000	5.101
1.000	1.000	4.580	1.000	3.084	-1.000	1.000	1.000	6.964	1.000	1.000	-1.000	4.419
-1.000	1.000	2.985	-1.000	1.816	1.000	1.000	1.000	-0.244	-1.000	-1.000	-1.000	-5.351
-1.000	-1.000	-3.942	1.000	-7.475	-1.000	1.000	1.000	-8.442	1.000	1.000	1.000	-5.178
-1.000	1.000	-1.549	-1.000	2.968	-1.000	1.000	1.000	3.047	1.000	1.000	1.000	1.926
-1.000	-1.000	0.672	1.000	0.112	1.000	1.000	1.000	-0.677	-1.000	-1.000	1.000	1.230
1.000	1.000	4.556	-1.000	8.623	-1.000	1.000	1.000	8.997	1.000	1.000	1.000	2.488
-1.000	-1.000	1.304	1.000	-1.025	-1.000	1.000	1.000	-3.760	-1.000	-1.000	1.000	-6.776
-1.000	1.000	-5.018	1.000	-6.400	1.000	1.000	1.000	-6.845	-1.000	-1.000	1.000	-5.812
-1.000	-1.000	-2.655	-1.000	-0.394	-1.000	1.000	1.000	-0.527	1.000	1.000	1.000	-4.497
-1.000	-1.000	-5.634	-1.000	-7.336	-1.000	1.000	1.000	-8.077	-1.000	-1.000	-1.000	-11.509
-1.000	1.000	-7.885	1.000	-8.042	1.000	1.000	1.000	-8.273	1.000	1.000	-1.000	-3.141
1.000	-1.000	5.000	-1.000	7.600	1.000	1.000	1.000	12.573	-1.000	-1.000	-1.000	10.657
1.000	1.000	8.213	-1.000	6.047	1.000	1.000	1.000	2.607	-1.000	-1.000	1.000	-1.111

Data till testexempel 2.

1.000	-1.000	0.856	1.000	-1.000	-4.453	-1.000	1.000	-0.765	-1.000	1.000	0.284
1.000	1.000	-0.858	1.000	1.000	-1.689	-1.000	-1.000	1.018	1.000	1.000	2.342
1.000	1.000	5.425	1.000	1.000	6.019	-1.000	-1.000	11.133	1.000	-1.000	5.048
1.000	-1.000	5.662	-1.000	-1.000	1.615	1.000	1.000	-0.474	-1.000	1.000	-0.338
-1.000	1.000	-0.223	1.000	1.000	-0.393	-1.000	1.000	-0.091	-1.000	1.000	0.697
1.000	1.000	-1.955	1.000	-1.000	-1.456	1.000	-1.000	2.717	-1.000	-1.000	2.219
-1.000	1.000	1.458	-1.000	1.000	0.971	1.000	1.000	-1.130	-1.000	-1.000	-0.671
1.000	1.000	-1.977	1.000	1.000	0.328	1.000	-1.000	1.423	1.000	-1.000	5.101
-1.000	-1.000	6.715	-1.000	1.000	4.223	1.000	-1.000	2.707	-1.000	1.000	-1.594
1.000	-1.000	-2.195	-1.000	-1.000	-1.544	-1.000	1.000	-2.367	-1.000	-1.000	-6.052
1.000	1.000	-5.885	1.000	1.000	-5.035	-1.000	1.000	0.431	-1.000	1.000	2.318
-1.000	1.000	4.169	-1.000	-1.000	0.723	1.000	-1.000	-2.906	-1.000	-1.000	-5.176
-1.000	1.000	-4.566	-1.000	1.000	-3.686	-1.000	1.000	-7.128	-1.000	-1.000	-8.269
1.000	-1.000	-6.566	1.000	-1.000	-6.661	1.000	-1.000	-5.402	1.000	1.000	0.870
1.000	-1.000	2.729	1.000	1.000	7.958	-1.000	1.000	11.221	-1.000	-1.000	11.651
-1.000	1.000	9.971	-1.000	1.000	6.661	1.000	-1.000	2.820	1.000	1.000	1.549
1.000	-1.000	1.319	-1.000	1.000	1.053	-1.000	1.000	-0.410	-1.000	-1.000	-1.065
1.000	1.000	-0.709	-1.000	-1.000	4.351	1.000	-1.000	3.096	1.000	1.000	3.102
1.000	-1.000	4.133	-1.000	1.000	3.020	1.000	-1.000	3.562	1.000	1.000	1.862
-1.000	1.000	2.704	1.000	-1.000	2.381	-1.000	-1.000	3.267	-1.000	1.000	3.209
1.000	-1.000	-1.398	-1.000	1.000	0.032	-1.000	-1.000	-2.108	1.000	-1.000	-3.555
1.000	1.000	-4.999	1.000	-1.000	-3.473	1.000	-1.000	0.443	-1.000	1.000	3.492
-1.000	-1.000	2.181	-1.000	1.000	0.337	-1.000	1.000	0.734	1.000	-1.000	-3.559
1.000	1.000	-4.092	1.000	-1.000	-1.351	1.000	-1.000	0.907	-1.000	1.000	3.274
-1.000	1.000	8.357	-1.000	1.000	4.737	-1.000	1.000	3.352	1.000	-1.000	0.584
-1.000	-1.000	0.343	-1.000	1.000	-1.790	1.000	-1.000	-4.491	1.000	1.000	-6.993
1.000	-1.000	-4.688	-1.000	1.000	-2.409	-1.000	1.000	-0.046	-1.000	-1.000	0.236
-1.000	-1.000	-5.298	1.000	-1.000	-5.948	-1.000	-1.000	-8.364	-1.000	-1.000	-5.522
-1.000	-1.000	-6.341	-1.000	1.000	-6.335	-1.000	-1.000	-8.838	1.000	-1.000	-9.807
1.000	-1.000	-7.355	1.000	-1.000	-4.895	1.000	-1.000	2.621	-1.000	1.000	7.140
1.000	1.000	9.439	-1.000	1.000	11.707	-1.000	-1.000	10.969	-1.000	1.000	7.920
1.000	1.000	3.574	-1.000	1.000	3.948	-1.000	1.000	1.250	-1.000	-1.000	-0.634
-1.000	-1.000	-4.435	-1.000	1.000	-5.379	1.000	-1.000	-7.307	1.000	1.000	-3.737
-1.000	1.000	-0.267	1.000	-1.000	2.457	-1.000	1.000	4.367	-1.000	1.000	5.706

Data till testexempel 2. (forts)

INPUT AND OUTPUT

-116.864	-28.185	-108.864	-80.185	-106.864	-94.185	-107.864	-104.185	-110.864	-111.185
123.136	-117.185	122.136	58.815	128.136	91.815	132.136	105.815	-89.864	117.815
-113.864	-35.185	121.136	-85.185	129.136	84.815	-88.864	115.815	-114.864	-34.185
125.136	-87.185	127.136	81.815	-84.864	111.815	-107.864	-38.185	129.136	-93.185
130.136	77.815	139.136	107.815	141.136	117.815	141.136	126.815	140.136	130.815
140.136	136.815	137.136	139.815	135.136	142.815	137.136	145.815	-85.864	146.815
-110.864	-13.185	122.136	-73.185	123.136	91.815	137.136	115.815	140.136	122.815
-86.864	126.815	-110.864	-29.185	-106.864	-85.185	-107.864	-102.185	-178.864	-112.185
122.136	-170.185	133.136	84.815	135.136	93.815	-87.864	101.815	-111.864	-52.185
-112.864	-105.185	-102.864	-119.185	-103.864	-129.185	-115.864	-136.185	132.136	-140.185
134.136	33.815	134.136	70.815	144.136	85.815	141.136	97.815	138.136	106.815
-80.864	112.815	-107.864	-41.185	126.136	-96.185	133.136	70.815	138.136	96.815
138.136	106.815	139.136	113.815	138.136	110.815	141.136	122.815	141.136	126.815
145.136	127.815	144.136	128.815	-82.864	129.815	-108.864	-28.185	-104.864	-88.185
-101.864	-105.185	130.136	-118.185	129.136	51.815	142.136	81.815	140.136	91.815
-84.864	97.815	-109.864	-55.185	131.136	-109.185	125.136	57.815	-86.864	85.815
-108.864	-63.185	-103.864	-117.185	-105.864	-128.185	126.136	-137.185	128.136	35.815
-87.864	65.815	-107.864	-79.185	128.136	-132.185	126.136	36.815	141.136	66.815
140.136	78.815	-84.864	87.815	-101.864	-63.185	129.136	-114.185	129.136	52.815
-78.864	81.815	-104.864	-67.185	-105.864	-118.185	-99.864	-128.185	-99.864	-135.185
-104.864	-141.185	-102.864	-143.185	-103.864	-145.185	-107.864	-145.185	-123.864	-146.185
-119.864	-146.185	-118.864	-150.185	-116.864	-151.185	-118.864	-151.185	111.136	-151.185
115.136	28.815	120.136	71.815	121.136	91.815	131.136	105.815	121.136	116.815
-97.864	124.815	-131.864	-30.185	110.136	-85.185	108.136	84.815	-105.864	113.815
-128.864	-37.185	-121.864	-90.185	-124.864	-103.185	-126.864	-112.185	-124.864	-119.185
110.136	-123.185	107.136	54.815	114.136	90.815	114.136	106.815	114.136	116.815
115.136	127.815	112.136	136.815	117.136	143.815	118.136	148.815	114.136	153.815
115.136	156.815	115.136	156.815	120.136	156.815	115.136	155.815	-103.864	155.815
-133.864	-6.185	-178.864	-66.185	105.136	-169.185	-109.864	121.815	-134.864	-33.185

Data till testexempel 3.

-130.864	-91.185	-122.864	-107.185	104.136	-121.185	104.136	48.815	-107.864	77.815
-128.864	-69.185	106.136	-119.185	107.136	50.815	114.136	80.815	112.136	92.815
-106.864	102.815	-132.864	-49.185	100.136	-103.185	102.136	64.815	-107.864	91.815
-135.864	-56.185	-130.864	-108.185	-128.864	-120.185	105.136	-127.185	105.136	45.815
110.136	81.815	119.136	95.815	-102.864	106.815	-129.864	-45.185	-125.864	-98.185
-126.864	-110.185	-124.864	-121.185	-124.864	-129.185	-129.864	-135.185	-129.864	-138.185
-121.864	-140.185	-122.864	-142.185	99.136	-143.185	107.136	35.815	-105.864	73.815
-133.864	-65.185	-129.864	-112.185	-125.864	-121.185	-127.864	-128.185	-126.864	-131.185
107.136	-132.185	104.136	46.815	113.136	83.815	119.136	102.815	115.136	117.815
117.136	125.815	-106.864	137.815	-131.864	-17.185	-120.864	-73.185	-122.864	-86.185
104.136	-94.185	105.136	81.815	115.136	116.815	116.136	131.815	-107.864	142.815
-131.864	-12.185	-127.864	-68.185	-128.864	-82.185	113.136	-93.185	105.136	82.815
-107.864	115.815	-130.864	-31.185	-127.864	-84.185	-124.864	-97.185	-127.864	-104.185
-128.864	-113.185	-125.864	-117.185	-125.864	-121.185	-126.864	-124.185	-129.864	-128.185
103.136	-130.185	107.136	51.815	-101.864	89.815	-137.864	-51.185	104.136	-98.185
107.136	78.815	-104.864	111.815	-129.864	-34.185	102.136	-87.185	104.136	88.815
-107.864	110.815	-135.864	-27.185	-130.864	-81.185	-129.864	-91.185	99.136	-97.185
107.136	79.815	111.136	113.815	112.136	127.815	111.136	137.815	116.136	146.815
-107.864	152.815	-131.864	-6.185	103.136	-62.185	106.136	109.815	-108.864	138.815
-133.864	-13.185	-131.864	-70.185	-124.864	-83.185	-127.864	-93.185	-124.864	-102.185
128.136	-110.185	104.136	64.815	114.136	95.815	116.136	107.815	-105.864	116.815
-128.864	-36.185	114.136	-89.185	110.136	80.815	-105.864	109.815	-131.864	-39.185
-124.864	-93.185	-122.864	-105.185	-124.864	-113.185	-123.864	-116.185	109.136	-120.185
108.136	57.815	118.136	93.815	116.136	107.815	-104.864	119.815	-126.864	-31.185
114.136	-84.185	108.136	87.815	-108.864	117.815	-127.864	-31.185	106.136	-82.185
109.136	89.815	-105.864	118.815	-130.864	-29.185	107.136	-83.185	107.136	89.815
116.136	119.815	118.136	131.815	119.136	139.815	119.136	146.815	116.136	151.815
122.136	153.815	127.136	153.815	118.136	154.815	-99.864	152.815	-126.864	-9.185
111.136	-70.185	112.136	94.815	120.136	119.815	120.136	128.815	-97.864	132.815
-123.864	-26.185	-118.864	-84.185	-121.864	-100.185	113.136	-113.185	119.136	56.815

Data till testexempel 3 (forts)

120.136	86.815	119.136	99.815	-96.864	109.815	-118.864	-43.185	-122.864	-98.185
-121.864	-111.185	-119.864	-120.185	-118.864	-127.185	110.136	-135.185	111.136	39.815
120.136	73.815	121.136	90.815	126.136	100.815	123.136	108.815	-98.864	115.815
-124.864	-40.185	112.136	-95.185	115.136	70.815	121.136	98.815	124.136	106.815
127.136	115.815	124.136	122.815	119.136	127.815	131.136	130.815	125.136	134.815
122.136	137.815	-100.864	142.815	-122.864	-16.185	-119.864	-77.185	-121.864	-95.185
114.136	-110.185	115.136	59.815	118.136	88.815	119.136	99.815	-99.864	108.815
-118.864	-44.185	114.136	-100.185	111.136	68.815	-98.864	97.815	-124.864	-52.185
-121.864	-108.185	-116.864	-122.185	116.136	-132.185	116.136	37.815	-95.864	71.815
-123.864	-69.185	110.136	-117.185	116.136	53.815	120.136	83.815	120.136	95.815
-101.864	101.815	-124.864	-50.185	110.136	-106.185	114.136	61.815	-97.864	91.815
-126.864	-55.185	-120.864	-108.185	-118.864	-119.185	-118.864	-126.185	-120.864	-133.185
-117.864	-137.185	-121.864	-139.185	-122.864	-142.185	-119.864	-143.185	-117.864	-145.185
-119.864	-147.185	-119.864	-151.185	-118.864	-154.185	115.136	-154.185	112.136	25.815
123.136	66.815	125.136	87.815	125.136	104.815	123.136	118.815	-99.864	128.815
-126.864	-26.185	111.136	-81.185	115.136	90.815	-100.864	121.815	-124.864	-29.185
-122.864	-83.185	-122.864	-95.185	-119.864	-104.185	-121.864	-113.185	110.136	-119.185
116.136	55.815	123.136	91.815	124.136	108.815	128.136	119.815	125.136	127.815
124.136	134.815	125.136	140.815	124.136	144.815	124.136	147.815	125.136	150.815
123.136	152.815	125.136	152.815	122.136	153.815	-98.864	155.815	-120.864	-6.185
113.136	-65.185	119.136	101.815	-99.864	126.815	-126.864	-27.185	-118.864	-84.185
-120.864	-101.185	110.136	-113.185	118.136	57.815	-97.864	87.815	-126.864	-58.185
108.136	-110.185	116.136	58.815	120.136	87.815	125.136	97.815	-93.864	105.815
-124.864	-47.185	109.136	-103.185	115.136	64.815	-97.864	90.815	-124.864	-59.185
-118.864	-113.185	-119.864	-124.185	112.136	-134.185	109.136	38.815	120.136	71.815
124.136	83.815	-100.864	92.815	-124.864	-58.185	-123.864	-112.185	-118.864	-123.185
-120.864	-132.185	-123.864	-136.185	-124.864	-138.185	-120.864	-140.185	-117.864	-145.185
-117.864	-148.185	110.136	-149.185	111.136	29.815	-90.864	67.815	-105.864	-71.185
-107.864	-121.185	-105.864	-130.185	-103.864	-134.185	-100.864	-137.185	133.136	-139.185
128.136	37.815	141.136	75.815	143.136	93.815	138.136	103.815	136.136	112.815

Data till testexempel 3 (forts)

-89.864	120.215	-114.864	-34.185	-110.864	-90.185	-108.864	-104.185	121.136	-114.185
132.136	59.815	136.136	92.815	134.136	102.815	-89.864	110.815	-110.864	-44.185
-110.864	-100.185	-107.864	-113.185	122.136	-122.185	125.136	50.815	-89.864	81.815
-116.864	-62.185	-109.864	-114.185	-107.864	-125.185	-104.864	-131.185	-105.864	-138.185
-106.864	-142.185	-107.864	-145.185	-109.864	-148.185	-112.864	-149.185	119.136	-150.185
117.136	50.815	-91.864	67.815	-115.864	-73.185	116.136	-123.185	119.136	51.815
-89.864	65.815	-115.864	-57.185	118.136	-108.185	123.136	66.815	-90.864	100.815
-113.864	-44.185	-110.864	-95.185	-112.864	-105.185	131.136	-113.185	121.136	63.815
129.136	97.815	132.136	111.815	129.136	121.815	134.136	130.815	-88.864	138.815
-113.864	-15.185	116.136	-70.185	126.136	97.815	-90.864	125.815	-115.864	-27.185
-112.864	-81.185	-108.864	-93.185	-109.864	-101.185	-112.864	-108.185	125.136	-112.185
122.136	64.815	130.136	101.815	130.136	117.815	-93.864	127.815	-114.864	-24.185
127.136	-79.185	125.136	91.815	-87.864	121.815	-114.864	-27.185	-110.864	-81.185
-105.864	-92.185	-109.864	-99.185	-109.864	-103.185				

Data till testexempel 3 (forts)

APPENDIX E

NÅGRA FÖRSLAG TILL ÄNDRINGAR OCH UTVECKLING AV PROGRAMMET

En del av dessa förslag, som kommit fram under arbetets gång, har redan implementerats av Ivar gustavsson.

1. En annan strategi kan tillämpas då C-polynomet är instabilt. Istället för att halvera ändringen $\Delta\theta$, ändrar man C-polynomet så att de rötter, som ligger utanför enhetscirkeln, transformeras till att ligga innanför.
2. Man bör vara misstänksam mot rötter till A-polynomet som ligger i en cirkelsektor med toppvinkel ca 60° i vänstra halvan av enhetscirkeln (det kritiska området har ett mer komplicerat utseende än denna approximation). Rötter inom detta område antyder frekvenser nästan lika höga som samplingsfrekvensen. Det vore lämpligt att testa A-polynomet i samband med att C-polynomet testas.
3. Programmet håller på att göras om till ett "kommandoprogram" istället för ett "frågeprogram".
4. Subrutiner som utför analys och syntes skall tillfogas programmet (en del är för övrigt redan inlagda). Exempelvis undersökning av residualerna, test på ordningstalet, Bodediagram etc.
5. SUBROUTINE MLPRI:s storlek minskas genom att använda SYMINV för invertering av $V_{\theta\theta}$.
6. Man bör undersöka möjligheten att öka dimensionen på parametervektorn $T(\theta)$ från 20 till t ex 25.
7. Erfarenhetsmässigt har det visat sig att uthopp vid konvergens till 90 % sker på grund av testen $q \leq 10^{-3}$. En jämnare fördelning mellan de tre olika konvergenstesterna $q \leq 10^{-3}$, $\max_i \left| \frac{\Delta\theta_i}{\theta_i} \right| \leq 10^{-4}$ och $\left| \frac{\Delta V}{V} \right| \leq 10^{-6}$ är önskvärd. Därför bör $q \leq 10^{-3}$ förslagsvis ersättas med $q \leq 10^{-4}$.