

DEAD - BEAT STYRNING AV EN  
VÄRME - DIFFUSIONSPROCESS

TORSTEN CROMNOW

RE-149 Oktober 1974  
Inst.för Reglerteknik  
Lunds Tekniska Högskola

DEAD-BEAT STYRNING AV EN VÄRME-DIFFUSIONSPROCESS

Torsten Cromnow

Handledare: B. Leden

Oktober 1974.

## ABSTRACT

The process is a pilot plant in the laboratory and consists of a long copper rod. The inputs of the process are the end temperatures of the rod and the outputs are the temperatures in seven equidistant points on the rod. The process is very accurately modelled by a one-dimensional heat equation.

Control strategies have been constructed which take the profile of the rod, from an arbitrary initial profile, to a stationary profile, i.e. the temperature of the profile varies linearly with the space-coordinate. The number of measured process outputs are varied from one experiment to another and the state of the process is reconstructed using Kalman filters, based on the available information. The reconstructed state vector is fed into the controller and a linear feedback from the reconstructed state vector is formed. The calculations of the control strategies and the Kalman filters are based on very accurately lumped models of the diffusion process.

It turns out that knowledge of the end temperatures of the rod is enough to obtain very satisfactory profile control and that additional information, obtained by measuring the temperature in some points along the rod, only increases the performance of the controller slightly. This result is encouraging because, in a practical situation, it is often possible only to measure the boundary temperatures of a heat process.

There exists a contradiction between the requirement of a fast settlement of the profile and the requirement of small control variables. However, despite the boundary temperatures are restricted to take on values which exceed the maximum value

of the terminal profile slightly, there is considerable gain in time when this terminal profile is obtained by using a minimum-time controller compared to any uncontrolled case.

#### ABSTRAKT

Processen är en försöksuppställning i institutionens laboratorium och består av en 45 cm lång kopparstav. Insignalerna till processen är ändtemperaturerna, och utsignalerna från processen är temperaturen i sju ekvidistanta punkter på staven. Processen beskrivs väl av den en-dimensionella värmeledningsekvationen.

Styrlagar har konstruerats vilka ändrar stavens profil från en godtycklig initialprofil till en stationär profil, dvs stavens temperatur varierar lineärt med lägeskoordinaten. Antalet mätbara ut signaler varierar från ett experiment till ett annat och processens tillstånd rekonstrueras mha Kalmanfilter, baserade på tillgänglig information. Den rekonstruerade tillståndsvektorn överföres till regulatorn och en lineär återkoppling från de rekonstruerade tillståndsvariablerna bildas. Beräkningarna av styrstrategierna och Kalmanfiltren baseras på mycket noggranna diskretiserade modeller av diffusionsprocessen.

Det visar sig att kännedom om stavens ändtemperaturer är tillräckligt för att erhålla mycket tillfredsställande profilreglering och att ytterligare information, erhållen genom mätning av temperaturen i några punkter längs staven, endast obetydligt ökar regulatorns prestanda. Detta resultat är uppmuntrande då det i en praktisk situation ofta endast är möjligt att mäta ändtemperaturerna i en värmeprocess.

Det existerar en motsägelse mellan kravet på snabb reglering och kravet på små styrsignaler. Även om ändtemperaturerna begränsas så att de endast antager värden vilka något överstiger det största värdet i slutprofilen, erhålles en avsevärd vinst då denna slutprofil fås via en minimaltid regulator jämfört med vilket oreglerat fall som helst.

## INNEHÅLLSFÖRTECKNING

	sid.
1. INLEDNING	1
2. VÄRME-DIFFUSIONSPROCESSEN	
2.1 Den experimentella processen.	2
2.2 Matematisk modell för processen.	3
2.3 Jämförelse mellan stegsvaren för de olika modellerna.	12
3. DEAD-BEAT STYRLAGAR	
3.1 Allmänt om dead-beat styrlagar.	14
3.2 Två teorem för beräkning av dead- beat styrlagar.	16
3.3 Dekomponering av $\Phi$ .	25
4. RESULTAT FRÅN DEAD-BEAT REGLERINGEN	
4.1 Beräkning av dead-beat styrlagar för diffusionsprocessen samt simulerade resultat.	28
4.2 Experimentella resultat under RSX.	38
5. KALMANFILTRERING	
5.1 Allmänt om Kalmanfiltrering.	45
5.2 Beräkning av Kalmanförstärkningen K.	47
5.3 Egenvärdena till matrisen $\Phi - K\theta$ .	50
6. KALMANFILTRERING OCH STOKASTISK MODELL FÖR DIFFUSIONSPROCESSEN	
6.1 Stokastisk modell för diffusionsprocessen.	51
6.2 Bestämning av de båda diskreta kovarians- matriserna R1D och R2D.	52
6.3 Experimentella resultat under RSX.	54

	sid.
7. PROGRAMBESKRIVNINGAR	
7.1 RODPAC.	81
7.2 DB.	82
7.3 KEST och SKEST.	83
7.4 Disposition av de i Kalmanfiltren ingående matrisernas filhuvuden.	84
7.5 Disposition av filhuvudena i återkopplings- matriserna.	85
7.6 Disposition av commonblocket IOCOM i de tre programpaketen DB,SKEST och KEST.	85
7.7 Illustrering av eventvariablernas funktion vid samkörning av DB och SKEST.	87
7.8 Länkstruktur för programpaketet SKEST.	88

## APPENDICES

- A Stegsvär för några olika modeller av  
diffusionsprocessen.

## PROGRAMLISTNINGAR

- B RODPAC  
C DB  
D KEST  
NOISE  
E SKEST

## 1. INLEDNING

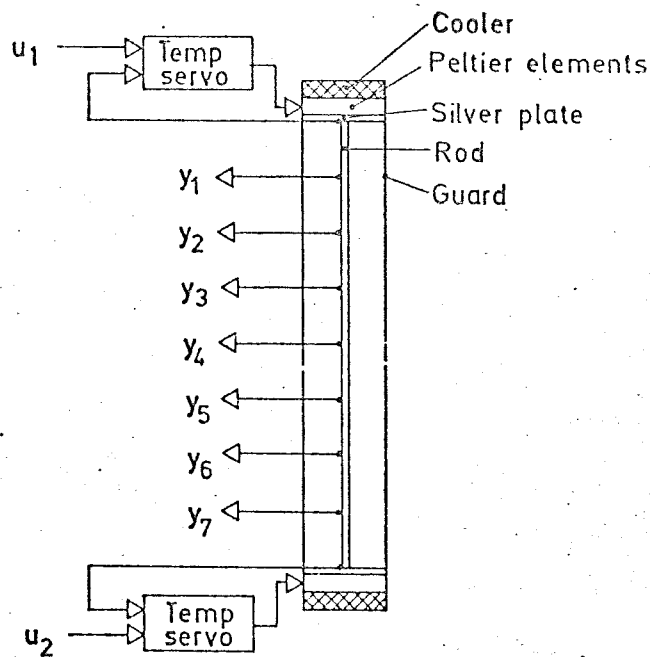
Examensarbetet syftar till att studera möjligheter att reglera profilen hos en endimensionell diffusionsprocess. Processen skall styras på en förutbestämd tid från en godtycklig initialprofil till en uppgiven slutprofil. Det antal mätsignaler från processen som antages tillgängliga för mätning varierar från ett experiment till ett annat och processens tillståndsvariabler rekonstrueras med ett Kalmanfilter baserat på tillgänglig information. Den rekonstruerade tillståndsvektorn matas till regulatorn som är en linjär återkoppling från processens tillståndsvariabler. Det visar sig att information om processens ändtemperaturer är tillräcklig för att erhålla en mycket noggrann profilreglering. Detta är ur praktisk synvinkel mycket tillfredsställande emedan man i många praktiska situationer endast kan mäta en värmeprocess ändtemperaturer.



## 2. VÄRME - DIFFUSIONSPROCESSEN

### 2.1 Den experimentella processen.

Den experimentella processen är schematiserad i Figur 2.1.



Figur 2.1 - Schematisk figur över Värme-Diffusionsprocessen.

Staven, som är tillverkad av koppar, är 1.4 cm i diameter samt 45 cm lång. Dessa dimensioner gör att staven kan betraktas som endimensionell. Avståndet mellan temperaturgivaren är  $l = 5.625$  cm

(2.1)

En cylindrisk mantel, också tillverkad av koppar, användes för att reducera värmeförlusterna till omgivningen. Staven är placerad i centrum av manteln och silverplattor med hög värmeledningsförmåga förbinder ändytorna på staven och manteln termiskt. Värmekällorna utgöres av Peltier-element.

Dessa fördelar värmet likformigt över silverplattorna. Kylarna användes för att upprätthålla en konstant temperatur på ena sidan av Peltier-elementen. Manteln omgives av en konventionell värmeisolering.

Stavens ändtemperatur styres m h a servon från insignalerna  $u_1, u_2$ . Termistortemperaturgivare avkänner temperaturen i 7 ekvidistanta punkter på staven. Termistorerna är monterade i små hål, borrade i staven, och är i god termisk kontakt med staven. Temperaturgivaren omvandlar resistansvariationen hos termistorerna till utsignaler  $y_1, y_2, \dots, y_7$ . Servona och temperaturgivarna är konstruerade så att temperaturen  $T^{\circ}\text{C}$  svarar mot en spänning  $v$  V, varvid

$$v = T - 25 \quad (2.2)$$

Processens arbetsområde är  $20^{\circ}\text{C} - 30^{\circ}\text{C}$ .

## 2.2 Matematisk modell för processen.

Vår avsikt är att härleda en modell på tillståndsform, som beskriver den experimentella processen. Staven antages vara så tunn att temperaturen i alla punkter på en tvärsnittsyta kan antagas vara konstant. Därmed specificeras temperaturen  $\theta = \theta(z, t)$  av tiden  $t$  och avståndet  $z$  längs staven. Stavens termiska egenskaper betecknas  $\kappa, \rho, c, k$ , där  $\kappa$  ( $\text{m}^2/\text{sek}$ ) är temperaturledningstalet,  $\rho$  ( $\text{kg}/\text{m}^3$ ) tätheten,  $c$  ( $\text{J}/\text{kg},^{\circ}\text{C}$ ) specifika värmekapaciteten samt  $k = \rho c \kappa$  ( $\text{W}/\text{m},^{\circ}\text{C}$ ) är värmeledningstalet.

Antager vi att staven har små radiella värmeförluster till manteln beskrives värmeledningen i staven av ekvationen

$$\frac{\partial \theta}{\partial t} + \mu \theta = \kappa \frac{\partial^2 \theta}{\partial z^2} \quad 0 < z < L \quad (2.3)$$

$$\theta(0, t) = \theta_1(t)$$

$$\theta(L, t) = \theta_2(t)$$

där  $\mu$  (1/sek) betecknar stavens ytvärmeförlustkoefficient. Denna konstant tar hänsyn till värmeförlusten p g a konvektion, ledning och strålning.

Den aktuella staven är tillverkad av kommersiell koppar. De båda konstanterna  $\kappa$  och  $\mu$  är bestämda på experimentell basis, varvid följande värden erhållits

$$\kappa = 1.159 \pm 0.001 \text{ cm}^2/\text{sek}$$

$$\mu = 0.36 \cdot 10^{-4} \pm 0.01 \cdot 10^{-4} \text{ 1/sek} \quad (2.4)$$

Modellen av diffusionsprocessen (2.3) kan representeras med ett ändligt antal ordinära differentialekvationer, om den partiella derivatan i z-led approximeras med en differensformel. Användes central approximation ger L.J. Comrie: "Chamber's shorter six-figure mathematical tables"

$$\frac{\partial^2 \theta}{\partial z^2} \Big|_{i,j} = \frac{1}{h_z^2} \left( \delta^2 - \frac{1}{12} \delta^4 + \frac{1}{90} \delta^6 - \frac{1}{560} \delta^8 + \frac{1}{3150} \delta^{10} - \frac{1}{16632} \delta^{12} + \dots \right) \theta_{i,j}$$

$$i = 1, 2, \dots, n-1 \quad (2.5)$$

där  $\theta_{i,j}$  är temperaturen i den i:te punkten vid tidpunkten j, n anger antalet indelningspunkter,  $h_z = L/n$  samt  $\delta$  är den centrala differensoperatorn. Observera att operatorn  $\delta$  endast verkar i z-led.

Olika approximationer av av andraderivatatan.

Vi har använt följande approximationer av  $\partial^2\theta/\partial z^2$ .

$$\frac{\partial^2\theta}{\partial z^2} \Big|_{i,j} = \frac{1}{h_z^2}(\delta^2)\theta_{i,j} + o(h_z^2) \quad (2.6.1)$$

$$\frac{\partial^2\theta}{\partial z^2} \Big|_{i,j} = \frac{1}{h_z^2}(\delta^2 - \frac{1}{12}\delta^4)\theta_{i,j} + o(h_z^4) \quad (2.6.2)$$

$$\frac{\partial^2\theta}{\partial z^2} \Big|_{i,j} = \frac{1}{h_z^2}(\delta^2 - \frac{1}{12}\delta^4 + \frac{1}{90}\delta^6)\theta_{i,j} + o(h_z^6) \quad (2.6.3)$$

där  $o(h^n)$  anger ordningen för felet i approximationen.

Ovanstående tre approximationer kräver i ordning kännedom av  $\theta$  i 3,5 respektive 7 ekvidistanta punkter.

Sambandet mellan förskjutningsoperatören  $E$  och den centrala differensoperatören  $\delta$  lyder

$$\delta^2 = E + E^{-1} - 2 \quad (2.7)$$

varför (2.6.1), (2.6.2) och (2.6.3) kan skrivas

$$\frac{\partial^2\theta}{\partial z^2} \Big|_{i,j} \approx \frac{1}{h_z^2}(\theta_{i-1,j} - 2\theta_{i,j} + \theta_{i+1,j}) \quad i = 1, 2, \dots, n-1 \quad (2.8.1)$$

$$\frac{\partial^2\theta}{\partial z^2} \Big|_{i,j} \approx \frac{1}{12h_z^2}(-\theta_{i-2,j} + 16\theta_{i-1,j} - 30\theta_{i,j} + 16\theta_{i+1,j} - \theta_{i+2,j}) \quad i = 2, 3, \dots, n-2 \quad (2.8.2)$$

$$\frac{\partial^2 \theta}{\partial z^2} \Big|_{i,j} \approx \frac{1}{180h_z^2} (2\theta_{i-3,j} - 27\theta_{i-2,j} + 270\theta_{i-1,j} - 490\theta_{i,j} + 270\theta_{i+1,j} - 27\theta_{i+2,j} + 2\theta_{i+3,j}) \quad i = 3, 4, \dots, n-3$$

(2.8.3)

Inför tillståndsvektorn (temperatur i  $n-1$  ekvidistanta punkter på staven)

$$x = \begin{bmatrix} \theta_{1,j} \\ \theta_{2,j} \\ \vdots \\ \theta_{n-1,j} \end{bmatrix}$$

(2.9)

och insignalvektorn (stavens ändtemperaturer)

$$u = \begin{bmatrix} \theta_{0,j} \\ \theta_{n,j} \end{bmatrix}$$

(2.10)

Vi kan nu ange olika approximationer till (2.3)

1) Baserad på (2.8.1)

Vi finner

$$\dot{x} = \begin{bmatrix} -a & b & 0 & & & & \\ b & -a & b & 0 & 0 & & \\ 0 & b & -a & b & 0 & & \\ & & & & & & 0 \\ & & & & & & b \\ & 0 & & & & & \\ & & & & & & 0 \\ & & & & & & b \\ & & & & & & 0 \\ & & & & & & b \\ & & & & & & 0 \\ & & & & & & b \\ & & & & & & 0 \\ & & & & & & b \\ & & & & & & 0 \\ & & & & & & b \\ & & & & & & 0 \\ & & & & & & b \\ & & & & & & 0 \\ & & & & & & b \end{bmatrix} x + \begin{bmatrix} b & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & b \end{bmatrix} u$$

(2.11)

med

$$\begin{aligned} a &= 2\kappa/h_z^2 + \mu \\ b &= \kappa/h_z^2 \end{aligned} \quad (2.12)$$

Denna modell benämnes i fortsättningen ROD1.

2) Baserad på (2.8.1) och (2.8.2). Approximationen enligt (2.8.1) användes för  $i = 1, n-1$  samt approximationen enligt (2.8.2) för  $i = 2, 3, \dots, n-2$ .

Vi finner

$$\dot{x} = \begin{bmatrix} -a & b & 0 & 0 & & & \\ & d-c & d & -e & 0 & & 0 \\ -e & & d & -c & d & -e & 0 \\ 0 & & & & & & 0 \\ & & & & & & -e \\ & & & & & & 0 \\ & & & 0 & -e & d & -c & d \\ & & & & & & & & & & & 0 \\ & & & & & & & & & & & 0 \\ & & & & & & & & & & & 0 \\ & & & & & & & & & & & 0 \\ & & & & & & & & & & & 0 \\ & & & & & & 0 & 0 & b & -a \end{bmatrix} x + \begin{bmatrix} b & 0 \\ -e & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & -e \\ 0 & b \end{bmatrix} u \quad (2.13)$$

med

$$\begin{aligned} a &= 2\kappa/h_z^2 + \mu \\ b &= \kappa/h_z^2 \\ c &= 30\kappa/(12h_z^2) + \mu \\ d &= 16\kappa/(12h_z^2) \\ e &= \kappa/(12h_z^2) \end{aligned} \quad (2.14)$$

Denna modell benämnes i fortsättningen ROD2.

3) I samband med Kalmanfiltrering beskriver vi även servodynamiken. Därmed införes processens ändtemperaturer som tillståndsvariabler. Detta är fysikaliskt mycket tillfredsställande ty stavens ändtemperaturer kan inte ändras oändligt snabbt, vilket (2.11) och (2.13) förutsätter.

Vi har ansatt en första ordningens servodynamik

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{1+Ts} \quad (\text{dvs } y(t)=1-e^{-t/T}) \text{ med tidskonstanten } T=1.5 \text{ sekunder.}$$

Vi erhåller då följande modell

$$\dot{x} = \begin{bmatrix} -1/T & 0 & \text{-----} & 0 \\ b & & & | \\ -e & & & | \\ 0 & & & | \\ \vdots & & & | \\ 0 & \text{-----} & & -1/T \end{bmatrix} x + \begin{bmatrix} 1/T & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ 0 & 1/T \end{bmatrix} u \quad (2.15)$$

≡ ROD2

där  $u$  är insignalen till servot samt  $b$  och  $e$  definierade i (2.14). Denna modell kallas fortsättningsvis ROD3.

4) Baserad på (2.8.1), (2.8.2) och (2.8.3). Approximationen enligt (2.8.1) användes för  $i=1, n-1$ , approximationen enligt (2.8.2) användes för  $i=2, n-2$  samt approximationen enligt (2.8.3) användes för  $i=3, 4, \dots, n-3$ .

Vi finner

$$\mathbf{x} = \begin{bmatrix} -a & b & 0 & 0 & 0 & & & & \\ d & -c & d & -e & 0 & 0 & & & 0 \\ -h & g & -f & g & -h & i & 0 & & \\ i & -h & g & -f & g & -h & i & 0 & \\ 0 & & & & & & & & 0 \\ & & & & & & & & i \\ & & & & i & -h & g & -f & g & -h \\ & & 0 & & & & & & & & 0 \\ & & & & 0 & 0 & -e & d & -c & d \\ & & & & & & & & & & b & -a \end{bmatrix} \mathbf{x} + \begin{bmatrix} b & 0 \\ -e & \\ i & \\ 0 & \\ & \\ & \\ 0 & \\ i & \\ -e & \\ 0 & b \end{bmatrix} \mathbf{u} \quad (2.16)$$

med

$$\begin{aligned} a &= 2\kappa/h_z^2 + \mu \\ b &= \kappa/h_z^2 \\ c &= 30\kappa/(12h_z^2) + \mu \\ d &= 16\kappa/(12h_z^2) \\ e &= \kappa/(12h_z^2) \\ f &= 490\kappa/(180h_z^2) + \mu \\ g &= 270\kappa/(180h_z^2) \\ h &= 27\kappa/(180h_z^2) \\ i &= 2\kappa/(180h_z^2) \end{aligned} \quad (2.17)$$

Denna modell benämnes fortsättningsvis ROD4.

Givetvis kan man fortsätta och förfina sin modell ytterligare, men arbetet att räkna ut koefficienter i en formel baserad på förskjutningsoperatoren blir mycket betungande om ännu fler termer i (2.5) medtages.



Slutligen skall nämnas att maximalt antal termer har medtagits vid approximation av  $\partial^2\theta/\partial z^2$  för varje betraktad punkt på staven inom den tillåtna noggrannhetsklassen.

För jämförelse mellan de olika modellerna finns på följande sida en tabell över egenvärdena till några olika A-matriser (där A definieras av  $\dot{x} = Ax + Bu$ ).

Som jämförelsematerial finns i tabellen även de teoretiska egenvärdena

$$\lambda_k = -(a + \pi^2 k^2)/T \quad k = 1, 2, \dots \quad (2.18)$$

där

$$a = \mu^2 l^2 / \kappa$$

$$T = l^2 / \kappa$$

namn	HRA11				HRA22				HRA42				teoretiska värden
	ROD1	ROD2	ROD4	ROD3	ROD2	ROD3	ROD4	ROD4	ROD2	ROD3	ROD4	ROD4	
modell	7				7				15				
nx	7				7				15				
E $\lambda_1$	-0.005 613	-0.005 677	-0.005 678	-0.005 677	-0.005 685	-0.005 685	-0.005 687	-0.005 687	-0.005 685	-0.005 687	-0.005 687	-0.005 687	-0.005 685
G $\lambda_2$	-0.021 493	-0.022 229	-0.022 249	-0.022 229	-0.022 613	-0.022 613	-0.022 618	-0.022 618	-0.022 613	-0.022 618	-0.022 618	-0.022 618	-0.022 631
E $\lambda_3$	-0.045 261	-0.047 569	-0.047 764	-0.047 569	-0.050 682	-0.047 569	-0.050 727	-0.050 727	-0.050 682	-0.050 727	-0.050 727	-0.050 727	-0.050 875
N $\lambda_4$	-0.073 296	-0.078 694	-0.079 760	-0.078 694	-0.089 421	-0.078 694	-0.089 618	-0.089 618	-0.089 421	-0.089 618	-0.089 618	-0.089 618	-0.090 417
V $\lambda_5$	-0.101 332	-0.114 556	-0.116 207	-0.114 556	-0.137 863	-0.114 556	-0.138 510	-0.138 510	-0.137 863	-0.138 510	-0.138 510	-0.138 510	-0.141 257
Ä $\lambda_6$	-0.125 099	-0.152 543	-0.157 156	-0.152 543	-0.194 613	-0.152 543	-0.196 412	-0.196 412	-0.194 613	-0.196 412	-0.196 412	-0.196 412	-0.203 394
R $\lambda_7$	-0.140 980	-0.183 380	-0.200 255	-0.183 380	-0.258 203	-0.183 380	-0.262 380	-0.262 380	-0.258 203	-0.262 380	-0.262 380	-0.262 380	-0.276 929
D $\lambda_8$													
E $\lambda_9$													
N $\lambda_{10}$													
$\lambda_{11}$													
$\lambda_{12}$													
$\lambda_{13}$													
$\lambda_{14}$													
$\lambda_{15}$													

Eigenvärden {  
servo

Tabell 2.1

## 2.3 Jämförelse mellan stegsvaren för de olika modellerna.

För att få tillförlitligt jämförelsematerial skrevs ett program kallat AROD som beräknar stegsvaret för staven baserat på det analytiska uttrycket

$$\theta(z,t) = \frac{\sinh \zeta(\ell-z)}{\sinh \zeta \ell} - \sum_{k=1}^{\infty} \frac{2\pi k(-1)^{k+1} \sin\left(\frac{\ell-z}{\ell} \pi k\right)}{a+\pi^2 k^2} e^{-\frac{a+\pi^2 k^2}{T} t}$$

där

$$\zeta = \sqrt{\mu/\kappa}$$

$$T = \ell^2/\kappa \quad (2.19)$$

De olika modellernas stegsvar har beräknats med hjälp av syntespaketet Synpac på så sätt att det kontinuerliga systemet samplats (med samplingsintervallen 10 respektive 60 sekunder), varefter den diskreta systemekvationen  $x(t+1) = \Phi x(t) + \Gamma u(t)$  itererats med  $x(0) = (0 \ 0 \ \dots 0)^T$  samt  $u(t) = (1 \ 0)^T$ ;  $t = 1, 2, 3, \dots$

I appendix A finns tabeller över stegsvaren i punkterna  $z = \ell/8$ ,  $z = 3\ell/8$ ,  $z = 5\ell/8$  och  $z = 7\ell/8$  för  $t = 0, 10, 20, \dots, 300$  sekunder samt  $t = 0, 60, 120, \dots, 1800$  sekunder för några olika modeller enligt nedan:

Par(A,B)	nx	nu	modell
HRA11,HRB11	7	2	ROD1
HRA21,HRB21	7	2	ROD2
HRA41,HRB41	7	2	ROD4
HRA22,HRB22	15	2	ROD2
HRA42,HRB42	15	2	ROD4

Dessutom finns i tabellerna felen  $E(i) = y_{\text{anal}}(i) - y_{\text{RODI}}(i)$  i ovan nämnda punkter.

Studerar man de simulerade stegsvaren, finner man att för att förbättra sin modell tjänar man betydligt mera på att införa fler tillstånd än på att förfina sin approximation av  $\partial^2\theta/\partial z^2$ , vilket vi även kunde konstatera vid jämförelserna av egenvärden i tabell 2.1.

### 3. DEAD-BEAT STYRLAGAR

#### 3.1 Allmänt om dead-beat styrlagar

Vi skall behandla problemet att styra ett system på sådant sätt att tillståndsvektorn på så få samplingsintervall som möjligt antager ett på förhand föreskrivet värde. Systemet antages vara linjärt diskret och tidsinvariant samt givet på formen

$$x(t+1) = \Phi x(t) + \Gamma u(t) \quad (3.1)$$

där  $x$  är en  $n$ -vektor,  $u$  är en  $m$ -vektor och  $\Phi, \Gamma$  är matriser med dimensionen  $n \times n$  respektive  $n \times m$ . Problemet är lösbart om och endast om systemet (3.1) är styrbart. Vi kommer därför att inskränka vårt studium till styrbara system. De styrstrategier som löser problemet ovan benämnes ofta i litteraturen dead-beat styrlagar.

Vi skall behandla dels tidsvariabla, dels tidsinvarianta styrlagar. De förra utgör en lösning till optimeringsproblemet.

$$V(x, t) = \min_{u(t), u(t+1), \dots, u(N-1)} x^T(N) Q_0 x(N), \quad Q_0 > 0 \quad (3.2)$$

där  $Q_0$  är en positivt definit matris samt  $x(N)$  tillståndsvektorns värde vid sluttidpunkten  $t = N$ . Beteckningen i (3.2) visar att lösningen endast beror på initialtillståndet  $x$  samt tidpunkten  $t$ . Enär det samplade systemet förutsättes styrbart existerar ett heltal  $t \leq N$  sådant att  $V(x, t) = 0$ .

Betecknar vi med  $t_{DB}$ , det största heltal med denna egenskap, finner vi att tillståndsvektorn kan styras från ett godtyckligt initialtillstånd  $x_0$  till origo på  $N-t_{DB}$  samplingsintervall. Därmed kan systemet även styras från ett godtyckligt initialtillstånd  $x_0$  till ett på förhand uppgivet värde  $x_1$  på  $N-t_{DB}$  samplingsintervall.

Optimeringsproblemet har i allmänhet ej någon entydig lösning. Flertydigheten kan emellertid elimineras genom att i varje samplingsögonblick välja den lösning  $u(t)$ , som har minimal norm. Vi erhåller därmed en unik lösning  $u_{opt}$  till optimeringsproblemet med följande ur reglerteknisk synpunkt mycket intressanta egenskaper

o Lösningen erhålles som en tillståndsåterkoppling  $u_{opt}(t) = L(t)x(t)$ ,  $t = t_0, t_0+1, \dots, N-1$

o Om  $u_{opt}(t) = L(t)x(t)$

utgör en lösning till (3.2) för  $t = t_0, t_0+1, \dots, N-1$  så är lösningen till (3.2) känd för varje  $t_1 \geq t_0$  och gives av  $u_{opt}(t) = L(t)x(t)$ ,  $t=t_1, t_1+1, \dots, N-1$

o Om  $u$  är en annan lösning till (3.2) så gäller att

$$\|u(t)\| \geq \|u_{opt}(t)\|, \quad t=t_0, t_0+1, \dots, N-1$$

En rekursiv metod för beräkning av  $L(t)$  gives i teorem 1.

De tidsinvarianta reglerstrategierna är suboptimala i den meningen att de utgör lösningar till optimeringsproblemet

$$V(x_0, t_{DB}) = \min_{u(t_{DB}), u(t_{DB}+1), \dots, u(N-1)} x^T(N)Q_0x(N), \quad Q_0 > 0 \quad (3.3)$$

där heltalet  $t_{DB}$  och matrisen  $Q_0$  har definierats ovan. Mängden av lösningar till problemet (3.3) är väsentligt större än motsvarande mängd till problemet (3.2) då  $t > t_{DB}$ .

Den teori

som kan utvecklas för de tidsvariabla styrlagarna ger oss emellertid en möjlighet till rationellt val av en tidsinvariant styrlag, som applicerad på systemet (3.1) genererar en lösning till problemet (3.3). En metod att beräkna denna styrlag gives i teorem 2.

### 3.2 Två teorem för beräkning av dead-beat styrlagar.

Studiet av dead-beat styrlagar är baserat på teorem 1 och 2. Efter genomgången av teoremen kommer de att tillämpas på två enkla exempel för att illustrera innebörden.

#### Teorem 1.

Låt systemet

$$x(t+1) = \Phi x(t) + \Gamma u(t)$$

och kriteriefunktionen

$$V(x,t) = \min_{u(t), u(t+1), \dots, u(N-1)} x^T(N) Q_0 x(N), \quad Q_0 > 0$$

vara givna. Den entydiga lösningen med minimal norm, som minimerar kriteriefunktionen gives av

$$u_{\text{opt}}(t) = L(t)x(t)$$

där

$$L(t) = -(\Gamma^T S(t+1) \Gamma)^+ \Gamma^T S(t+1) \Phi$$

och  $S(t)$  gives av den rekursiva ekvationen

$$\begin{aligned}
S(t) &= \phi^T S(t+1) \phi - \phi^T S(t+1) \Gamma (\Gamma^T S(t+1) \Gamma)^+ \Gamma^T S(t+1) \phi \\
&= (\phi + \Gamma L(t))^T S(t+1) \phi \\
&= (\phi + \Gamma L(t))^T S(t+1) (\phi + \Gamma L(t))
\end{aligned}$$

$$S(N) = Q_0$$

Det minimala värdet på kriteriefunktionen vid tidpunkten  $t = t_0$  ges av  $V(x, t_0) = x^T(t_0) S(t_0) x(t_0)$ . Dessutom, om  $u$  är en annan lösning som minimerar kriteriefunktionen så gäller  $\|u(t)\| \geq \|u_{\text{opt}}(t)\|$ ,  $t = t_0, t_0+1, \dots, N-1$ .

För att ej i onödan komplicera utsagan i teorem 2, antages systemmatriserna  $\phi, \Gamma$  i (3.1) uppfylla  $\phi$  reguljär samt  $\text{rk } \Gamma = m$ . Detta gäller för vår process och är i allmänna fallet ej någon väsentlig inskränkning. Det bör påpekas att matrisen  $\phi$  är reguljär, om det diskreta systemet har erhållits genom sampling av ett kontinuerligt system med rationell överföringsfunktion. Vidare innebär villkoret  $\text{rk } \Gamma < m$  att systemet (3.1) innehåller insignaler, som ej bidrager till systemets styrbarhet. Dessa kan därför utelämnas, varvid ett system erhålles, där kolonnerna i matrisen  $\Gamma$  är linjärt oberoende.

Med styrbarhetsindex  $v$  för systemet (3.1) avses det minsta positiva heltal för vilket matrisen

$$W_v = [\Gamma, \phi\Gamma, \dots, \phi^{v-1}\Gamma] \quad (3.4)$$

har rangen  $n$ . Då matrisen  $\phi$  är reguljär gäller



$$R[\Gamma, \phi\Gamma, \dots, \phi^{v-1}\Gamma] = R[\phi^{-1}\Gamma, \phi^{-2}\Gamma, \dots, \phi^{-v}\Gamma] = R^n \quad (3.5)$$

Därmed är det möjligt att plocka ut  $n$  linjärt oberoende vektorer ur kolonnrummet  $\{\phi^{-1}\Gamma, \phi^{-2}\Gamma, \dots, \phi^{-v}\Gamma\}$ ; vilka genererar  $R^n$ . Den väsentliga restriktionen vid valet av dessa vektorer är att ingen vektor av formen  $\phi^{-j}\gamma_k$  väljes, såvida inte alla lägre potenser av  $\phi$  gånger  $\gamma_k$  har valts.

Proceduren nedan beskriver valet av de  $n$  linjärt oberoende vektorerna i  $\{\phi^{-1}\Gamma, \phi^{-2}\Gamma, \dots, \phi^{-v}\Gamma\}$ .

- o Välj samtliga vektorer  $\phi^{-1}\Gamma$ .
- o Välj en vektor  $\phi^{-2}\gamma_k$ . Om vektorn är linjärt oberoende av vektorerna  $\phi^{-1}\Gamma$  acceptera den, i annat fall utelämna den.
- o I varje steg av proceduren välj en ny vektor  $\phi^{-j}\gamma_k$ , där alla potenser av  $\phi$  gånger  $\gamma_k$  redan tidigare har valts. Om den nya vektorn är linjärt oberoende av de tidigare valda vektorerna acceptera den, i annat fall utelämna den.
- o Proceduren avslutas då  $n$  linjärt oberoende vektorer plockats ut.

De genom proceduren erhållna vektorerna ordnas i en matris

$$\bar{W}_v = [\phi^{-1}\Gamma(1), \phi^{-2}\Gamma(2), \dots, \phi^{-v}\Gamma(v)] \quad (3.6)$$

där

$$\Gamma^{(1)} = \Gamma ; \text{rk } \Gamma^{(1)} = m$$

$$\Gamma^{(k)} = [\gamma_{\ell_1}, \gamma_{\ell_2}, \dots, \gamma_{\ell_k}] ; \text{rk } \Gamma^{(k)} = r_k , \quad k = 2, 3, \dots, v$$

(3.7)

Konstruktionen ovan implicerar

$$R(\Gamma^{(v)}) \subset R(\Gamma^{(v-1)}) \subset \dots \subset R(\Gamma^{(1)}) = R(\Gamma) \quad (3.8)$$

Vi är nu mogna för att ange teorem 2.

Teorem 2.

Givet det styrbara systemet

$$x(t+1) = \phi x(t) + \Gamma u(t)$$

där  $\phi$  är reguljär,  $\text{rk } \Gamma = m$  och  $v$  är styrbarhetsindex.

Låt  $L$  vara en  $n \times n$  matrix sådan att

$$L[\phi^{-1}\Gamma^{(1)}, \phi^{-2}\Gamma^{(2)}, \dots, \phi^{-v}\Gamma^{(v)}] = -[I_m, 0_{r_1}, \dots, 0_{r_v}]$$

där

matrisen  $[\phi^{-1}\Gamma^{(1)}, \phi^{-2}\Gamma^{(2)}, \dots, \phi^{-v}\Gamma^{(v)}]$  innehåller  $n$  linjärt oberoende kolonner sådana att

$$\Gamma^{(1)} = \Gamma ; \text{rk } \Gamma^{(1)} = m$$



Matrisen  $J$  säges vara nilpotent med index  $q = \max_{1 \leq i \leq m} (r_i)$

och matrisen innehåller  $m$  olika Jordanblock  $J_i$ . För  $J$  gäller sålunda  $J^q = 0$ .

### Exempel 1

Bestäm en tidsvariabel dead-beat strategi för den endimensionella partikeln

$$x(t+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u(t) = \Phi x(t) + \Gamma u(t)$$

### Lösning:

Enligt teorem 1 gives en tidsvariabel dead-beat strategi av

$$L(t) = -(\Gamma^T S(t+1) \Gamma)^+ \Gamma^T S(t+1) \Phi$$

där  $S(t)$  gives av rekursionsekvationen

$$S(t) = \Phi^T (S(t+1) - S(t+1) \Gamma (\Gamma^T S(t+1) \Gamma)^+ \Gamma^T S(t+1)) \Phi$$

$$S(N) = I$$

Vi har godtyckligt valt att straffa de båda tillståndsvariablerna lika, d v s  $Q_0 = I$ .

Iteration nr 1:

$$L_2 = - \begin{bmatrix} 0.4 & 1.2 \\ & \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 0.8 & 0.4 \\ 0.4 & 0.2 \end{bmatrix}$$

$$\Phi + \Gamma L_2 = \begin{vmatrix} 0.8 & 0.4 \\ -0.4 & -0.2 \end{vmatrix}$$

$$\det(\lambda I - (\Phi + \Gamma L_2)) = \lambda(\lambda - 0.6)$$

Härav matrisen  $\Phi + \Gamma L_2$  har egenvärden  $\lambda_1 = 0$ ,  $\lambda_2 = 0.6$

Iteration nr 2:

$$L_1 = -[1 \quad 1.5]$$

$$S_1 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\Phi + \Gamma L_1 = \begin{bmatrix} 0.5 & 0.25 \\ -1 & -0.5 \end{bmatrix}$$

$$\det(\lambda I - (\Phi + \Gamma L_1)) = \lambda^2$$

Härav matrisen  $\Phi + \Gamma L_1$  har samtliga egenvärden  $\lambda = 0$ .

Då  $S_1 = 0$  är systemet styrbart och styrbarhetsindex  $v = 2$ .

Exempel 2.

Givet det linjära tidsinvarianta systemet

$$x(t+1) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.75 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} x(t) + \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix} u(t) \stackrel{\Delta}{=} \Phi x(t) + \Gamma u(t)$$

Sök en tidsinvariant dead-beat återkoppling L.

Lösning:

$$R^n = R[\Phi^{-1}\Gamma, \Phi^{-2}\gamma_1], \quad v = 2$$

$$\Phi^{-1}\Gamma = \begin{bmatrix} 1 & 1 \\ 0 & 4/3 \\ 2 & 2 \end{bmatrix}, \quad \Phi^{-2}\gamma_1 = \begin{bmatrix} 1 \\ 0 \\ 4 \end{bmatrix}$$

Enligt teorem 2 gäller:

$$L[\Phi^{-1}\Gamma, \Phi^{-2}\gamma_1] = -[I_2, 0_1] \quad L \begin{bmatrix} 1 & 1 & 1 \\ 0 & 4/3 & 0 \\ 2 & 2 & 4 \end{bmatrix} = - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$L = - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 4/3 & 0 \\ 2 & 2 & 4 \end{bmatrix}^{-1} = - \begin{bmatrix} 2 & -0.75 & -0.5 \\ 0 & 0.75 & 0 \end{bmatrix}$$

Bilda

$$\Phi + \Gamma L = \begin{bmatrix} -1 & 0 & 0.5 \\ 0 & 0 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

$$\det(\lambda I - (\Phi + \Gamma L)) = \lambda^3$$

Härav matrisen  $\Phi + \Gamma L$  har samtliga egenvärden  $\lambda = 0$ .

### Kommentar

Enligt kommentar till teorem 2 finner vi

$$(\Phi^{-1}\gamma_1, \Phi^{-2}\gamma_1, \Phi^{-1}\gamma_2)^{-1}(\Phi + \Gamma L)(\Phi^{-1}\gamma_1, \Phi^{-2}\gamma_1, \Phi^{-1}\gamma_2) =$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} J_1 & \\ & J_2 \end{bmatrix} = J$$

3.3 Dekomponering av  $\Phi$ .

Beräkningen av den tidsinvarianta dead-beat styrlagen  $L$  i teorem 2 kan göras mer numeriskt välkonditionerad genom att dekomponera  $\Phi$  enligt följande:

$$\Phi = \begin{bmatrix} \phi_1 & \\ & \phi_0 \end{bmatrix} \quad (3.9)$$

Dekomponeringen svarar mot de båda generaliserade egenrummen

$$\begin{aligned} V_1 &= \{x \mid (\Phi - \lambda_k I)^{n_k} x = 0 \text{ och } |\lambda_k| \geq \epsilon, k = 1, 2, \dots, \ell\} \\ V_0 &= \{x \mid (\Phi - \lambda_k I)^{n_k} x = 0 \text{ och } |\lambda_k| < \epsilon, k = \ell+1, \ell+2, \dots, r\} \end{aligned} \quad (3.10)$$

där  $\lambda_k$  betecknar ett egenvärde till matrisen  $\Phi$  av index  $n_k$ . Antalet olika egenvärden antages vara  $r$ . Då  $\epsilon=0$  och  $n_i=1$ ;  $i = \ell+1, \ell+2, \dots, r$  gäller  $V_0 = \ker \Phi$ .

Dekomponeringen (3.9) innebär

$$R^n = V_1 \oplus V_0$$

Låt  $V_0$  och  $V_1$  vara baser för  $V_0$  respektive  $V_1$  och inför transformationen  $z = T^{-1}x$  där  $T = [V_1, V_0]$

Vi finner

$$\begin{aligned} z(t+1) &= T^{-1} \Phi T z(t) + T^{-1} \Gamma u(t) = \\ &= T^{-1} \begin{bmatrix} \phi V_1 & \phi V_0 \end{bmatrix} z(t) + T^{-1} \Gamma u(t) = \\ &= T^{-1} \begin{bmatrix} V_1 & V_0 \end{bmatrix} \begin{bmatrix} A_{11} & \\ & A_{22} \end{bmatrix} + T^{-1} \Gamma u(t) = \end{aligned}$$



$$= \begin{bmatrix} A_{11} & \\ & A_{22} \end{bmatrix} z(t) + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t)$$

där

$$A_{11} = \begin{bmatrix} J_1 & & & \\ & J_2 & & \\ & & \ddots & \\ & & & J_\ell \end{bmatrix} \quad ; \quad A_{22} = \begin{bmatrix} J_{\ell+1} & & & \\ & J_{\ell+2} & & \\ & & \ddots & \\ & & & J_r \end{bmatrix}$$

Jordanblocken  $J_1, J_2, \dots, J_\ell$  innehåller diagonalelement  $\lambda_i$  sådana att  $|\lambda_i| \geq \varepsilon$  medan blocken  $J_{\ell+1}, J_{\ell+2}, \dots, J_r$  innehåller diagonalelement  $\lambda_i$  sådana att  $|\lambda_i| < \varepsilon$ . Av (3.10) följer att matrisen  $A_{11}$  är reguljär samt med lämpligt val av  $\varepsilon$  även numeriskt välkonditionerad.

Vi låter nu  $L_1$  vara dead-beat styrlagen för paret  $(A_{11}, B_1)$  och inför återkopplingen

$$u(t) = \begin{bmatrix} L_1, 0 \end{bmatrix} z(t) \quad (3.11)$$

där  $L_1$  beräknas enligt teorem 2.

Dead-beat styrlagen i det ursprungliga koordinatsystemet lyder

$$L = \begin{bmatrix} L_1, 0 \end{bmatrix} T^{-1} = \begin{bmatrix} L_1, 0 \end{bmatrix} \begin{bmatrix} V_1, V_0 \end{bmatrix}^{-1} \quad (3.12)$$

vilket utgör en approximativ dead-beat styrlag för paret  $(\Phi, \Gamma)$ .

Kommentar

Enligt kommentar till teorem 2 kan  $\Phi+PL$  transformeras till en Jordanblockmatris. Speciellt i ett fall då vi har 2 insignaler och 7 tillstånd får vi

$$J = \begin{bmatrix} 0 & 1 & 0 & 0 & & & \\ 0 & 0 & 1 & 0 & & & \\ 0 & 0 & 0 & 1 & & & \\ 0 & 0 & 0 & 0 & & & \\ - & - & - & - & \bar{0} & \bar{1} & \bar{0} \\ & & & & 0 & 0 & 1 \\ & & & & 0 & 0 & 0 \end{bmatrix}, \text{ då vi använder teorem 2.}$$

För att erhålla en reell vinst bör vi välja  $\epsilon$  så att  $\dim V_0 = 1, 3$  eller  $5$  (annars blir Jordanblocken olika stora). En sådan dekomponering av  $\Phi$  ger upphov till symmetriska L-matriser.

## 4. RESULTAT FRÅN DEAD-BEAT REGLERINGEN

## 4.1 Beräkning av dead-beat styrlagar för diffusionsprocessen samt simulerade resultat.

Vi kommer att studera följande dead-beat styrlagar:

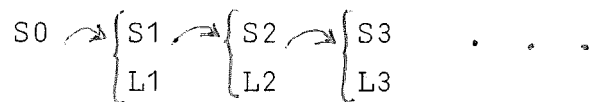
namn	typ	nx	samlingsintervall (sek)	namn utvidgad
HRF21	tidsvariabel (teorem 2)	7	60	HRF31
HRF26	"	7	180	HRF36
HRF28	"	15	60	HRF38
HRF22	tidsinvariant "dekomponering" $\dim V_0 = 3$	7	60	HRF32

Dessutom gäller i samtliga här redovisade fall att styrlagarna är baserade på den matematiska modellen ROD2. HRF.. betecknar i det tidsvariabla fallet matrisen  $(L(1), L(2), \dots, L(N-1))^T$  med beteckningar enligt teorem 2 avsnitt 3, medan HRF.. i det tidsinvarianta fallet är lika med L. Med en utvidgad matris förstås

$$\begin{bmatrix} 0 & & 0 \\ 0 & \boxed{\text{HRF..}} & 0 \\ \vdots & & \vdots \\ 0 & & 0 \end{bmatrix}$$

vilken är avsedd att användas tillsammans med en estimator (för resultat se avsnitt 6). Observera att styrlagen HRF28 kräver en estimator.

Vid presentation av simulerade resultat i det tidsvariabla fallet har vi vänt på ordningsnumren för L och S matriserna. Detta betyder att startmatrisen till Riccati-ekvationen benämnes  $S_0$ , denna matris genererar  $L_1$  och  $S_1$ ,  $S_1$  genererar  $L_2$  och  $S_2$  osv enligt figur



I samtliga tre redovisade tidsvariabla fall gäller dessutom att  $S_0$  är en enhetsmatris av lämplig dimension.

a) HRF21 (Tsamp = 60 sek)

Med användande av teorem 2 får vi (se sid 30,31)  $S_1, L_1, S_2, L_2, S_3, L_3$ . Vi ser att  $S_3$  är markant mindre än  $S_2$  samt att största diagonalelement i  $S_3$  är  $\approx 10^{-11}$  varför vi slutar beräkningen. Vi får  $\text{HRF21} = (L_3, L_2, L_1)^T$ .

Simulering:

Vi applicerar HRF21 på vår diskreta modell (3.1) med  $x(t_0) = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)^T$  och får (med  $t_0 = 0$ )

$X = (x(0), x(1), x(2), x(3))$  samt  $U = (u(0), u(1), u(2))$ , där  $X$  och  $U$  finns utskrivna på sid 31. Vi ser att som väntat  $x(3) \approx (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)^T$ .

S1

0.128238E-01    0.203291E-01    0.274403E-01    0.277151E-01    0.222035E-01    0.136151E-01    0.720204E-02  
 0.203291E-01    0.329892E-01    0.457209E-01    0.476453E-01    0.394546E-01    0.249616E-01    0.136151E-01  
 0.274403E-01    0.457209E-01    0.652552E-01    0.703609E-01    0.603609E-01    0.394546E-01    0.222035E-01  
 0.277151E-01    0.476453E-01    0.703609E-01    0.788635E-01    0.703609E-01    0.476453E-01    0.277151E-01  
 0.222035E-01    0.394546E-01    0.603609E-01    0.703609E-01    0.652552E-01    0.457209E-01    0.274403E-01  
 0.136151E-01    0.249616E-01    0.394546E-01    0.476453E-01    0.457209E-01    0.329892E-01    0.203291E-01  
 0.720204E-02    0.136151E-01    0.222035E-01    0.277151E-01    0.274403E-01    0.203291E-01    0.128238E-01

S2

0.113201E-04    0.198254E-04    0.296350E-04    0.333532E-04    0.295796E-04    0.197548E-04    0.112615E-04  
 0.198254E-04    0.347300E-04    0.519281E-04    0.584603E-04    0.518612E-04    0.346448E-04    0.197548E-04  
 0.296350E-04    0.519281E-04    0.776640E-04    0.874605E-04    0.776115E-04    0.518612E-04    0.295796E-04  
 0.333532E-04    0.584603E-04    0.874605E-04    0.985260E-04    0.874605E-04    0.584603E-04    0.333532E-04  
 0.295796E-04    0.518612E-04    0.776115E-04    0.874605E-04    0.776640E-04    0.519281E-04    0.296350E-04  
 0.197548E-04    0.346448E-04    0.518612E-04    0.584603E-04    0.519281E-04    0.347300E-04    0.198254E-04  
 0.112615E-04    0.197548E-04    0.295796E-04    0.333532E-04    0.296350E-04    0.198254E-04    0.113201E-04

S3

0.697599E-12    0.122332E-11    0.163112E-11    0.206403E-11    0.183344E-11    0.122634E-11    0.700180E-12  
 0.122380E-11    0.214582E-11    0.321164E-11    0.361993E-11    0.321545E-11    0.215079E-11    0.122806E-11  
 0.183231E-11    0.321246E-11    0.480771E-11    0.541868E-11    0.481323E-11    0.321965E-11    0.183847E-11  
 0.206520E-11    0.362050E-11    0.541815E-11    0.610664E-11    0.542452E-11    0.362879E-11    0.207228E-11  
 0.183372E-11    0.321457E-11    0.481062E-11    0.542206E-11    0.481675E-11    0.322254E-11    0.184050E-11  
 0.122582E-11    0.214885E-11    0.321578E-11    0.362468E-11    0.322029E-11    0.215468E-11    0.123076E-11  
 0.699267E-12    0.122581E-11    0.183450E-11    0.206790E-11    0.183738E-11    0.122953E-11    0.702408E-12

L1

-0.216124  
-0.263994E-01  
-0.264749  
-0.618784E-01  
-0.274409  
-0.127741  
-0.208980  
-0.208980  
-0.127741  
-0.127741  
-0.618784E-01  
-0.263994E-01  
-0.264749  
-0.264749

L2

-0.354382  
-0.108449  
-0.488244  
-0.217929  
-0.581453  
-0.382006  
-0.523726  
-0.523725  
-0.382006  
-0.581453  
-0.217929  
-0.488244  
-0.108449  
-0.354382

L3

-0.357715  
-0.111003  
-0.493873  
-0.222618  
-0.589542  
-0.389353  
-0.532425  
-0.532415  
-0.389363  
-0.589533  
-0.222625  
-0.493866  
-0.111008  
-0.357710

X

1.00000  
1.00000  
1.00000  
1.00000  
1.00000  
1.00000  
1.00000  
-1.32999  
-0.268675  
0.365394  
0.570465  
0.365424  
-0.268600  
-1.32985  
0.212137E-01  
-0.24541E-01  
-0.202261E-02  
0.183445E-01  
-0.201210E-02  
-0.245341E-01  
0.211956E-01  
-0.572191E-06  
0.269890E-05  
-0.573247E-05  
0.718930E-05  
-0.568340E-05  
0.265990E-05  
-0.561853E-06

U

-2.69662  
-2.69640  
0.154440  
0.154379  
-0.150569E-03  
-0.150092E-03

b) HRF26 (Tsamp = 180 sek)

Vi ser här enligt utskrift på sid 33 att redan i S2 är största diagonalelementet  $\approx 10^{-10}$  varför vi slutar beräkningen efter två steg (samplingsintervallet är 3 ggr så långt som motsvarande intervall i a)) och vi får  $\text{HRF26} = (\text{L2}, \text{L1})^T$ .

Simulering:

Vi applicerar HRF26 på vår diskreta modell (3.1) med  $x(t_0) = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)^T = \text{X0C}$  och får (med  $t_0 = 0$ )  $X1 = x(1), X2 = x(2), U1 = u(0)$  samt  $U2 = u(1)$ : enligt nedan.

X0C

```
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
1.000000
```

X2

```
0.814465E-05
-0.186946E-04
0.173622E-05
0.181553E-04
0.173611E-05
-0.186947E-04
0.814474E-05
```

X1

```
-0.289839
-0.564912E-01
0.991834E-01
0.153805
0.991834E-01
-0.564911E-01
-0.289839
```

U1

```
-0.566122
-0.566122
```

U2

```
0.952664E-04
0.952661E-04
```

S1

0.138646E-02	0.209629E-02	0.283813E-02	0.307011E-02	0.282209E-02	0.207450E-02	0.136658E-02
0.209629E-02	0.317127E-02	0.429635E-02	0.465115E-02	0.427876E-02	0.314739E-02	0.207450E-02
0.283813E-02	0.429635E-02	0.582516E-02	0.631212E-02	0.581221E-02	0.427876E-02	0.282209E-02
0.307011E-02	0.465115E-02	0.631212E-02	0.684740E-02	0.631212E-02	0.465115E-02	0.307011E-02
0.282209E-02	0.427876E-02	0.581221E-02	0.631212E-02	0.582516E-02	0.429635E-02	0.283813E-02
0.207450E-02	0.314739E-02	0.427876E-02	0.465115E-02	0.429635E-02	0.317127E-02	0.209629E-02
0.136658E-02	0.207450E-02	0.282209E-02	0.307011E-02	0.283813E-02	0.209629E-02	0.138646E-02

S2

0.894522E-11	0.135539E-10	0.183960E-10	0.199562E-10	0.183942E-10	0.135514E-10	0.894290E-11
0.135540E-10	0.205372E-10	0.278741E-10	0.302381E-10	0.278713E-10	0.205334E-10	0.135505E-10
0.183960E-10	0.278739E-10	0.378317E-10	0.410404E-10	0.378279E-10	0.278686E-10	0.183912E-10
0.199560E-10	0.302376E-10	0.410399E-10	0.445206E-10	0.410357E-10	0.302319E-10	0.199508E-10
0.183941E-10	0.278710E-10	0.378277E-10	0.410360E-10	0.378238E-10	0.278656E-10	0.183892E-10
0.135513E-10	0.205331E-10	0.278684E-10	0.302320E-10	0.278656E-10	0.205292E-10	0.135477E-10
0.894285E-11	0.135503E-10	0.183911E-10	0.199509E-10	0.183892E-10	0.135477E-10	0.894047E-11

L1

-0.454905E-01	-0.665356E-01	-0.864988E-01	-0.890482E-01	-0.777743E-01	-0.546867E-01	-0.346797E-01
-0.346797E-01	-0.546867E-01	-0.777743E-01	-0.890482E-01	-0.864988E-01	-0.665356E-01	-0.454905E-01

L2

-0.561038E-01	-0.822729E-01	-0.107301	-0.110899	-0.972565E-01	-0.686319E-01	-0.436581E-01
-0.436581E-01	-0.686319E-01	-0.972565E-01	-0.110899	-0.107301	-0.822729E-01	-0.561038E-01



c) HRF28 (Tsamp = 60 sek)

Eftersom S-matriserna här är av dimension 15x15 listar vi dem inte utan nöjer oss med att konstatera att största diagonalelementet i S3 är  $\approx 10^{-11}$  och vi får  $\text{HRF28} = (\text{L3}, \text{L2}, \text{L1})^T$  där L-matriserna återfinnes på sid 35 .

Simulering:

Vi applicerar HRF28 på vår diskreta modell (3.1) med  $x(t_0) = (1 \ 1 \ 1 \ \dots \ 1)^T$  och får (med  $t_0 = 0$ )

XX = (x(0),x(1),x(2),x(3)) samt UU = (u(0),u(1),u(2)) enligt nedan.

XX

1.000000	-1.96277	0.639847E-01	-0.333296E-05
1.000000	-1.31417	0.148512E-01	0.461654E-05
1.000000	-0.737893	-0.153333E-01	0.332573E-05
1.000000	-0.256985	-0.246898E-01	-0.221832E-05
1.000000	0.116441	-0.173249E-01	-0.518555E-05
1.000000	0.379479	-0.161741E-02	-0.259045E-05
1.000000	0.534701	0.128433E-01	0.273876E-05
1.000000	0.585884	0.185964E-01	0.539570E-05
1.000000	0.534699	0.128422E-01	0.272937E-05
1.000000	0.379474	-0.161929E-02	-0.260380E-05
1.000000	0.116432	-0.173270E-01	-0.519536E-05
1.000000	-0.256998	-0.246912E-01	-0.221972E-05
1.000000	-0.737911	-0.153330E-01	0.333170E-05
1.000000	-1.31419	0.148541E-01	0.462299E-05
1.000000	-1.96280	0.639909E-01	-0.333522E-05

UU

-2.65130	0.124693	-0.194703E-04
-2.65134	0.124703	-0.194881E-04

Om vi jämför med HRF21 finner vi att styrsignalerna här är något mindre än motsvarande signaler under a).

## L1

-0.421399E-01	-0.649594E-01	-0.910233E-01	-0.105112	-0.109092	-0.104432	-0.934481E-01
-0.787955E-01	-0.629319E-01	-0.477586E-01	-0.344801E-01	-0.236353E-01	-0.152142E-01	-0.861547E-02
-0.466060E-02						
-0.466060E-02	-0.861547E-02	-0.152142E-01	-0.236353E-01	-0.344801E-01	-0.477586E-01	-0.629319E-01
-0.787955E-01	-0.934481E-01	-0.104432	-0.109092	-0.105112	-0.910233E-01	-0.649594E-01
-0.421399E-01						

## L2

-0.680106E-01	-0.139569	-0.203214	-0.247320	-0.273885	-0.282694	-0.275022
-0.253629	-0.222310	-0.185303	-0.146653	-0.109636	-0.762608E-01	-0.459447E-01
-0.261103E-01						
-0.261103E-01	-0.459447E-01	-0.762607E-01	-0.109636	-0.146653	-0.185303	-0.222310
-0.253629	-0.275022	-0.282694	-0.273885	-0.247320	-0.203214	-0.139569
-0.680106E-01						

## L3

-0.896345E-01	-0.142296	-0.207410	-0.252845	-0.280562	-0.290236	-0.283041
-0.261677	-0.229940	-0.192134	-0.152416	-0.114192	-0.795812E-01	-0.480173E-01
-0.273206E-01						
-0.273215E-01	-0.480187E-01	-0.795833E-01	-0.114195	-0.152419	-0.192138	-0.229944
-0.261681	-0.283045	-0.290240	-0.280565	-0.252848	-0.207413	-0.142297
-0.896353E-01						

d) HRF22 (Tsamp = 60 sek)

Detta är en tidsinvariant styrlag beräknad med hjälp av först dekomponering av  $\Phi$  därefter med användande av teorem 1.

Egenvärdena till  $\Phi$  är:

EVAL

0.263492	0.000000
0.711317	0.000000
0.890062E-02	0.000000
0.576040E-01	0.000000
0.103496E-02	0.000000
0.166351E-04	0.000000
0.105942E-03	0.000000

där vi ansatt  $\varepsilon \approx 0.5 \cdot 10^{-2}$ . Efter diverse mellanräkningar får vi L:

L

-0.357398	-0.493344	-0.588792	-0.531629
-0.110783	-0.222208	-0.388699	-0.531629

Simulering:

Precis som tidigare applicerar vi styrlagen  $u = Lx$  med  $L = \text{HRF22}$  på vår diskreta modell (3.1). Vi låter vidare  $x(t_0) = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1)^T = X0C$ . Vi får (med  $t_0 = 0$ )  $X1 = x(1), X2 = x(2), X3 = x(3), U1 = u(0), U2 = u(1)$  samt  $U3 = u(2)$ .

X0C

1.00000  
1.00000  
1.00000  
1.00000  
1.00000  
1.00000  
1.00000

U1

-2.69285  
-2.69285

U2

0.151595  
0.151595

X1

-1.32762  
-0.267383  
0.366035  
0.570888  
0.366036  
-0.267383  
-1.32762

U3

0.186853E-07  
0.180057E-07

X2

0.198716E-01  
-0.247777E-01  
-0.165373E-02  
0.188904E-01  
-0.165373E-02  
-0.247777E-01  
0.198716E-01

X3

0.192789E-04  
-0.217418E-04  
-0.735297E-05  
0.258069E-04  
-0.735267E-05  
-0.217422E-04  
0.192788E-04

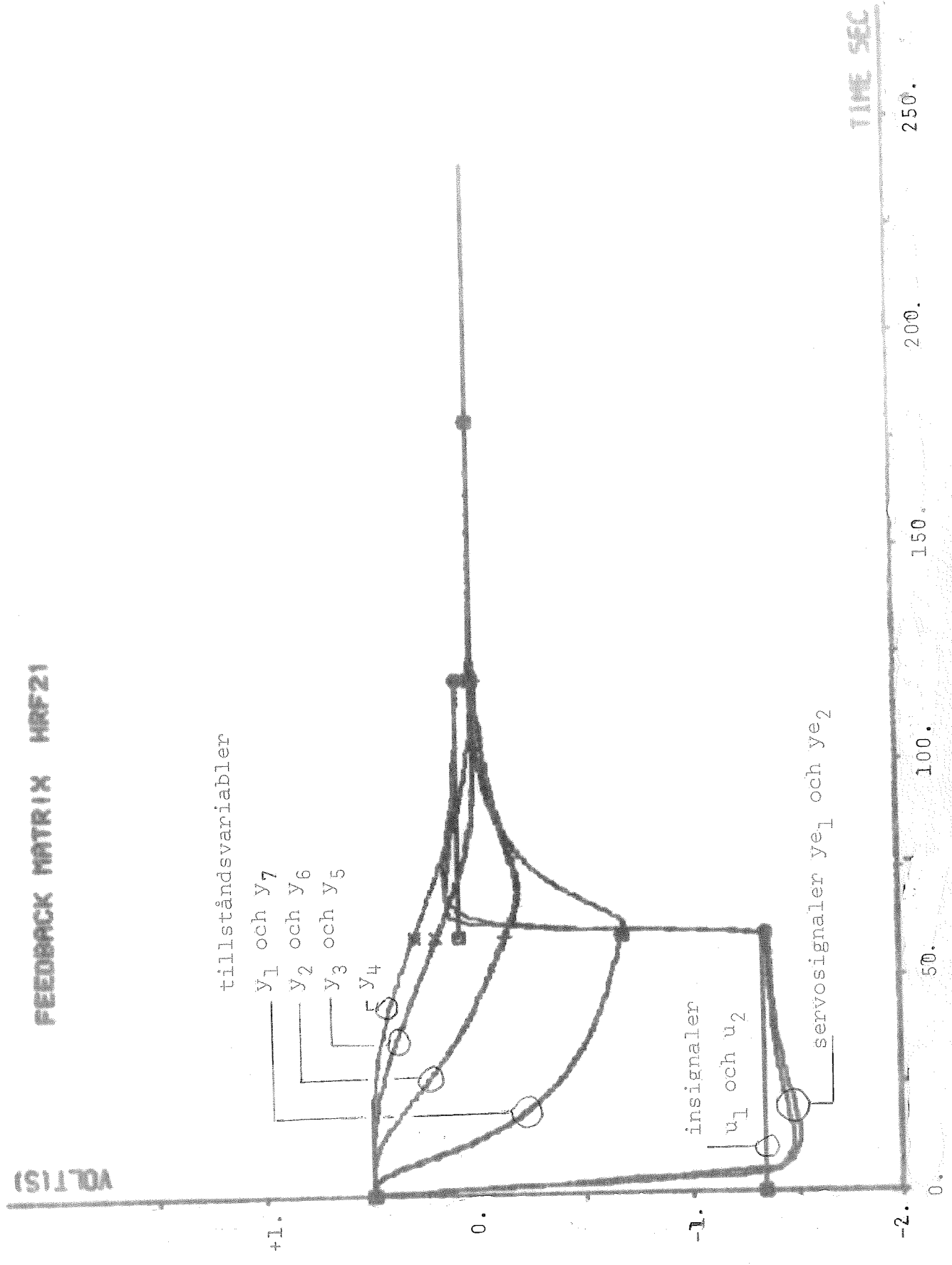
#### 4.2 Experimentella resultat under RSX.

De här redovisade styrlagarna är HRF21, HRF26 samt HRF22. I experimenten där vi studerat styrlagarna HRF21 och HRF22 har vi låtit staven antaga den stationära profilen av 25.50 °C, dvs  $x(t_0) = (0.50 \ 0.50 \ \dots \ 0.50)^T$  medan vi låtit staven antaga temperaturen 26.00 °C då vi studerat styrlagen HRF26. Vi styr därefter i samtliga fall ner stavens profil till 25.00 °C, dvs  $x(t_1) = (0 \ 0 \ \dots \ 0)^T$ .

Varje figur åtföljes av en utskrift som anger värdena på tillståndsvektorn  $YI$ , insignalerna  $UI$  samt servosignalerna  $YEI$  vid samplingstidpunkterna.

PROFILE OF DIFFUSION PROCESS 4 OCT 1974 (10:55:38)

FEEDBACK MATRIX HRF21



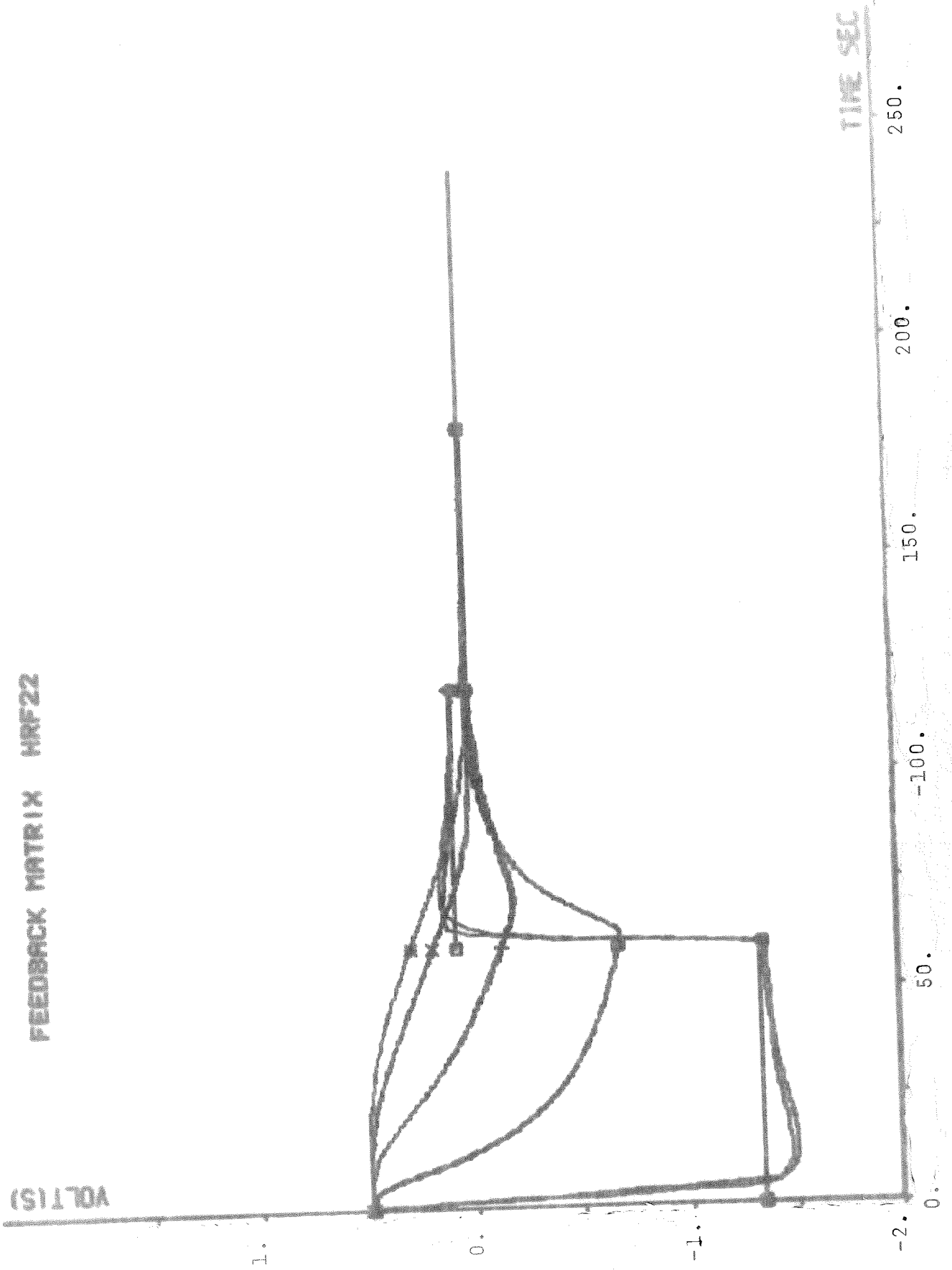
PROFILE OF DIFFUSION PROCESS  
 \*\*\*\*\*

INITIAL PROFILE U1= 0.506 U2= 0.505

TIME	U1	U2	Y1	Y2	Y3	Y4	Y5	Y6	Y7	YE1	YE2
0.000	-1.338	-1.341	0.483	0.498	0.498	0.498	0.498	0.498	0.498	0.498	0.498
60.000	0.079	0.082	-0.688	-0.132	0.190	0.293	0.186	-0.142	-0.688	-1.357	-1.362
120.000	-0.001	0.000	0.010	0.000	0.000	0.000	-0.005	-0.010	0.015	0.078	0.073
180.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
240.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

PROFILE OF DIFFUSION PROCESS 4 OCT 1979 (19:51:17)

FEEDBACK MATRIX WRF22





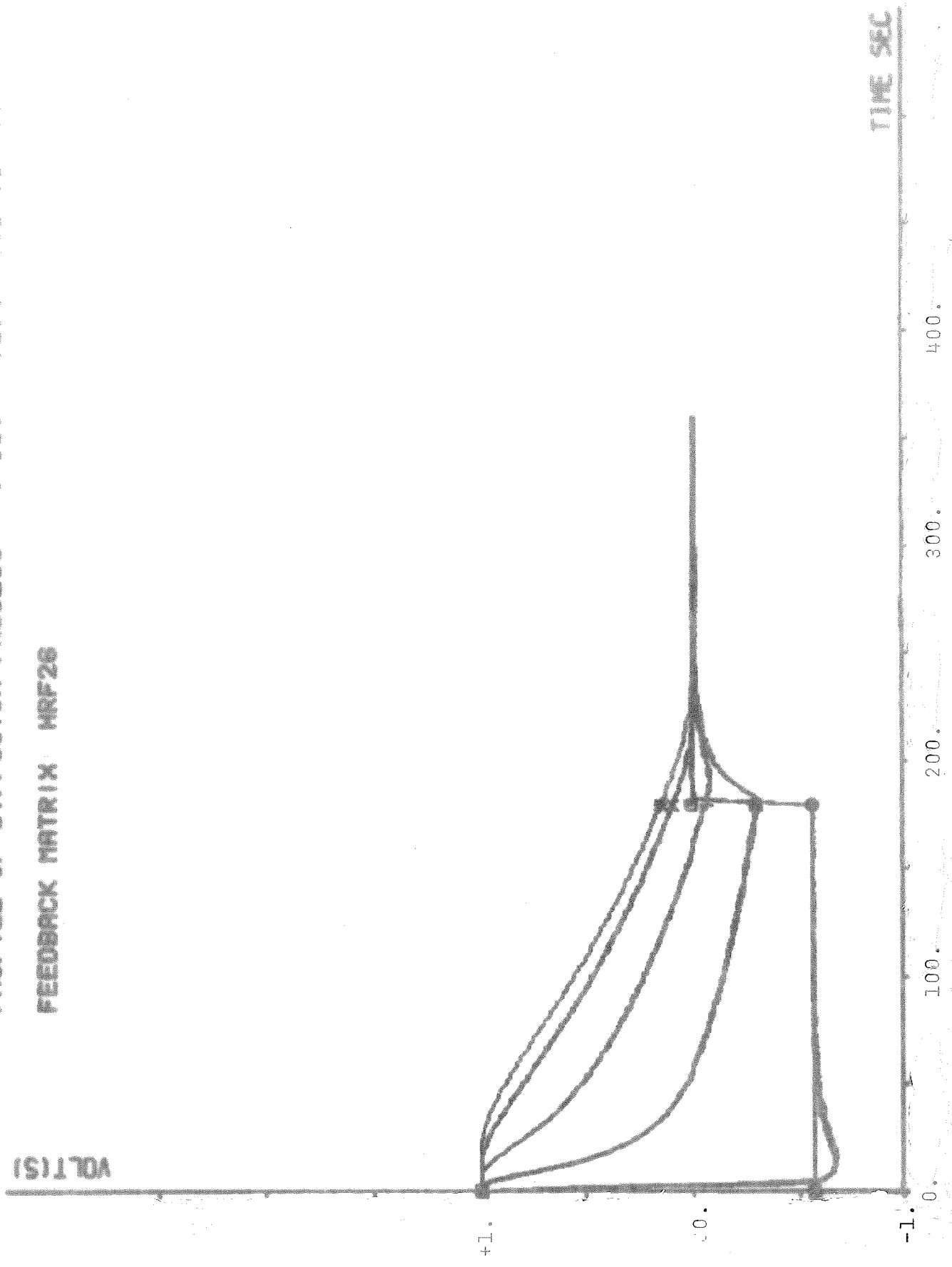
PROFILE OF DIFFUSION PROCESS  
 \*\*\*\*\*

INITIAL PROFILE U1= 0.500 U2= 0.500

TIME	U1	U2	Y1	Y2	Y3	Y4	Y5	Y6	Y7	YE1	YE2
0.000	-1.332	-1.336	0.463	0.498	0.493	0.498	0.498	0.498	0.493	0.488	0.488
60.000	0.078	0.079	-0.688	-0.132	0.190	0.293	0.186	-0.142	-0.679	-1.348	-1.357
120.000	-0.002	-0.003	0.005	0.000	0.000	0.000	0.000	-0.005	0.015	0.073	0.078
180.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
240.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

PROFILE OF DIFFUSION PROCESS 4 OCT 1974 (19:10: 7)

FEEDBACK MATRIX HRF26



PROFILE OF DIFFUSION PROCESS  
 \*\*\*\*\*

INITIAL PROFILE U1= 1.001 U2= 1.001

TIME	U1	U2	Y1	Y2	Y3	Y4	Y5	Y6	Y7	YE1	YE2
0.000	-0.563	-0.563	0.981	0.996	0.996	0.996	0.991	0.996	0.996	0.996	0.996
180.000	0.003	0.003	-0.298	-0.059	0.093	0.146	0.088	-0.063	-0.293	-0.566	-0.566
360.000	0.000	0.000	-0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

## 5. KALMANFILTRERING

### 5.1 Allmänt om Kalmanfiltrering.

Systemekvationerna gives av

$$x(t+1) = \Phi x(t) + \Gamma u(t) + v(t)$$

$$y(t) = \theta x(t) + e(t) \quad (5.1)$$

där

processbruset  $\{v(t)\}$  är en sekvens av oberoende likafördelade gaussiska stokastiska variabler med medelvärde  $E\{v(t)\} = 0 \quad \forall t$  och kovariansmatris  $R_1 = \text{cov}\{v(t), v(t)\} \quad \forall t$  samt

mätbruset  $\{e(t)\}$  är en sekvens av oberoende likafördelade gaussiska stokastiska variabler med medelvärde  $E\{e(t)\} = 0 \quad \forall t$  och kovariansmatris  $R_2 = \text{cov}\{e(t), e(t)\} \quad \forall t$ .

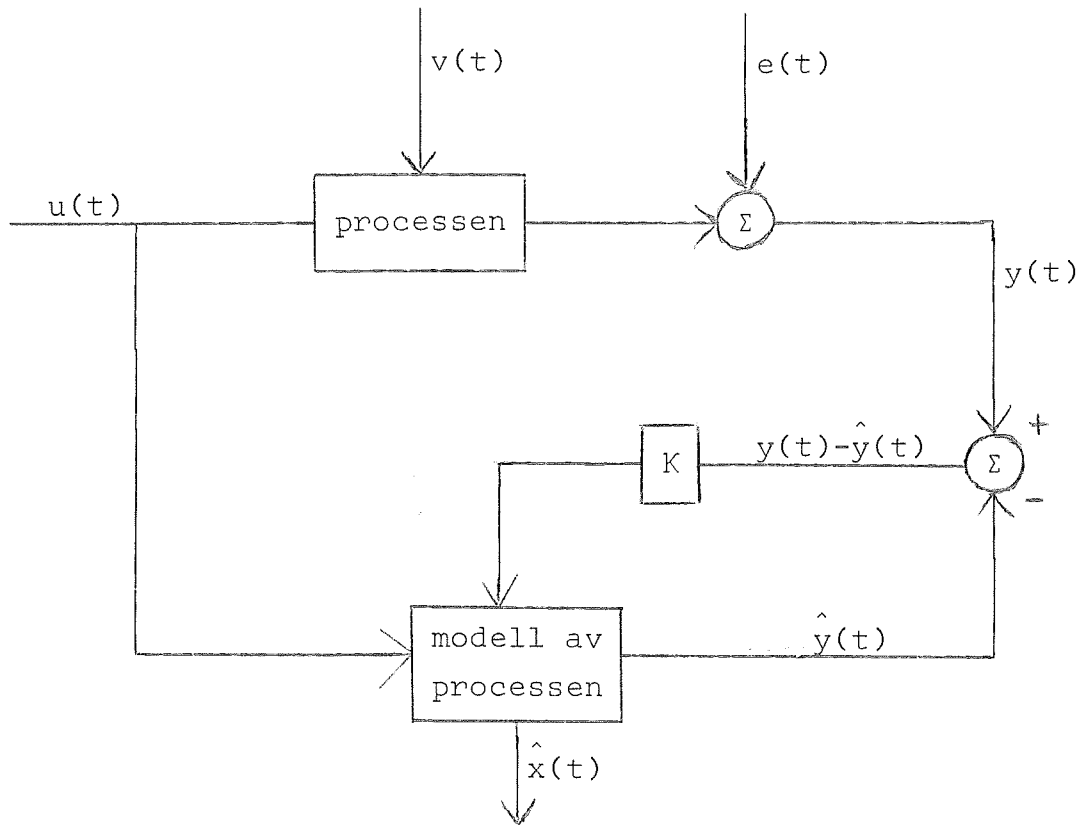
Processerna  $\{v(t)\}$  och  $\{e(t)\}$  antages sinsemellan oberoende.

Om tillståndsvektorn  $x(t)$  (eller delar av den) ej är mätbar kan den rekonstrueras med följande estimator:

$$\hat{x}(t+1) = \Phi \hat{x}(t) + \Gamma u(t) + K(t)(y(t) - \theta \hat{x}(t)) \quad (5.2)$$

där  $\hat{x}(t)$  är det rekonstruerade (estimerade) värdet av tillståndsvektorn  $x(t)$  vid tidpunkt  $t$  och  $\epsilon(t) = y(t) - \theta \hat{x}(t)$  är residualerna.

När vi sedermera önskar styra processen återkopplar vi från de estimerade tillståndsvariablerna  $\hat{x}$ .



Figur 5.1 Schematisk bild över Kalmanfilter.

Estimationsfelet  $\tilde{x} = x - \hat{x}$  bestäms av

$$\tilde{x}(t+1) = (\Phi - K\Theta)\tilde{x}(t) + v(t) - Ke(t) \quad (5.3)$$

enligt ekv. (5.1) och (5.2). Den tidsvariabla förstärkningsparametern  $K(t)$  har bestämts så att  $P(t) = \text{cov}\{\tilde{x}(t), \tilde{x}(t)\}$  blir minimal.

Egenvärdena till matrisen  $(\Phi - K\Theta)$  bestämmer hur snabbt estimationsfelet avtar. Om egenvärdena till  $(\Phi - K\Theta)$  ligger innanför enhetscirkeln så går estimationsfelet mot noll oberoende av valet av  $\hat{x}(t_0)$ . Ansätter vi  $K = 0$  kommer estimationsfelet att gå mot noll med samma tidskonstant som processen själv.

Då  $R_2 > 0$  och förutsättningarna på sekvenserna  $\{v(t)\}$  och  $\{e(t)\}$  är uppfyllda kan man visa att residualerna  $\{\epsilon(t)\}$  blir en sekvens oberoende gaussiska likafördelade stokastiska variabler med  $E\{\epsilon(t)\} = 0 \quad \forall t$  samt kovariansmatris  $R_\epsilon = \text{cov}\{\epsilon(t), \epsilon(t)\} = \theta P_\infty \theta^T + R_2 \quad \forall t$ , där  $P_\infty$  betecknar den stationära lösningen till ekvation (5.4).

Teorem 3:

Låt  $R_2 > 0$  samt  $\{v(t)\}$  och  $\{e(t)\}$  vara sekvenser av oberoende, sinsemellan oberoende, likafördelade gaussiska stokastiska variabler med medelvärden och kovariansmatriser  $E\{v(t)\} = E\{e(t)\} = 0 \quad \forall t$  respektive  $\text{cov}\{v(t), v(t)\} = R_1$ ,  $\text{cov}\{e(t), e(t)\} = R_2 \quad \forall t$ .

Då erhålles:

- (i)  $\{\epsilon(t)\}$  är en sekvens oberoende gaussiska likafördelade stokastiska variabler
- (ii)  $E\{\epsilon(t)\} = 0 \quad \forall t$
- (iii)  $\text{cov}\{\epsilon(t), \epsilon(t)\} = R_\epsilon = \theta P_\infty \theta^T + R_2 \quad \forall t$

Teoremet användes för teständamål i avsnitt 6.3. Notera att teoremet endast gäller i den ena riktningen.

## 5.2 Beräkning av Kalmanförstärkningen K.

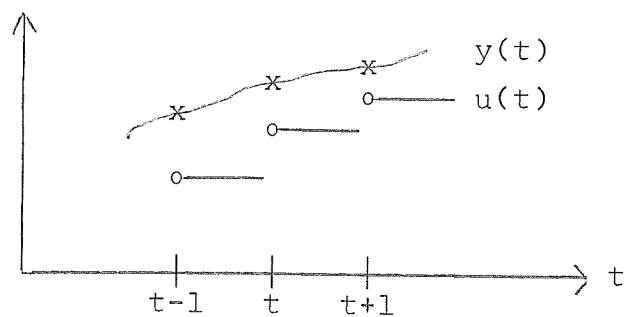
Estimationsfelen lyder differensekvationen (5.3). Förstärkningen  $K(t)$  kan beräknas ur de rekursiva ekvationerna (5.4) och (5.5):

$$P(t+1) = \Phi P(t) \Phi^T + R_1 - (\Phi P(t) \theta^T + R_{12}) (\theta P(t) \theta^T + R_2)^{-1} \cdot (\theta P(t) \Phi^T + R_{12}^T) \quad (5.4)$$

$$K(t) = (\Phi P(t) \theta^T + R_{12}) (\theta P(t) \theta^T + R_2)^{-1} \quad (5.5)$$

För att spara räknearbete för datorn under styrningen använder vi oss av ett så kallat asymptotiskt Kalmanfilter (vilket sker på bekostnad av optimaliteten). Detta erhålles genom att man itererar fram den stationära lösningen  $P_\infty$  till ekvation (5.4) och sedan använder man detta  $P_\infty$  i ekvation (5.5) och erhåller på så sätt en konstant Kalmanförstärkning.

Det är mycket viktigt att man vid rekonstruktion av tillståndsvariabler använder  $y$ - respektive  $u$ -värden härrörande från korrekt tidpunkt. Tidsföljden diskuteras nedan.



Diskret system

$$x(t+1) = \Phi x(t) + \Gamma u(t)$$

där

$x(t)$  är tillståndens värde vid tiden  $t$ ,

$u(t)$  är insignalens värde i intervallet  $[t, t+1)$

Estimator

$$\hat{x}(t+1) = \Phi \hat{x}(t) + \Gamma u(t) + K(y(t) - \theta \hat{x}(t))$$

där

$\hat{x}(t)$  är estimatet av  $x(t)$  vid tidpunkten  $t$  (beräknat vid tidpunkten  $t-1$ )

$u(t)$  är insignalens värde i intervallet  $[t, t+1)$

$y(t)$  är utsignalens värde vid tidpunkten  $t$ .

Vi har vidare vid dead-beat reglering på estimatet  $\hat{x}(t)$  att processens insignal beräknas enligt

$$u(t) = L(t)\hat{x}(t)$$

vilket betyder att ordningsföljden vid styrning blir

1) beräkna styrsignalen  $u(t)$

2) beräkna det nya estimatet  $\hat{x}(t+1)$ .



5.3 Egenvärdena till matrisen  $\Phi - K\theta$ .

Om paret  $(\Phi, \theta)$  är observerbart så kan matrisen  $K$  väljas så att egenvärdena till matrisen  $\Phi - K\theta$  antar föreskrivna värden.

Betrakta fallet då paret  $(\Phi, \theta)$  är icke observerbart. Kälmans kanoniska dekomponering implicerar att systemet genererat av paret  $(\Phi, \theta)$  genom ett byte av koordinater i tillståndsrummet  $z = T^{-1}x$  kan bringas på formen

$$z(t+1) = \begin{bmatrix} \Phi_{11} & 0 \\ \Phi_{12} & \Phi_{22} \end{bmatrix} z(t) = \Phi z(t)$$

$$y(t) = [\theta_1, 0] z(t) = \theta z(t)$$

I de nya koordinaterna bildar vi matrisen  $\Phi - K\theta$

$$\Phi - K\theta = \begin{bmatrix} \Phi_{11} & 0 \\ \Phi_{12} & \Phi_{22} \end{bmatrix} - \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} [\theta_1, 0] = \begin{bmatrix} \Phi_{11} - K_1 \theta_1 & 0 \\ \Phi_{12} & \Phi_{22} \end{bmatrix}$$

Då paret  $(\Phi_{11}, \theta_1)$  är observerbart, existerar ett val av  $K_1$  så att egenvärdena till  $\Phi_{11} - K_1 \theta_1$  antar föreskrivna värden. Vidare påverkar valet av  $K$  ej egenvärdena till matrisen  $\Phi_{22}$ . Vi finner således, då paret  $(\Phi, \theta)$  är icke observerbart, att vissa egenvärden till matrisen  $\Phi - K\theta$  kan givas föreskrivna värden medan andra ej beror på valet av  $K$ . De senare egenvärdena är innehållna i mängden av egenvärden till matrisen  $\Phi$ .

## 6. KALMANFILTRERING OCH STOKASTISK MODELL FÖR DIFFUSIONSPROCESSEN.

### 6.1 Stokastisk modell för diffusionsprocessen.

Den stokastiska kontinuerliga modellen för diffusionsprocessen blir:

$$\dot{x} = Ax + Bu + Bv' \quad (6.1)$$

där A och B definieras av (2.15), x är tillståndsvektorn, u är insignalvektorn samt v' är de till u hörande omvandlingsfelen härrörande från datorns D/A-omvandlare.

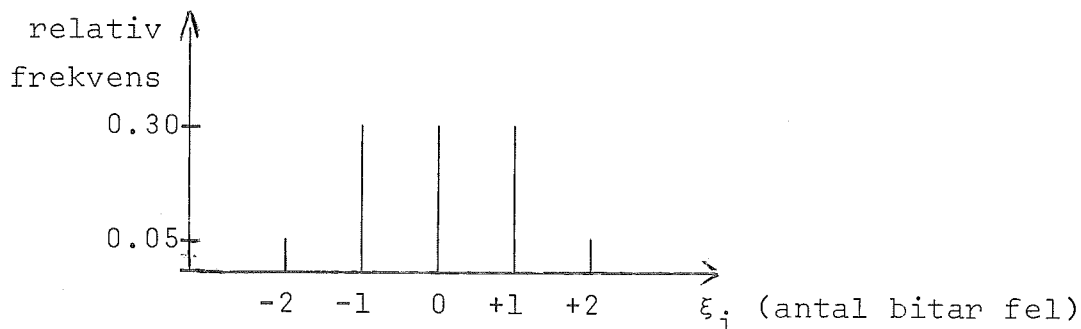
Eftersom v'(t) (liksom u(t)) är konstant i intervallet [t, t+1) blir den mot (6.1) svarande diskreta modellen:

$$x(t+1) = \Phi x(t) + \Gamma u(t) + \Gamma v'(t) \stackrel{\Delta}{=} \hat{x}(t) + \Gamma v(t) \quad (6.2)$$

Jämför med ekvation (5.1). v(t) blir alltså  $\Gamma v'(t)$ , där v'(t) antages vara en sekvens oberoende gaussiska likafördelade stokastiska variabler sådana att  $E\{v'(t)\} = 0 \quad \forall t$  samt  $\text{cov}\{v'(t), v'(t)\} = R_{v'v'} \quad \forall t$ . Då gäller att v(t) är en sekvens oberoende gaussiska likafördelade stokastiska variabler och  $E\{v(t)\} = 0 \quad \forall t$  samt  $R_{vv} = \Gamma R_{v'v'} \Gamma^T \quad \forall t$ .

## 6.2 Bestämning av de båda diskreta kovariansmatriserna R1D och R2D.

En felkurva för datorns A/D- och D/A-omvandlare upptogs:



Vi finner om vi betecknar antal bitar fel i A/D- och D/A-omvandlarna med  $\xi_1$  resp  $\xi_2$

$$E\{\xi_i\} = 0 \quad ; \quad i = 1,2$$

$$\sigma_i = \sqrt{V(\xi_i)} = \sqrt{\frac{1}{100}(2 \cdot 2^2 \cdot 5 + 2 \cdot 1^2 \cdot 30)} = 1. \quad ; \quad i = 1,2$$

I medeltal ger alltså datorns A/D- och D/A-omvandlare ett värde som är 0 bitar fel med standardavvikelsen 1 bit. Då vi förstärker alla från datorn utgående signaler 4 ggr (för att få bättre upplösning och för att reducera mätfelen) finner vi att en bit motsvaras av  $0.02/4 = 0.005$  °C.

Vi har vidare observerat att servona är behäftade med stationära fel vilka vi betecknar  $\zeta$ . Dessa antages likafördelade i intervallet  $[-a, a]$  dvs

$$f(x) = \begin{cases} 1/2a & x \in [-a, a] \\ 0 & \text{annars} \end{cases}$$

Vi finner:

$$m = E\{\zeta\} = \int_{-a}^a xf(x)dx = 0$$

$$\sigma^2 = V(\zeta) = \int_{-a}^a (x-m)^2 f(x)dx = \int_{-a}^a x^2 \cdot \frac{1}{2a} dx = a^2/3$$

och alltså  $\sigma = a/\sqrt{3}$ . Om vi antar att det stationära felet är  $a = 0.02$  får vi  $\sigma = 0.02/\sqrt{3} = 0.0115$ .

Vi är nu mogna att ange explicita uttryck på kovariansmatriserna R1D och R2D.

1) Mätbruset R2D.

Vi antar att mätfelen i huvudsak härrör från A/D-omvandlarna samt att mätfelen kan modularas som diskret gaussiskt vitt brus. Därmed blir:

$$R2D = \begin{bmatrix} (0.5 \cdot 10^{-2})^2 & 0 \\ 0 & (0.5 \cdot 10^{-2})^2 \end{bmatrix} \quad (6.3)$$

(ny x ny)

2) Processbruset R1D.

Till processbruset hänföres både servots stationära fel samt D/A-omvandlarnas fel.

Enligt tidigare har vi:

omvandlingsfel  $\xi_2$  ;  $V(\xi_2) = (0.5 \cdot 10^{-2})^2$

stationärt fel  $\zeta$  ;  $V(\zeta) = (1.15 \cdot 10^{-2})^2$

Då processerna  $\xi_2$  och  $\zeta$  antages oberoende erhåller vi:

$$V(\xi_2 + \zeta) = V(\xi_2) + V(\zeta) = (1.25 \cdot 10^{-2})^2$$

dvs  $\sigma_{\text{tot}} = 1.25 \cdot 10^{-2}$ . Slutligen får vi:

$$R1D = \Gamma \begin{bmatrix} (1.25 \cdot 10^{-2})^2 & 0 \\ 0 & (1.25 \cdot 10^{-2})^2 \end{bmatrix} \Gamma^T \quad (6.4)$$

### 6.3 Experimentella resultat under RSX.

#### 1) Uttestning av de olika Kalmanfiltren.

Följande olika filter har studerats:

FI311	GA311	TH311	AK311
FI311	GA311	TH312	AK312
FI311	GA311	TH313	AK313

Samtliga filter är baserade på modellen ROD3, med  $n_x = 9$  och  $n_u = 2$ , samplingsintervallet är 10 sek, vilket definierar paret (FI311, GA311). I AK311 antages endast stavens ändtemperaturer mätbara, i AK312 antages dessutom stavens mittpunktstemperatur mätbar, medan i AK313 samtliga stavens 9 tillstånd antages mätbara.

Först studerades de olika filtrens insvängningsförlopp, då  $\hat{x}(0) = 1$  och  $u(t) = y(t) = 0$  ;  $t = 1, 2, \dots$  . Under dessa förutsättningar lyder estimatet  $\hat{x}(t)$  differential-ekvationen

$$\begin{cases} \hat{x}(t+1) = (\phi - K\theta)\hat{x}(t) \\ \hat{x}(0) = 1 \end{cases}$$

I figurerna är initialfelet  $1^\circ\text{C}$ . Egenvärdena till matrisen  $(\phi - K\theta)$  finns inskrivna i tillhörande figur. Egenvärdena till  $\phi = \text{FI311}$  är:

0.944 809	
0.800 684	
0.621 454	
0.455 236	
0.318 044	
0.217 526	
0.159 806	
0.001 273	} egenvärden servo
0.001 273	

Vidare bifogas utskrifter på Kalmanförstärkning,  $P\text{-STAT} = P_\infty$  samt  $\theta P_\infty \theta^T + R_2$  för de tre olika filtren.

P-STAT

0.15585257E-03	0.36748664E-05	-0.72460248E-07	-0.55668249E-07
0.36163864E-04	0.15174405E-05	0.30229723E-06	0.14108239E-06
0.36748664E-05	0.66906883E-06	0.36236056E-06	0.22193202E-06
-0.72460248E-07	0.36236056E-06	0.30164396E-06	0.23664777E-06
-0.55668249E-07	0.22193202E-06	0.23664777E-06	0.23567113E-06
-0.43168438E-08	0.14670786E-06	0.18997242E-06	0.23664777E-06
0.16431364E-09	0.97156228E-07	0.14670786E-06	0.22193202E-06
0.22815674E-10	0.51094410E-07	0.84848645E-07	0.14108239E-06
0.00000000	0.16431364E-09	-0.43168438E-08	-0.55668249E-07

-0.43168438E-08	0.22815674E-10	0.00000000
0.84848645E-07	0.25197171E-07	0.22815674E-10
0.14670786E-06	0.51094410E-07	0.16431364E-09
0.18997242E-06	0.84848645E-07	-0.43168438E-08
0.23664777E-06	0.14108239E-06	-0.55668249E-07
0.30164396E-06	0.30229723E-06	-0.72460248E-07
0.36236056E-06	0.15174405E-05	0.36748664E-05
0.30229723E-06	0.93217357E-05	0.36163864E-04
-0.72460248E-07	0.36748664E-05	0.15585257E-03

KALMANGAIN AK311

0.10967123E-02	0.00000000	0.16065257E-03	0.00000000
0.13429129	-0.34491384E-05	0.00000000	0.18085257E-03
0.62878000E-01	-0.35345457E-04	0.00000000	
0.12274603E-01	-0.14835053E-03		
0.75035360E-03	0.75035360E-03		
-0.14835053E-03	0.12274603E-01		
-0.35345457E-04	0.62878000E-01		
-0.34491384E-05	0.13429129		
0.00000000	0.10967123E-02		

TH\*PSTAT\*TH+R2

0.16065257E-03	0.00000000
0.00000000	0.18085257E-03

P-STAT

0.15585257E-03	0.36748665E-05	0.36163864E-04	0.72460168E-07	-0.55668166E-07
0.36163864E-04	0.15111759E-05	0.93181634E-05	0.29447990E-06	0.13277759E-06
0.36748665E-05	0.65804527E-06	0.15111759E-05	0.34856225E-06	0.20725364E-06
-0.72460168E-07	0.34856225E-06	0.29447990E-06	0.28432309E-06	0.21819805E-06
-0.55668166E-07	0.20725364E-06	0.13277759E-06	0.21819805E-06	0.21600458E-06
-0.43167624E-08	0.13292471E-06	0.77042209E-07	0.17266221E-06	0.21819805E-06
0.16438627E-09	0.86154255E-07	0.44845328E-07	0.13292471E-06	0.20725363E-06
0.22861874E-10	0.44845327E-07	0.21636235E-07	0.77042206E-07	0.13277759E-06
-0.38039563E-17	0.16438627E-09	0.22861874E-10	-0.43167624E-08	-0.55668166E-07

-0.43167624E-08	0.22861874E-10	0.16438627E-09	-0.38039563E-17	
0.77042209E-07	0.21636235E-07	0.44845328E-07	0.22861874E-10	
0.13292471E-06	0.44845327E-07	0.86154255E-07	0.16438627E-09	
0.17266221E-06	0.77042206E-07	0.13292471E-06	-0.43167624E-08	
0.21819805E-06	0.13277759E-06	0.20725363E-06	-0.55668166E-07	
0.28432307E-06	0.29447988E-06	0.34856222E-06	-0.72460168E-07	
0.34856222E-06	0.15111758E-05	0.65804522E-06	0.36748665E-05	
0.29447988E-06	0.93181634E-05	0.15111758E-05	0.36163864E-04	
-0.72460168E-07	0.36163864E-04	0.36163864E-04	0.15585257E-03	

P-STAT (=den stationära lösningen till ekv.(5.4)) då stavens ändtemperaturer samt stavens mittpunktstemperatur antages tillgängliga för mätning.



KALMANGAIN AK312

0.10967121E-02	-0.38842742E-06	-0.11956172E-09
0.13429275	0.47167594E-02	-0.19970552E-05
0.62880281E-01	0.74150850E-02	-0.33062661E-04
0.12277363E-01	0.83155490E-02	-0.14579049E-03
0.75295390E-03	0.84463218E-02	0.75295393E-03
-0.14579049E-03	0.83155488E-02	0.12277363E-01
-0.33062660E-04	0.74150849E-02	0.62880283E-01
-0.19970553E-05	0.47167592E-02	0.13429275
-0.11956171E-09	-0.38842740E-06	0.10967122E-02

TH\*PSTAT\*THT+R2

0.18085257E-03	-0.55668166E-07	-0.38039563E-17
-0.55668166E-07	0.25216001E-04	-0.55668166E-07
-0.38039563E-17	-0.55668166E-07	0.18085257E-03

P-STAT

0.15585257E-03	0.36163553E-04	0.36746603E-05	-0.72543546E-07	-0.55698039E-07
0.36163553E-04	0.91752165E-05	0.13711595E-05	0.19646347E-06	0.68259698E-07
0.36746603E-05	0.13711595E-05	0.50231577E-06	0.22288358E-06	0.11287511E-06
-0.72543546E-07	0.19646347E-06	0.22288358E-06	0.16588719E-06	0.11504948E-06
-0.55698039E-07	0.68259698E-07	0.11287511E-06	0.11504948E-06	0.11152116E-06
-0.43283610E-06	0.33893714E-07	0.62428823E-07	0.84647677E-07	0.11504948E-06
0.15947485E-09	0.16561557E-07	0.35729824E-07	0.62428823E-07	0.11287511E-06
0.21030325E-10	0.67361073E-08	0.16561557E-07	0.33893714E-07	0.68259698E-07
0.42522694E-15	0.21030325E-10	0.15947485E-09	-0.43283610E-08	-0.55698039E-07
-0.43283610E-08	0.15947485E-09	0.21030325E-10	0.42522694E-15	
0.33893714E-07	0.16561557E-07	0.67361073E-08	0.21030325E-10	
0.62428823E-07	0.35729824E-07	0.16561557E-07	0.19647485E-09	
0.84647677E-07	0.62428823E-07	0.33893714E-07	-0.43283610E-08	
0.11504948E-06	0.11287511E-06	0.68259698E-07	-0.55698039E-07	
0.16588719E-06	0.22288358E-06	0.19646347E-06	-0.72543546E-07	
0.22288358E-06	0.50231577E-06	0.13711595E-05	0.36746603E-05	
0.19646347E-06	0.13711595E-05	0.91752165E-05	0.36163553E-04	
-0.72543546E-07	0.36746603E-05	0.36163553E-04	0.15585257E-03	

P-STAT (=den stationära lösningen till ekv.(5.4)) då samtliga stavens 9 tillstånd antages tillgängliga för mätning.

KALMANGAIN AK313

0.10491856E-02  
0.12451717  
0.56399873E-01  
0.96566791E-02  
-0.18596819E-03  
-0.51034840E-03  
-0.18743261E-03  
-0.59564248E-04  
0.13365261E-07  
0.23568434E-03  
0.47298004E-01  
0.30867953E-01  
0.12142873E-01  
0.41816751E-02  
0.15546302E-02  
0.63217548E-03  
0.22822525E-03  
-0.54111498E-07  
0.19562098E-04  
0.15661407E-01  
0.15199512E-01  
0.95151770E-02  
0.50754165E-02  
0.26474576E-02  
0.13707382E-02  
0.57817067E-03  
-0.15619903E-06  
-0.26396640E-05  
0.55869315E-02  
0.73343011E-02  
0.62650581E-02  
0.45806331E-02  
0.32609178E-02  
0.22098582E-02  
0.11225737E-02  
-0.37824402E-06  
-0.11981856E-05  
0.23262735E-02  
0.37807800E-02  
0.42280856E-02  
0.42627028E-02  
0.42280856E-02  
0.37807801E-02  
0.23262736E-02  
-0.11981856E-05

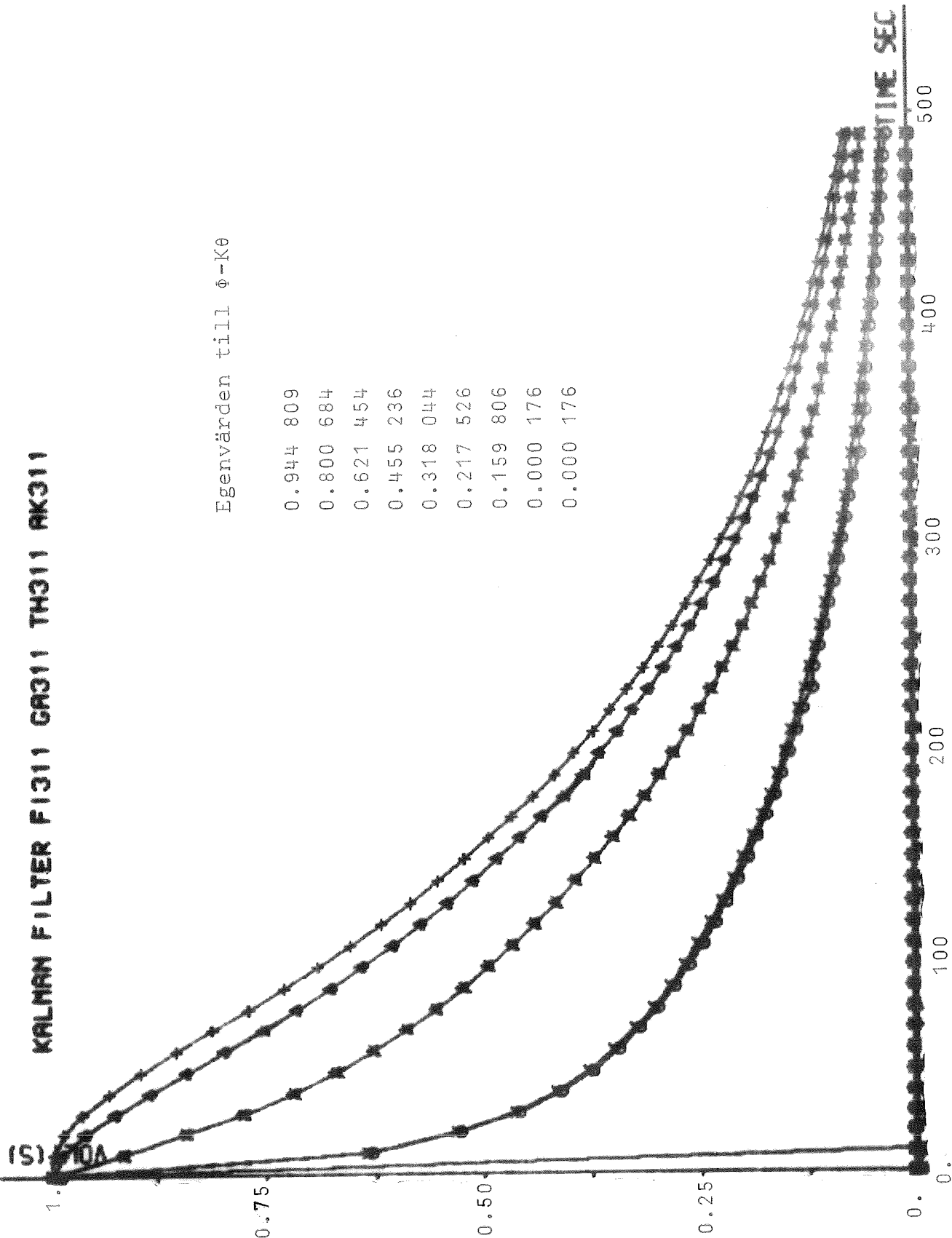
-0.37824402E-06  
0.11225737E-02  
0.22098582E-02  
0.32609178E-02  
0.45806332E-02  
0.62650582E-02  
0.73343012E-02  
0.55869317E-02  
-0.26396640E-05  
-0.15619903E-06  
0.57817067E-03  
0.13707382E-02  
0.26474576E-02  
0.50754164E-02  
0.95151768E-02  
0.15199512E-01  
0.15661407E-01  
0.19562096E-04  
-0.54111499E-07  
0.22822526E-03  
0.63217552E-03  
0.15546302E-02  
0.41816753E-02  
0.12142873E-01  
0.30867954E-01  
0.47298003E-01  
0.23568435E-03  
0.13365261E-07  
-0.59564250E-04  
-0.18743262E-03  
-0.51034841E-03  
-0.18596823E-03  
0.96566789E-02  
0.56399873E-01  
0.12451717  
0.10491856E-02

TH\*PSTAT\*TH+R2

0.18085257E-03	0.36163553E-04	0.36746603E-05	-0.72543546E-07	-0.55698039E-07
0.36163553E-04	0.34175213E-04	0.13711595E-05	0.19646347E-06	0.68259698E-07
0.36746603E-05	0.13711595E-05	0.25502312E-04	0.22288358E-06	0.11287511E-06
-0.72543546E-07	0.19646347E-06	0.22288358E-06	0.25165884E-04	0.11504948E-06
-0.55698039E-07	0.68259698E-07	0.11287511E-06	0.11504948E-06	0.25111518E-04
-0.43283610E-08	0.33893714E-07	0.62428823E-07	0.84647677E-07	0.11504948E-06
0.15947485E-09	0.16561557E-07	0.35729824E-07	0.62428823E-07	0.11287511E-06
0.21030325E-10	0.67361073E-08	0.16561557E-07	0.33893714E-07	0.68259698E-07
0.42522694E-15	0.21030325E-10	0.15947485E-09	-0.43283610E-08	-0.55698039E-07
-0.43283610E-08	0.15947485E-09	0.21030325E-10	0.42522694E-15	
0.33893714E-07	0.16561557E-07	0.67361073E-08	0.21030325E-10	0.21030325E-10
0.62428823E-07	0.35729824E-07	0.35729824E-07	0.62428823E-07	0.21030325E-10
0.84647677E-07	0.62428823E-07	0.33893714E-07	-0.43283610E-08	0.15947485E-09
0.11504948E-06	0.11287511E-06	0.68259698E-07	-0.55698039E-07	0.68259698E-07
0.25165884E-04	0.22288358E-06	0.19646347E-06	-0.72543546E-07	0.36746603E-05
0.22288358E-06	0.25502312E-04	0.13711595E-05	0.36163553E-04	0.36163553E-04
0.19646347E-06	0.13711595E-05	0.34175213E-04	0.19646347E-06	0.18085257E-03
-0.72543546E-07	0.34175213E-04	0.36163553E-04	0.36746603E-05	
-0.55698039E-07	0.36163553E-04	0.36163553E-04	0.36746603E-05	

KALMAN ESTIMATION 20 SEPT 74

KALMAN FILTER F1311 GA311 TH311 AK311

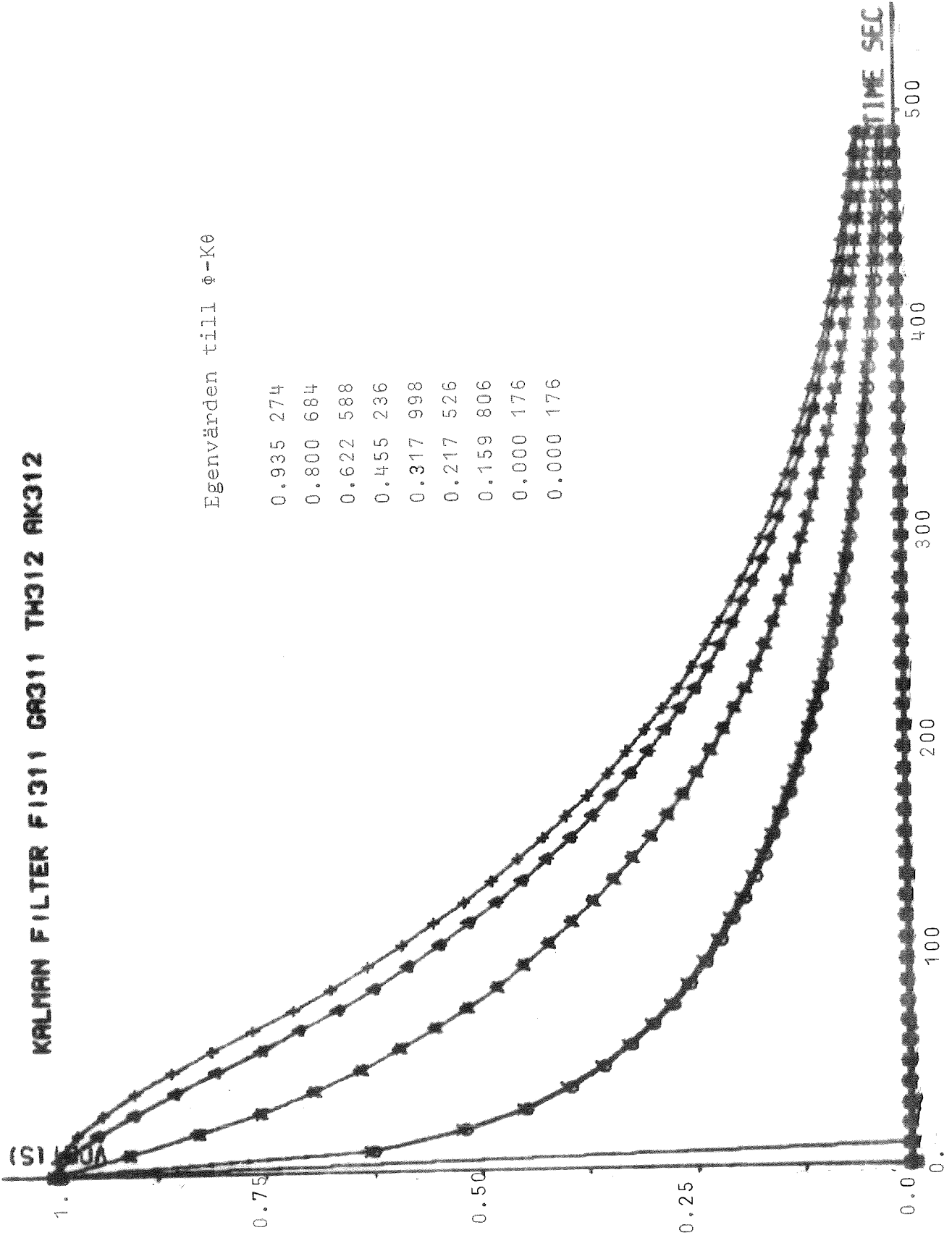


Eigenvärden till  $\phi-K_0$

0.944	809
0.800	684
0.621	454
0.455	236
0.318	044
0.217	526
0.159	806
0.000	176
0.000	176

KALMAN ESTIMATION 20 SEPT 74

KALMAN FILTER F1311 GA311 TH312 AK312

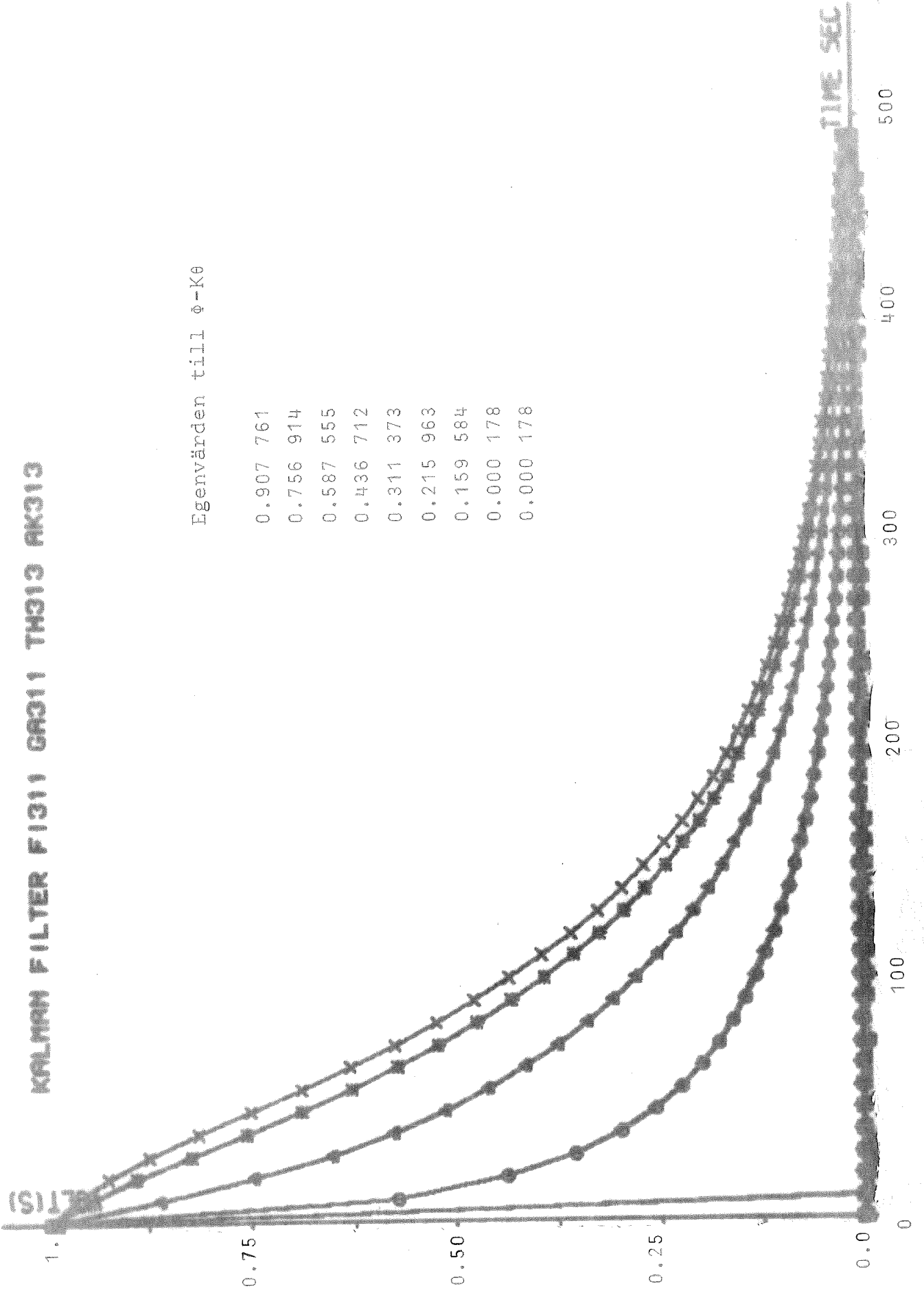


Egenvärden till  $\hat{\phi}-K\hat{\theta}$

0.935	274
0.800	684
0.622	588
0.455	236
0.317	998
0.217	526
0.159	806
0.000	176
0.000	176

KALMAN ESTIMATION 1 OCT 74

KALMAN FILTER F1311 GR311 TH313 AK313



Vi noterar att insvängningsförloppet blir snabbare då fler tillstånd antages mätbara, jämför med största egenvärde till matriserna  $\Phi$ - $K_0$  i de tre olika fallen.

För att visa hur filtren beter sig då insignalerna ändras har vi gjort 2 experiment där initialfelet satts till  $0.35^\circ\text{C}$  resp  $1.0^\circ\text{C}$  och vi efter hand ändrat insignalerna till processen  $0.35^\circ\text{C}$  resp  $1.0^\circ\text{C}$ .

Av figurerna framgår tydligt att servona är behäftade med stationära fel samt att de är olinjära. Då insignalerna till processen endast ändras  $0.35^\circ\text{C}$  saknar servona översläng medan en översläng på ca 4% erhålles då insignalerna till processen ändras  $1.0^\circ\text{C}$ . Det gjorda antagandet om servodynamiken blir därmed endast giltigt då insignalerna till processen ändras mindre än ca  $0.5^\circ\text{C}$ .

Vi ser vidare att estimationsfelen hörande till servona, i intervallet efter det att insignalen ställts ut, blir relativt stora, då processens insignal ändras  $1.0^\circ\text{C}$ , medan övriga estimationsfel endast påverkas i ringa omfattning. Vi observerar att estimationsfelen hörande till servona är lika stora som servots översläng vid motsvarande tidpunkt, dvs estimationsfelen beror på att vi inte lyckats beskriva servodynamiken exakt i det betraktade fallet.

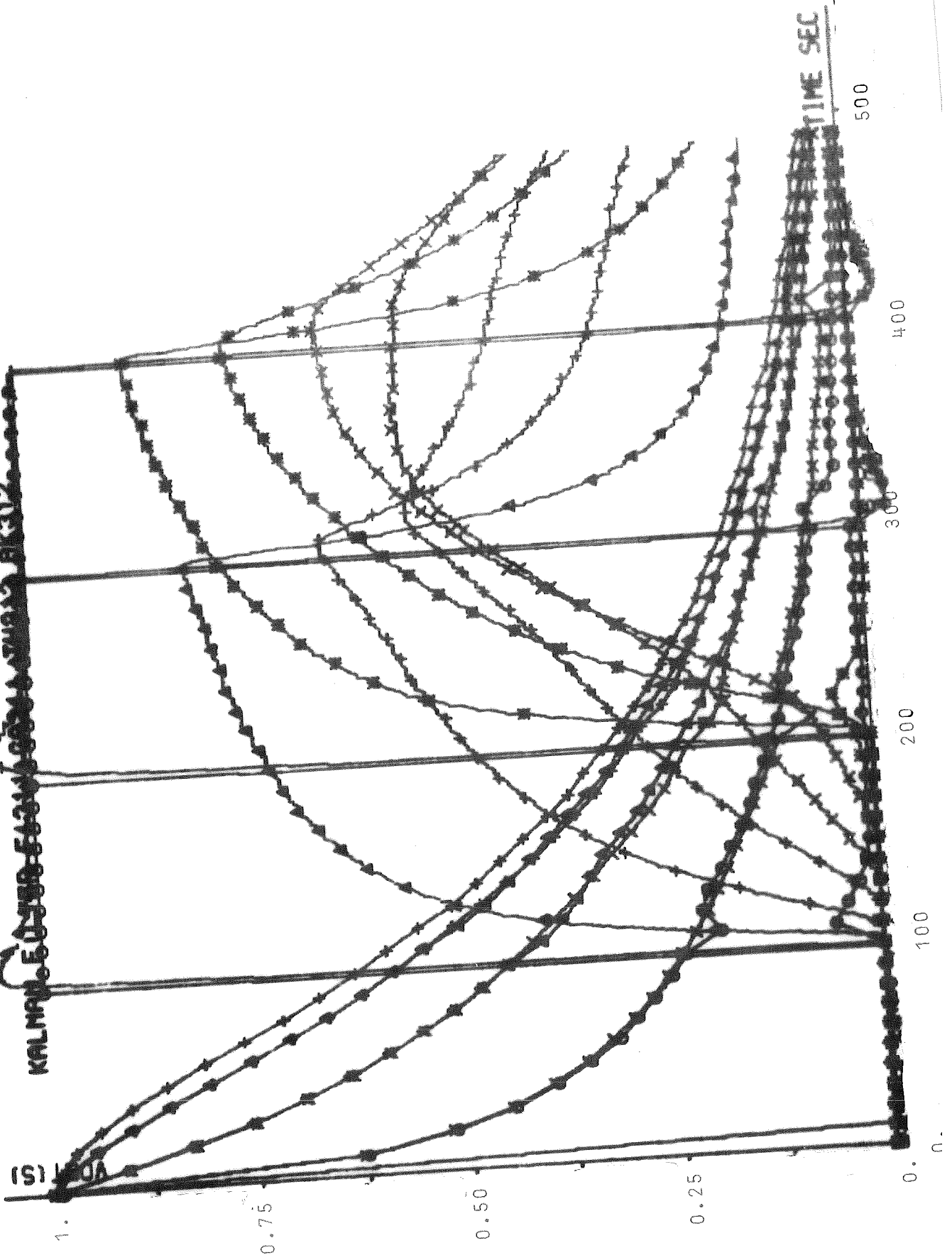


74

20 SEPT

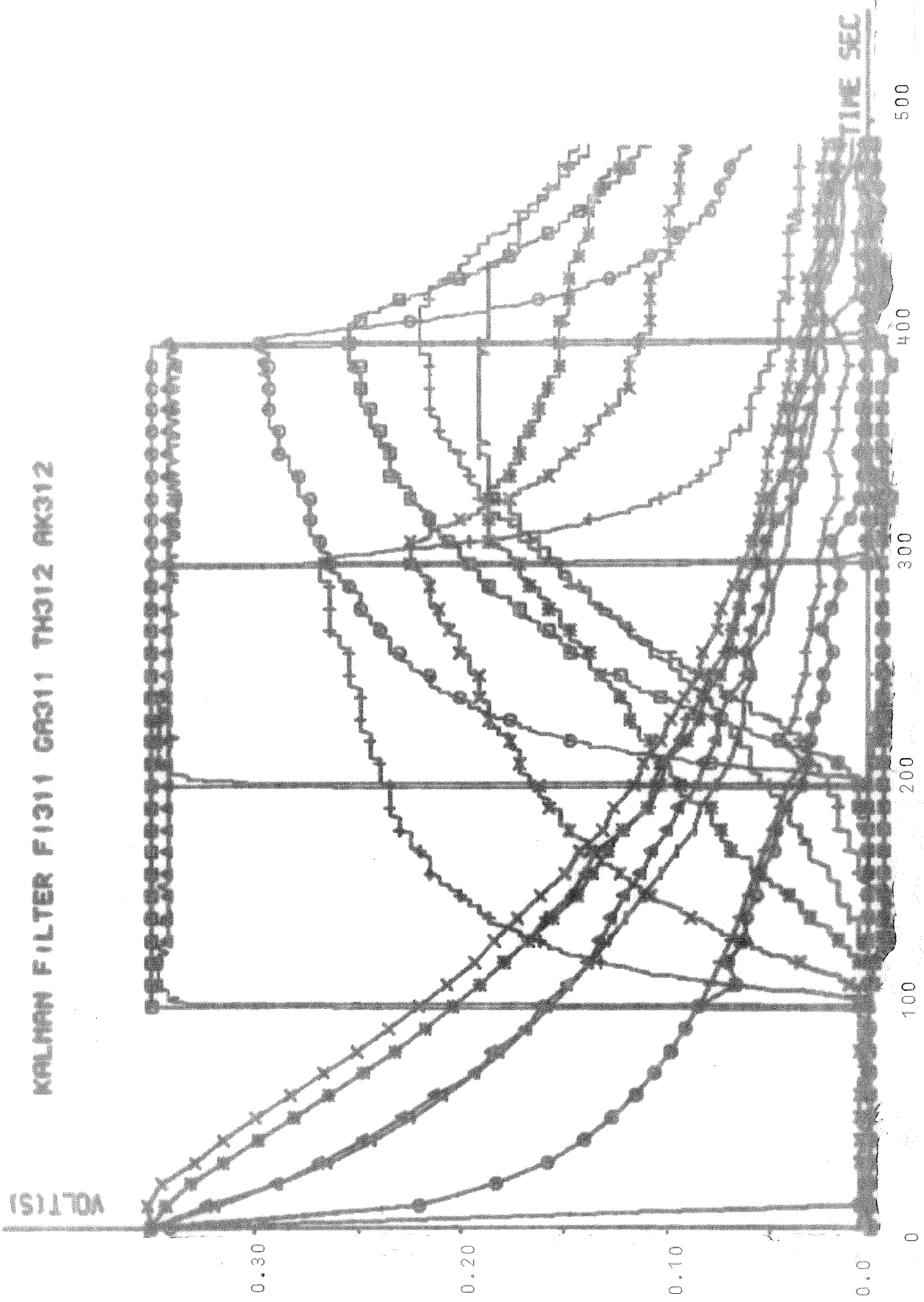
KALMAN ESTIMATION

KALMAN E-0388-5434-0004-20912 AK312



KALMAN ESTIMATION 1 OCT 74

KALMAN FILTER F1311 GR311 TH312 AK312



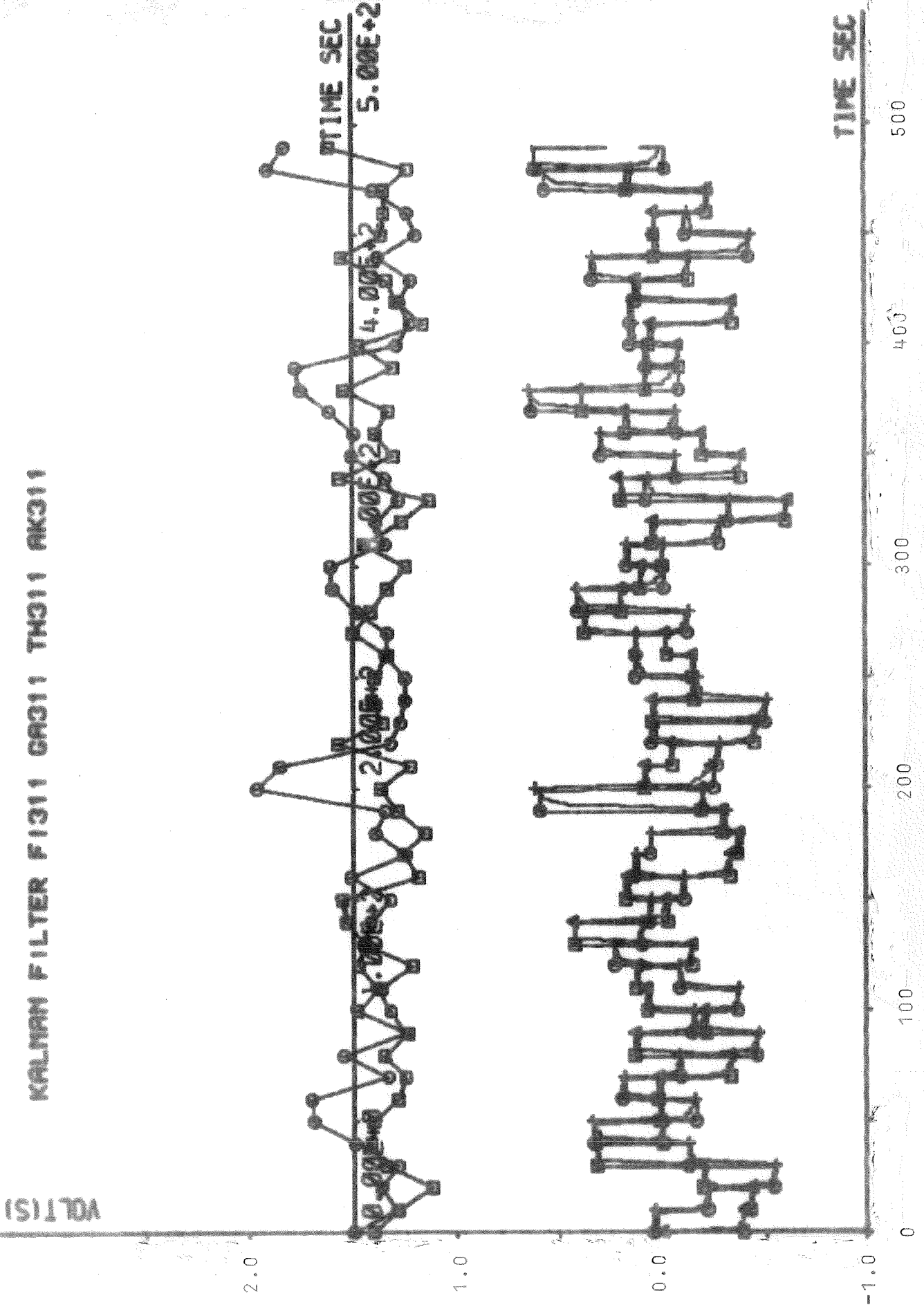
För att jämföra de tre Kalmanfiltren efter insvängningsförloppet skrev vi ett program kallat NOISE (se appendix D) som var tionde sekund ställer ut normalfördelade insignaler, sinsemellan oberoende, med specificerad standardavvikelse.

I figurerna finns enbart de tillståndsvariabler och estimationsfel inritade som motsvarar mätbara tillstånd. I alla tre fallen gäller att insignalernas standardavvikelse är  $0.25^{\circ}\text{C}$ . Det låga valet av bruskillornas standardavvikelse får till följd att den antagna modellen för servona är realistisk.

Vidare gäller i samtliga tre fall att estimationsfelen är förstärkta 20 gånger samt att deras nollpunkt är förlagd till  $+1.5$ .

KALMAN ESTIMATION 1 OCT 74

KALMAN FILTER F1311 GA311 TH311 AK311



Statistisk test av residualerna då endast stavens ändtemperaturer antages tillgängliga för mätning.

A

res nr	1	2
m (°C)	-0.007306	-0.002996
$\sigma$ (°C)	0.006137	0.009796
$\chi^2_{\text{norm}}$	17.135	23.547
$\chi^2_{\text{korr}}$	2.769	25.306

där  $\chi^2_{\text{norm}}$  är  $\chi^2$ -testkvantiteten för normalitet hos residualerna (17 frihetsgrader, ensidigt konfidensintervall (5%) :  $0 \leq \chi^2 \leq 27.6$  )

samt  $\chi^2_{\text{korr}}$  är  $\chi^2$ -testkvantiteten för icke korrelerade residualer (5 frihetsgrader, ensidigt konfidensintervall (5%) :  $0 \leq \chi^2 \leq 11.07$  )

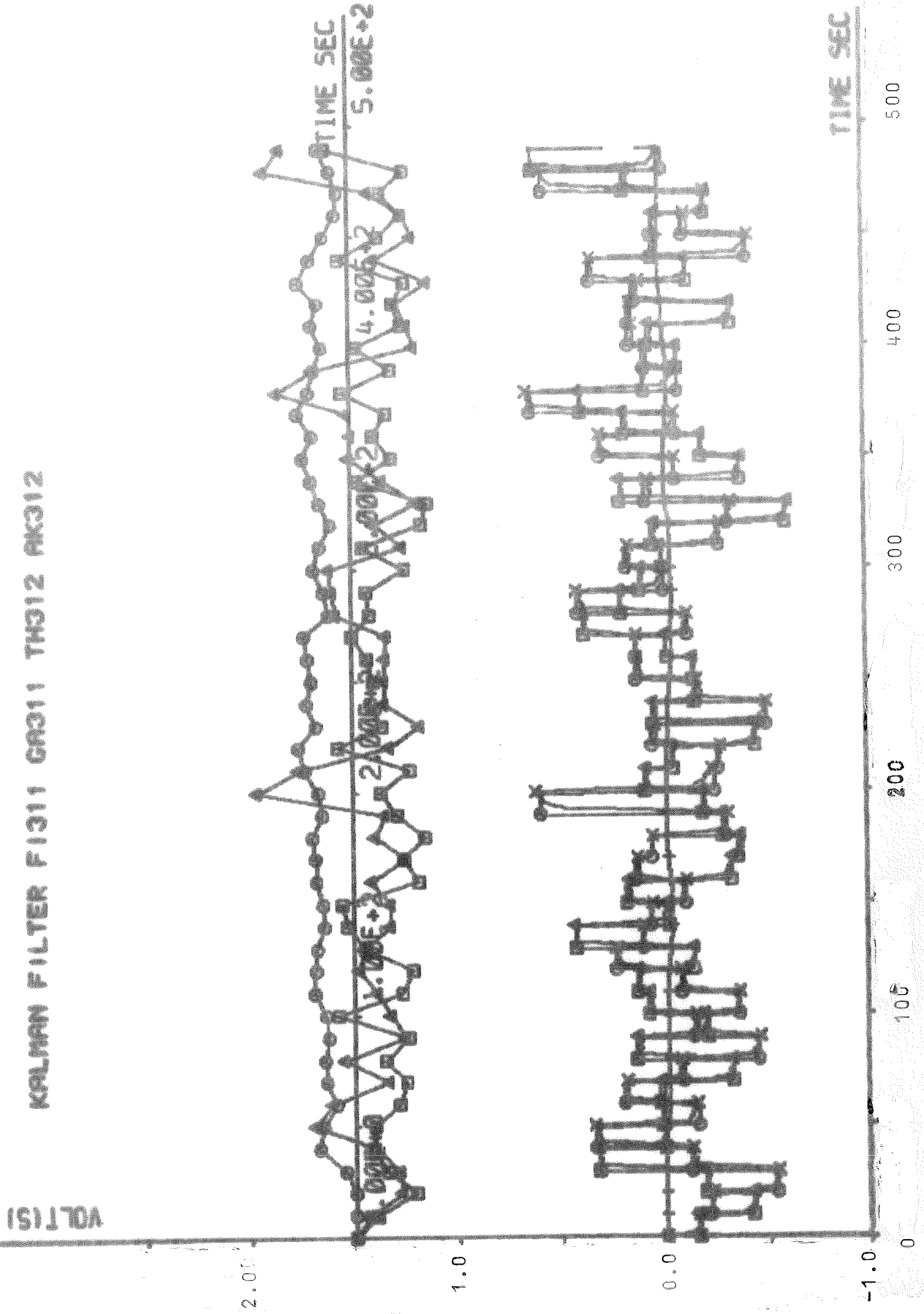
Kovariansmatrisen  $R_\epsilon$  för residualerna blev:

$$\begin{bmatrix} 0.376591\text{E-}04 & 0.183609\text{E-}05 \\ 0.183609\text{E-}05 & 0.959625\text{E-}04 \end{bmatrix}$$

Residualerna nr 2 är ej oberoende samt kovariansmatrisen  $R_\epsilon$  är signifikant skild från den teoretiska kovariansmatrisen (TH·PSTAT·THT + R2 sid 56).

KALMAN ESTIMATION 1 OCT 74

KALMAN FILTER F1311 GR311 TH312 AK312



Statistisk test av residualerna då förutom stavens ändtemperaturer även stavens mittpunktstemperatur antages mätbar.

B

res nr	1	2	3
m (°C)	-0.007102	0.007713	-0.003726
$\sigma$ (°C)	0.006235	0.003113	0.009902
$\chi^2_{\text{norm}}$	18.932	35.682	18.210
$\chi^2_{\text{korr}}$	2.897	31.045	20.901

där  $\chi^2_{\text{norm}}$  och  $\chi^2_{\text{korr}}$  definieras på sid 70.

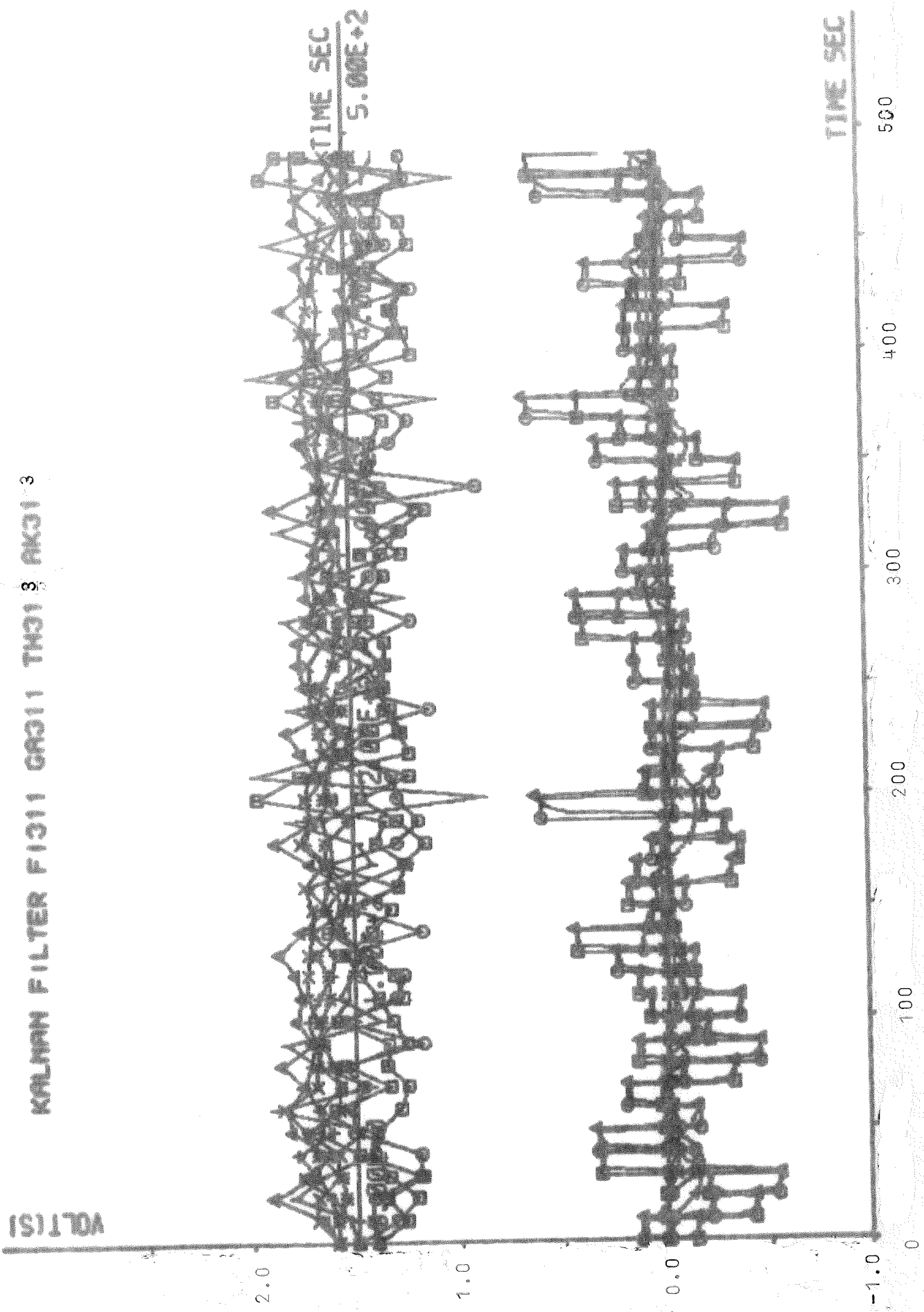
Kovariansmatrisen  $R_e$  för residualerna blev:

$$\begin{bmatrix} 0.388730E-04 & 0.149331E-05 & 0.721108E-05 \\ 0.149331E-05 & 0.968975E-05 & 0.349808E-06 \\ 0.721108E-05 & 0.349808E-06 & 0.980549E-04 \end{bmatrix}$$

Residualerna nr 2 och 3 är ej oberoende. Residualerna nr 2 är dessutom ej normalfördelade samt kovariansmatrisen  $R_e$  är signifikant skild från den teoretiska kovariansmatrisen (TH·PSTAT·THT + R2 sid 58).

KALMAN ESTIMATION 1 OCT 74

KALMAN FILTER F1311 GR311 TH313 AK313





Statistisk test av residualerna då stavens samtliga tillstånd antages mätbara.

C

res nr	1	2	3	4	5	6	7	8	9
m (°C)	0.00651	-0.00649	0.01045	0.00756	0.00861	0.00407	0.00269	0.00288	-0.00351
$\sigma$ (°C)	0.00624	0.00800	0.00454	0.00307	0.00310	0.00351	0.00416	0.01151	0.00991
$\chi_{\text{norm}}^2$	9.918	20.918	11.092	14.110	18.097	17.734	8.711	24.308	24.093
$\chi_{\text{korrr}}^2$	5.805	11.059	6.181	3.894	16.064	37.047	5.003	13.731	19.714

där  $\chi_{\text{norm}}^2$  och  $\chi_{\text{korrr}}^2$  definieras på sid 70.

Kovariansmatrisen för residualerna listas ej pga utrymmesskäl. Dock kan det sägas att det principiella utseendet överensstämmer med  $R_{\epsilon_{\text{AK311}}}$  och  $R_{\epsilon_{\text{AK312}}}$ .

Ur tabellen ovan finner vi att residualerna nr 5,6,8 och 9 ej är oberoende.

Kommentar:

Vi observerar att de tre filtren uppför sig ungefär likartat. Således efter insvängningsförloppet erhålles i vårt fall ingen vinst om vi utöver ändtemperaturerna utökar antalet mätbara tillståndsvariabler. Detta beror på att störningarna på vår process i huvudsak introduceras via servona.

Samtliga förutsättningar (i),(ii),(iii) om residualerna  $\{\epsilon(t)\}$  i teorem 3 är ej uppfyllda i något fall A,B,C. Emellertid gäller att residualerna approximativt är gaussiska och oberoende. Överensstämmelsen mellan teoretiska och uppmätta värden på kovariansmatrisen är dock mindre god. Detta orsakas av att servomodellen är för enkel. Notera att standardavvikelsen för residualerna i stavens mittpunkt är mindre än den teoretiska standardavvikelsen. Detta beror på att insignalerna ej exiterar mittpunktstemperaturen tillräckligt.

2) Dead-beat reglering av diffusionsprocessen, där styrningen sker på estimerade tillståndsvariabler.

Här studeras styrlagarna HRF31 och HRF36, båda definierade i avsnitt 4. I båda fallen har Kalmanfiltret FI311 GA311 TH311 AK311 använts. Därmed antages endast stavens ändtemperaturer mätbara. I övrigt är experimenten identiska med de som utfördes under avsnitt 4.2.

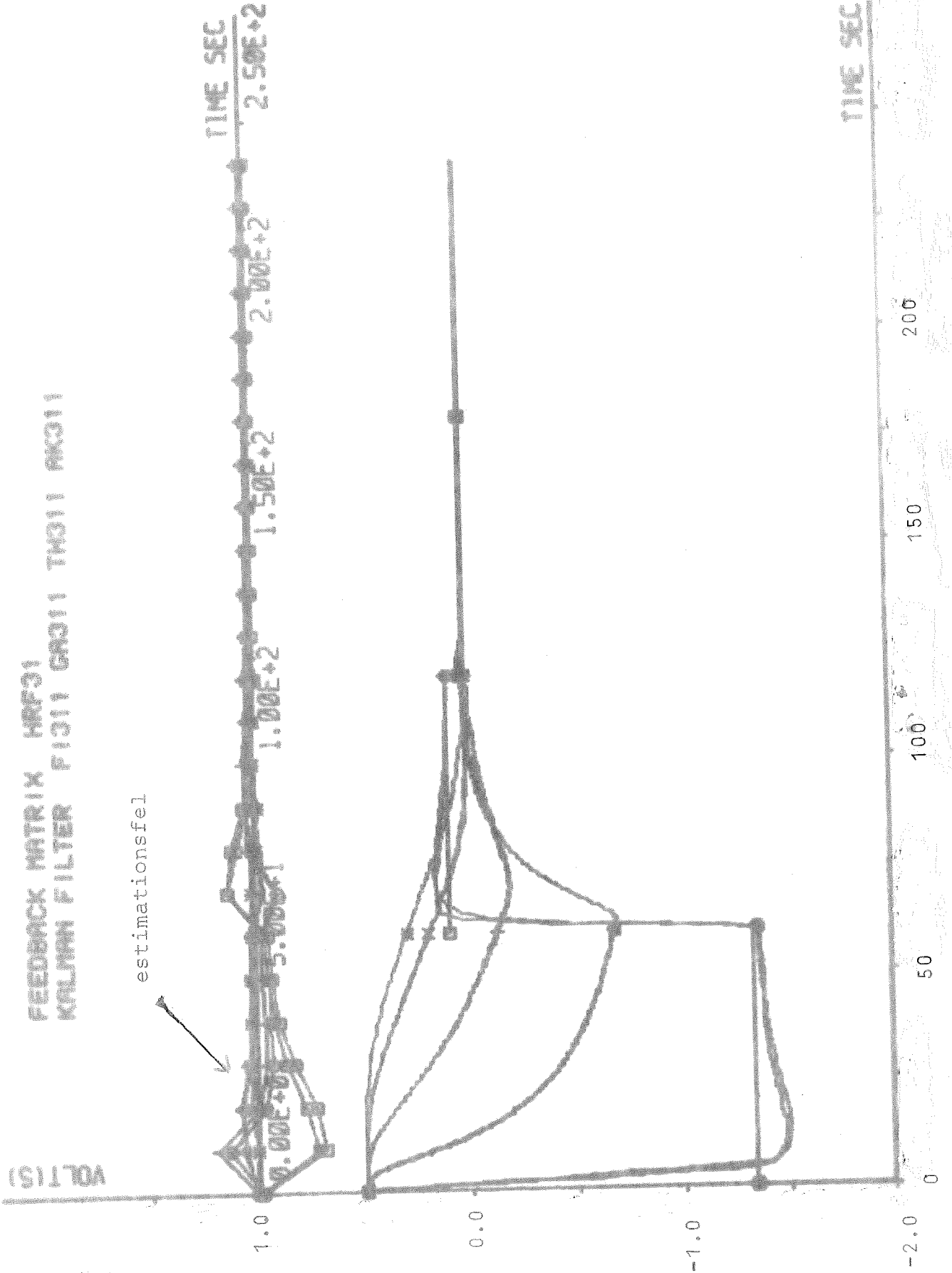
I figurerna finns förutom insignaler, servosignaler och processens tillståndsvariabler även estimationsfelen, vilka är förstärkta 2 gånger.

Vi observerar det anmärkningsvärda faktum att styrningen går lika bra nu som då vi mätte processens tillståndsvariabler.

Notera att estimationsfelen är approximativt noll då en ny insignal till processen ställes ut. Detta är en följd av att samplingsintervallet för dead-beat regulatorn är förhållandevis långt. Kalmanfiltret behöver därmed ej beskriva processens snabba dynamik noggrant. Detta är en distinkt fördel hos dead-beat regulatorn.

PROFILE OF DIFFUSION PROCESS 4 OCT 1974 (19:42: 0)

FEEDBACK MATRIX HRF31  
KALMAN FILTER F1311 G3311 TH011 AK311



PROFILE OF DIFFUSION PROCESS  
 \*\*\*\*\*

INITIAL PROFILE U1= 0.500 U2= 0.500

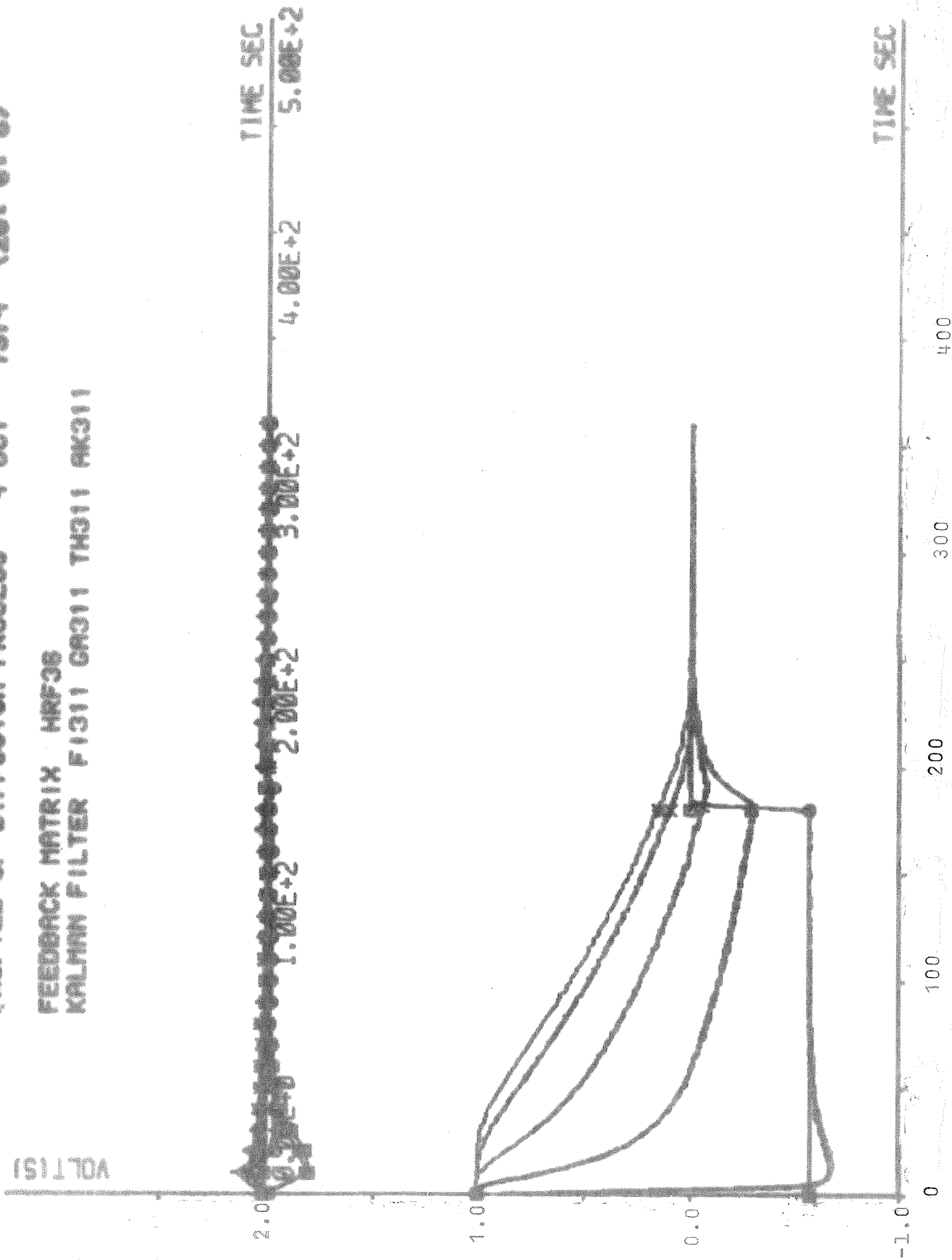
KALMAN FILTER F1311 GA311 TH311 AK311

TIME	U1	U2	Y1	Y2	Y3	Y4	Y5	Y6	Y7	YE1	YE2
0.000	-1.336	-1.335	0.488	0.503	0.503	0.498	0.498	0.498	0.498	0.498	0.493
60.000	0.089	0.092	-0.688	-0.132	0.190	0.293	0.186	-0.142	-0.684	-1.353	-1.362
120.000	0.004	0.004	0.015	0.000	0.000	0.000	0.000	-0.005	0.024	0.083	0.083
180.000	0.006	0.006	0.000	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000
240.000	0.003	0.003	0.000	0.005	0.005	0.000	0.000	0.000	0.005	0.000	0.000

PROFILE OF DIFFUSION PROCESS 4 OCT 1974 (20:01:0)

FEEDBACK MATRIX HRF36

KALMAN FILTER F1311 GR311 TH311 AK311



PROFILE OF DIFFUSION PROCESS  
 \*\*\*\*\*

INITIAL PROFILE U1= 1.007 U2= 1.007

KALMAN FILTER F1311 GA311 TH311 AK311

TIME	U1	U2	Y1	Y2	Y3	Y4	Y5	Y6	Y7	YE1	YE2
0.000	-0.566	-0.566	0.996	1.011	1.006	1.006	1.006	1.006	1.006	1.006	1.006
180.000	0.001	0.001	-0.298	-0.049	0.098	0.151	0.093	-0.059	-0.288	-0.566	-0.571
360.000	0.001	0.001	-0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

## 7. PROGRAMBESKRIVNINGAR.

## 7.1 RODPAC.

Programpaketet för beräkning av bl. a. Kalmanfilter har givits namnet RODPAC och är interaktivt. De olika kommandona och en kortfattad beskrivning av dessa framgår av listningen nedan.

ROD1	} Genererar olika modeller av diffusionsprocessen.
ROD2	
ROD3	
ROD4	
SAMP	Samplar ett kontinuerligt system.
THETA	Genererar en $\theta$ -matris.
CMTX	Genererar en diagonal kovariansmatris.
TRANS	Transformerar kontinuerliga kovariansmatriser till diskreta kovariansmatriser.
KALFI	Beräknar (med beteckningar enligt avsnitt 5) $P_{\infty}, K, \Phi - K\theta, \theta P_{\infty} \theta^T + R_2$ samt egenvärden till $\Phi - K\theta$ .
EIGEN	Beräknar egenvärdena till en matris.
SYST	Genererar ett system innehållande 4 filnamn.
ZEROM	Genererar en nollmatris.
EXTEN	Utvidgar en matris.



MATOP           Utför aritmetiska operationer på matriser.

DEFI            Ändrar värdet på specificerad parameter.

STOP            Avbryter körningen.

Samtliga kommandon förutom STOP innehåller ett antal argument. En mer detaljerad beskrivning av kommandostrukturen framgår av listningen i appendix B.

## 7.2 DB.

Programpaketet för dead-beat reglering under RSX har givits namnet DB och är interaktivt. De olika kommandona och en kortfattad beskrivning av dessa framgår av listningen nedan.

INIT            initialiserar regleringen genom att användaren anger antal samplingsintervall som experimentet skall pågå, förstärkning på yttre interface till datorn samt stavens initialprofil.

FBCK           läser in en specificerad feedback-matris från dec-tape.

AXIS           läser in specificerad plottningstäthet, maxvärden på x- och y-axel samt ritar axlar på tv-skärmen. Dessutom finns vid samkörning med SKEST möjlighet att plotta estimationsfelen förstärkta kring separat x-axel.

EXEC           läser in önskad slutprofil samt styr processen till denna profil. Vidare plottas temperaturen i 7 punkter längs staven, de 2 servosignalerna samt de 2 insignalerna. Om Kalmanestimering utföres via programpaketet SKEST plottas även estimationsfelen.

PRINT skriver ut feedback-matris, initialprofil samt insignalernas, servosignalernas och stavens temperaturprofil i samplingstidpunkterna efter utförd reglering.

AXIS avbryter exekveringen.

En mer detaljerad beskrivning av kommandostrukturen framgår av programlistningen i appendix C. Programpaketet DB kan köras endera separat eller tillsammans med programpaketet SKEST.

### 7.3 KEST och SKEST.

Programpaketen för Kalmanestimering har givits namnen KEST och SKEST. KEST är en större variant som innehåller plottningsrutiner och är avsedd för uttestning av de olika filtren. I övrigt är de 2 programpaketen identiska. De olika kommandona och en kortfattad beskrivning av dessa framgår av listningen nedan.

INIT initialiserar estimeringen genom att användaren anger antal samplingsintervall som experimentet skall pågå, förstärkning på yttre interface till datorn samt initialvärde på estimatet  $\hat{x}(t)$ .

KALF läser in specificerat Kalmanfilter från dec-tape.

AXIS läser in specificerad plottningstäthet, maxvärden (endast på x- och y-axel samt ritar axlar på tv-skärmen i KEST) Dessutom finns möjlighet att plotta estimationsfelen förstärkta kring separat x-axel.

EXEC utför estimeringen, i fallet KEST plottas  
 estimatets och estimationsfelens värden i  
 7 punkter längs staven samt dessutom plottas  
 de 2 servosignalerna.

PRINT skriver ut Kalmanfiltret. I KEST utföres  
 även statistisk test av residualerna.

STOP avbryter exekveringen.

En mer detaljerad beskrivning av kommandostrukturen fram-  
 går av listningarna i appendix D (KEST) och E (SKEST).  
 SKEST kan köras endera separat eller tillsammans med DB,  
 medan KEST som är avsett för uttestning av Kalmanfiltren  
 endast kan köras separat.

#### 7.4 Disposition av de i Kalmanfiltren ingående matrisernas filhuvuden.

	FHEAD									
	1	2	3	4	5	6	7	8	9	10
FI	nx	nx	-	itick	ne	kxe	-	-	-	-
GA	nx	nu	-	itick	ne	kxe	-	-	-	-
TH	ny	nx	-	-	kth	-	-	-	-	-
AK	nx	ny	-	-	-	-	-	-	-	-

där

nx - antal tillstånd  
 nu - antal insignaler  
 ny - antal mätbara tillstånd (vid Kalmanfiltrering)  
 itick - samplingsintervall uttryckt i tick (1 tick = 20 ms)  
 ne - antal felsignaler  
 kth - specificerar vilka tillstånd som antages vara mät-  
 bara i det aktuella experimentet  
 kxe - specificerar vilka tillstånd som är mätbara

## 7.5 Disposition av filhuvudet i återkopplingsmatriserna.

	FHEAD									
	1	2	3	4	5	6	7	8	9	10
HRF..	nx	nu	nz	itick	-	-	-	-	-	-

där

nx - antal tillstånd

nu - antal insignaler

nz - blocklängd

itick - samplingsintervall uttryckt i tick (1 tick = 20 ms)

## 7.6 Disposition av commonblocket IOCOM i de tre program-paketen DB, SKEST och KEST.

## 1) DB, SKEST

ISYSA

18	ne	- antal felsignaler (från SKEST)
19	itickk	- samplingsintervall för Kalmanfilter
20	iper	- periodtid för SKEST relativt ALIO
22	iswd	- anger om EXEC-kommando givits i DB
23	iswk	- anger om EXEC-kommando givits i SKEST
24	iple	- anger när felsignaler skall plottas (från SKEST)

IEV (eventvariabler)

1	iev1	} styr ordningsföljden mellan DB's och SKEST's exekvering (se avsnitt 7.7)
3	iev3	
4	iev4	- synkroniserar DB till SKEST
5	iev5	- anger om stavens slutprofil skall ändras under pågående reglering

## USER

1 gain - förstärkning på yttre interface till datorn  
 3-6 fnamk - namn på Kalmanfilter  
 7-24 xe - estimerade värden på tillståndsvektorn  
 25-33 e - estimationsfelen

## 2) KEST

## ISYSA

18 nx - antal tillstånd  
 19 nu - antal insignaler  
 20 ny - antal mätbara tillstånd  
 21 ne - antal felsignaler  
 22 itick - samplingsintervall

## IEV (eventvariabler)

1 iev1 - synkroniserar beräkningen av  $\hat{x}(t+1)$  till ALIO  
 5 iev5 - ger möjlighet att ändra insignalernas värden  
 under pågående estimering

## USER

1 - förstärkning på yttre interface till datorn  
 7-24 - estimerade värden på tillståndsvektorn  
 25-33 - estimationsfelen

## Anmärkning

Då IOCOM disponeras olika i KEST respektive DB,SKEST  
 anbefalles nollställning av IOCOM mellan körningar av  
 de olika programpaketet.

### 7.7 Illustrering av eventvariablernas funktion vid samkörning av DB och SKEST.

De aktuella programmen har följande prioriteter (lägst prioriterat program exekverar först då flera program begär exekvering vid samma tidpunkt)

DB 100

SKEST 125

ALIO 150

Då programmet ALIO (som utför logisk och analog in/utmatning mellan datorn och datorns interface) exekverat färdigt sättes  $iev1 = 1$ .

Vi antar att EXEC-kommando givits i både DB och SKEST samt att DB kommit in i huvudloopen. Då ser programmen schematiskt ut som följer :

DB	SKEST
40 CALL WAITFR(IEV1) IEV1=0 beräkning av $u(t)$ IEV3=1 74 CALL WAIT IF(IEV3.EQ.0) GO TO 74 plottning av $u(t), y(t), e(t)$ GO TO 40	40 CALL WAITFR(IEV3) beräkning av $\hat{x}(t+1)$ IEV3=0 GO TO 40

Då händer i tidsföljd följande:

- 1) ALIO - mäter  $y(t)$ , sätter  $iev1 = 1$
- 2) DB - beräknar  $u(t) = L(t)(\hat{x}(t) - x_r) + u_r$
- 3) SKEST - beräknar  $\hat{x}(t+1) = \Phi \hat{x}(t) + \Gamma u(t) + K(y(t) - \theta \hat{x}(t))$
- 4) ALIO - ställer ut  $u(t)$  och gör därefter exit
- 5) DB - plottar  $u(t), y(t)$  och  $e(t)$ . Detta tillgår så att ett antal tecken skickas iväg till en buffert varpå ett systemprogram tar hand om själva utmatningen på skärmen. När hela plottningen är avklarad ställer sig DB i vänteloop på  $iev1$ , varpå händelse 1 åter inträffar osv.

## 7.8 Länkstruktur för programpaketet SKEST.

Eftersom SKEST är avsett att exekvera tillsammans med DB måste SKEST (som totalt kräver mindre utrymme än DB) exekvera i den näst största avdelningen (PUSER). För att få plats med programmen i denna avdelning måste programmen i SKEST länkas enligt följande

Resident code (finns i kärnminnet hela tiden)

SKEST,RLINE,#.ER (#.ER är en biblioteksrutin)

Links & structures (en länk överföres till kärnminnet vid anrop från endera huvudprogrammet eller från en annan länk)

L1=ERROR/FRMTOT

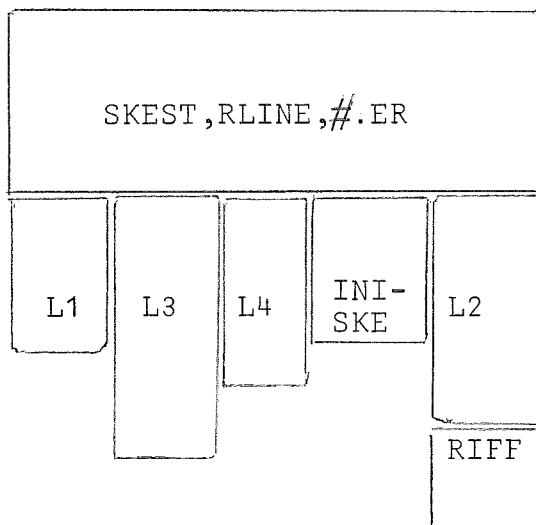
L2=KALFS/RDMTX

L3=PRTSKE/FRMTOT

L4=EXESKE/ISENRX

L1:L3:INISKE:L4:L2,RIFF

vilket kan åskådliggöras med följande figur



Anmärkning

RIFF måste ligga efter den största av de länkar som den användes utav. RIFF (som ombesörjer inläsning från teletype) användes av INISKE och KALFS och eftersom länken L2

APPENDIX A



## STEGSVAR HRA11 HRB11

T	Y1	Y3	Y5	Y7	E1	E3	E5	E7
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.264468	0.004868	0.000030	0.000000	-0.021845	-0.004412	-0.000030	-0.000000
20	0.408270	0.024447	0.000563	0.000007	0.000321	-0.011258	-0.000527	-0.000007
30	0.496188	0.054371	0.002565	0.000064	0.003565	-0.011395	-0.001821	-0.000061
40	0.555268	0.088591	0.006718	0.000277	0.003550	-0.008984	-0.003231	-0.000234
50	0.597924	0.123348	0.013165	0.000784	0.003044	-0.006487	-0.004189	-0.000533
60	0.630410	0.156761	0.021669	0.001712	0.002552	-0.004516	-0.004612	-0.000890
70	0.656155	0.188040	0.031824	0.003139	0.002151	-0.003078	-0.004630	-0.001220
80	0.677185	0.216956	0.043196	0.005090	0.001835	-0.002054	-0.004395	-0.001466
90	0.694774	0.243548	0.055391	0.007544	0.001584	-0.001327	-0.004023	-0.001608
100	0.709761	0.267967	0.068078	0.010446	0.001383	-0.000807	-0.003590	-0.001652
110	0.722729	0.290406	0.080997	0.013724	0.001221	-0.000430	-0.003139	-0.001616
120	0.734090	0.311061	0.093946	0.017297	0.001088	-0.000152	-0.002697	-0.001521
130	0.744150	0.330116	0.106772	0.021088	0.000978	0.000057	-0.002278	-0.001388
140	0.753137	0.347737	0.119362	0.025025	0.000887	0.000217	-0.001887	-0.001233
150	0.761228	0.364072	0.131636	0.029046	0.000811	0.000343	-0.001529	-0.001069
160	0.768559	0.379248	0.143535	0.033095	0.000747	0.000445	-0.001204	-0.000906
170	0.775239	0.393377	0.155020	0.037130	0.000694	0.000530	-0.000910	-0.000748
180	0.781358	0.406558	0.166068	0.041114	0.000650	0.000602	-0.000648	-0.000601
190	0.786985	0.418875	0.176665	0.045021	0.000613	0.000664	-0.000414	-0.000465
200	0.792180	0.430403	0.186804	0.048828	0.000583	0.000719	-0.000207	-0.000342
210	0.796992	0.441207	0.196489	0.052520	0.000559	0.000767	-0.000024	-0.000232
220	0.801461	0.451344	0.205723	0.056086	0.000539	0.000810	0.000137	-0.000134
230	0.805622	0.460866	0.214517	0.059518	0.000522	0.000848	0.000277	-0.000049
240	0.809504	0.469818	0.222881	0.062813	0.000509	0.000881	0.000398	0.000026
250	0.813134	0.478241	0.230830	0.065968	0.000498	0.000911	0.000503	0.000091
260	0.816531	0.486171	0.238378	0.068984	0.000489	0.000937	0.000593	0.000146
270	0.819717	0.493642	0.245541	0.071861	0.000482	0.000958	0.000670	0.000194
280	0.822707	0.500685	0.252335	0.074603	0.000476	0.000977	0.000735	0.000234
290	0.825516	0.507325	0.258775	0.077212	0.000470	0.000991	0.000789	0.000268
300	0.828159	0.513589	0.264877	0.079692	0.000465	0.001003	0.000834	0.000296

## STEGSVAR HRA11 HRB11

T	Y1	Y3	Y5	Y7	E1	E3	E5	E7
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
60	0.630410	0.156761	0.021669	0.001712	0.002552	-0.004516	-0.004612	-0.000890
120	0.734090	0.311061	0.093946	0.017297	0.001088	-0.000152	-0.002697	-0.001521
180	0.781358	0.406558	0.166068	0.041114	0.000650	0.000602	-0.000648	-0.000601
240	0.809504	0.469818	0.222881	0.062813	0.000509	0.000881	0.000398	0.000026
300	0.828159	0.513589	0.264877	0.079692	0.000465	0.001003	0.000834	0.000296
360	0.841088	0.544457	0.295256	0.092132	0.000435	0.001012	0.000956	0.000379
420	0.850213	0.566392	0.317056	0.101123	0.000396	0.000944	0.000927	0.000378
480	0.856700	0.582026	0.332653	0.107573	0.000348	0.000836	0.000831	0.000342
540	0.861324	0.593181	0.343798	0.112186	0.000296	0.000714	0.000712	0.000294
600	0.864623	0.601145	0.351759	0.115483	0.000246	0.000593	0.000593	0.000245
660	0.866979	0.606831	0.357445	0.117838	0.000200	0.000483	0.000483	0.000200
720	0.868661	0.610892	0.361505	0.119519	0.000160	0.000387	0.000387	0.000160
780	0.869862	0.613791	0.364404	0.120720	0.000127	0.000306	0.000306	0.000127
840	0.870719	0.615861	0.366474	0.121578	0.000099	0.000240	0.000240	0.000099
900	0.871331	0.617340	0.367953	0.122190	0.000077	0.000187	0.000187	0.000077
960	0.871769	0.618395	0.369009	0.122627	0.000059	0.000144	0.000144	0.000059
020	0.872081	0.619149	0.369762	0.122940	0.000046	0.000110	0.000110	0.000046
080	0.872304	0.619688	0.370301	0.123163	0.000035	0.000084	0.000084	0.000035
140	0.872463	0.620072	0.370685	0.123322	0.000026	0.000064	0.000064	0.000026
200	0.872577	0.620346	0.370960	0.123436	0.000020	0.000048	0.000048	0.000020
260	0.872658	0.620542	0.371156	0.123517	0.000015	0.000036	0.000036	0.000015
320	0.872716	0.620682	0.371296	0.123575	0.000011	0.000027	0.000027	0.000011
380	0.872757	0.620782	0.371395	0.123616	0.000008	0.000020	0.000020	0.000008
440	0.872787	0.620854	0.371467	0.123646	0.000006	0.000015	0.000015	0.000006
500	0.872808	0.620905	0.371518	0.123667	0.000004	0.000011	0.000011	0.000004
560	0.872823	0.620941	0.371554	0.123682	0.000003	0.000008	0.000008	0.000003
620	0.872834	0.620967	0.371580	0.123693	0.000002	0.000006	0.000006	0.000002
680	0.872842	0.620986	0.371599	0.123700	0.000002	0.000004	0.000004	0.000002
740	0.872847	0.620999	0.371612	0.123706	0.000001	0.000003	0.000003	0.000001
800	0.872851	0.621008	0.371621	0.123710	0.000001	0.000002	0.000002	0.000001

## STEGSVAR HRA21 HRB21

T	Y1	Y3	Y5	Y7	E1	E3	E5	E7
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
60	0.628990	0.153160	0.017964	0.000987	0.003972	-0.000915	-0.000907	-0.000165
120	0.733978	0.310394	0.091695	0.016176	0.001200	0.000515	-0.000446	-0.000400
180	0.781502	0.406761	0.165465	0.040613	0.000506	0.000399	-0.000045	-0.000099
240	0.809747	0.470419	0.223164	0.062780	0.000267	0.000281	0.000115	0.000059
300	0.828455	0.514375	0.265550	0.079887	0.000168	0.000217	0.000161	0.000101
360	0.841403	0.545291	0.296052	0.092412	0.000120	0.000178	0.000160	0.000100
420	0.850518	0.567189	0.317841	0.101416	0.000091	0.000147	0.000142	0.000085
480	0.856977	0.582741	0.333365	0.107846	0.000071	0.000121	0.000119	0.000069
540	0.861565	0.593797	0.344413	0.112426	0.000055	0.000098	0.000097	0.000054
600	0.864826	0.601660	0.352274	0.115686	0.000043	0.000078	0.000078	0.000042
660	0.867146	0.607252	0.357866	0.118005	0.000033	0.000062	0.000062	0.000033
720	0.868796	0.611230	0.361843	0.119654	0.000025	0.000048	0.000048	0.000025
780	0.869969	0.614060	0.364673	0.120828	0.000019	0.000038	0.000038	0.000019
840	0.870804	0.616072	0.366686	0.121663	0.000014	0.000029	0.000029	0.000014
900	0.871398	0.617504	0.368117	0.122256	0.000011	0.000022	0.000022	0.000011
960	0.871820	0.618522	0.369136	0.122679	0.000008	0.000017	0.000017	0.000008
1020	0.872120	0.619247	0.369860	0.122979	0.000006	0.000013	0.000013	0.000006
1080	0.872334	0.619762	0.370375	0.123193	0.000005	0.000010	0.000010	0.000005
1140	0.872486	0.620129	0.370742	0.123345	0.000003	0.000007	0.000007	0.000003
1200	0.872594	0.620389	0.371002	0.123453	0.000002	0.000005	0.000005	0.000002
1260	0.872671	0.620575	0.371188	0.123530	0.000002	0.000004	0.000004	0.000002
1320	0.872726	0.620707	0.371320	0.123585	0.000001	0.000003	0.000003	0.000001
1380	0.872765	0.620800	0.371414	0.123624	0.000001	0.000002	0.000002	0.000001
1440	0.872793	0.620867	0.371480	0.123651	0.000001	0.000002	0.000002	0.000001
1500	0.872812	0.620915	0.371528	0.123671	0.000000	0.000001	0.000001	0.000000
1560	0.872826	0.620948	0.371562	0.123685	0.000000	0.000001	0.000001	0.000000
1620	0.872836	0.620972	0.371586	0.123695	0.000000	0.000001	0.000001	0.000000
1680	0.872843	0.620990	0.371603	0.123702	0.000000	0.000000	0.000001	0.000000
1740	0.872848	0.621002	0.371615	0.123707	0.000000	0.000000	0.000000	-0.000000
1800	0.872852	0.621010	0.371624	0.123711	-0.000000	0.000000	0.000000	-0.000000

STEGSVAR HRA21 HRB21

T	Y1	Y3	Y5	Y7	E1	E3	E5	E7
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.262409	0.000235	-0.000048	0.000000	-0.019786	0.000222	0.000048	-0.000000
20	0.404990	0.017191	-0.000201	-0.000006	0.003601	-0.004001	0.000237	0.000005
30	0.493028	0.047342	0.000708	-0.000023	0.006725	-0.004366	0.000035	0.000025
40	0.552699	0.082723	0.003892	0.000025	0.006119	-0.003116	-0.000405	0.000018
50	0.595980	0.118707	0.009723	0.000303	0.004987	-0.001846	-0.000748	-0.000051
60	0.628990	0.153160	0.017964	0.000987	0.003972	-0.000915	-0.000907	-0.000165
70	0.655139	0.185265	0.028120	0.002203	0.003168	-0.000303	-0.000926	-0.000284
80	0.676471	0.214825	0.039665	0.004002	0.002549	0.000077	-0.000864	-0.000377
90	0.694282	0.241917	0.052132	0.006368	0.002076	0.000303	-0.000765	-0.000432
100	0.709435	0.266731	0.065143	0.009243	0.001710	0.000429	-0.000655	-0.000449
110	0.722525	0.289485	0.078405	0.012543	0.001425	0.000492	-0.000546	-0.000435
120	0.733978	0.310394	0.091695	0.016176	0.001200	0.000515	-0.000446	-0.000400
130	0.744109	0.329659	0.104850	0.020053	0.001020	0.000514	-0.000356	-0.000353
140	0.753150	0.347455	0.117751	0.024091	0.000874	0.000499	-0.000276	-0.000299
150	0.761283	0.363938	0.130312	0.028220	0.000755	0.000477	-0.000205	-0.000244
160	0.768649	0.379241	0.142475	0.032381	0.000656	0.000451	-0.000144	-0.000191
170	0.775359	0.393482	0.154201	0.036525	0.000575	0.000425	-0.000091	-0.000143
180	0.781502	0.406761	0.165465	0.040613	0.000506	0.000399	-0.000045	-0.000099
190	0.787150	0.419165	0.176257	0.044617	0.000448	0.000374	-0.000006	-0.000061
200	0.792364	0.430770	0.186570	0.048514	0.000400	0.000352	0.000028	-0.000027
210	0.797193	0.441642	0.196409	0.052288	0.000358	0.000331	0.000056	0.000001
220	0.801677	0.451841	0.205780	0.055927	0.000323	0.000312	0.000080	0.000024
230	0.805852	0.461418	0.214694	0.059426	0.000293	0.000295	0.000099	0.000043
240	0.809747	0.470419	0.223164	0.062780	0.000267	0.000280	0.000115	0.000059
250	0.813388	0.478885	0.231205	0.065987	0.000244	0.000267	0.000129	0.000072
260	0.816796	0.486853	0.238833	0.069049	0.000225	0.000255	0.000139	0.000082
270	0.819991	0.494357	0.246064	0.071966	0.000208	0.000244	0.000147	0.000089
280	0.822990	0.501427	0.252916	0.074742	0.000193	0.000234	0.000154	0.000095
290	0.825807	0.508091	0.259406	0.077381	0.000180	0.000225	0.000158	0.000099
300	0.828455	0.514375	0.265550	0.079887	0.000168	0.000217	0.000161	0.000101

STEGSVAR HRA41 HRB41

T	Y1	Y3	Y5	Y7	E1	E3	E5	E7
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
60	0.628984	0.153041	0.017941	0.001031	0.003978	-0.000796	0.000000	-0.000209
120	0.733956	0.310344	0.091610	0.016156	0.001223	0.000565	-0.000361	-0.000360
180	0.781489	0.406736	0.165421	0.040592	0.000519	0.000424	-0.000001	-0.000078
240	0.809741	0.470408	0.223145	0.062769	0.000273	0.000292	0.000134	0.000070
300	0.828453	0.514372	0.265545	0.079882	0.000171	0.000220	0.000167	0.000106
360	0.841403	0.545293	0.296053	0.092412	0.000121	0.000175	0.000158	0.000100
420	0.850519	0.567194	0.317846	0.101417	0.000090	0.000142	0.000137	0.000084
480	0.856979	0.582747	0.333370	0.107848	0.000069	0.000115	0.000113	0.000067
540	0.861567	0.593803	0.344419	0.112428	0.000053	0.000092	0.000091	0.000052
600	0.864828	0.601665	0.352279	0.115688	0.000041	0.000073	0.000073	0.000041
660	0.867148	0.607257	0.357870	0.118006	0.000031	0.000057	0.000057	0.000031
720	0.868797	0.611234	0.361847	0.119656	0.000024	0.000044	0.000044	0.000024
780	0.869970	0.614063	0.364676	0.120829	0.000018	0.000034	0.000034	0.000018
840	0.870805	0.616075	0.366688	0.121664	0.000013	0.000026	0.000026	0.000013
900	0.871399	0.617506	0.368119	0.122257	0.000010	0.000020	0.000020	0.000010
960	0.871821	0.618524	0.369137	0.122679	0.000008	0.000015	0.000015	0.000007
1020	0.872121	0.619248	0.369861	0.122980	0.000006	0.000012	0.000012	0.000006
1080	0.872335	0.619763	0.370376	0.123193	0.000004	0.000009	0.000009	0.000004
1140	0.872487	0.620129	0.370742	0.123345	0.000003	0.000007	0.000007	0.000003
1200	0.872595	0.620390	0.371003	0.123453	0.000002	0.000005	0.000005	0.000002
1260	0.872671	0.620575	0.371188	0.123530	0.000002	0.000004	0.000004	0.000002
1320	0.872726	0.620707	0.371320	0.123585	0.000001	0.000003	0.000003	0.000001
1380	0.872765	0.620801	0.371414	0.123624	0.000001	0.000002	0.000002	0.000001
1440	0.872793	0.620867	0.371481	0.123651	0.000001	0.000001	0.000001	0.000001
1500	0.872812	0.620915	0.371528	0.123671	0.000000	0.000001	0.000001	0.000000
1560	0.872826	0.620949	0.371562	0.123685	0.000000	0.000001	0.000001	0.000000
1620	0.872836	0.620973	0.371586	0.123695	0.000000	0.000001	0.000001	0.000000
1680	0.872843	0.620990	0.371603	0.123702	0.000000	0.000000	0.000000	0.000000
1740	0.872848	0.621002	0.371615	0.123707	-0.000000	0.000000	0.000000	-0.000000
1800	0.872852	0.621010	0.371624	0.123711	-0.000000	0.000000	0.000000	-0.000000

## STEGSVAR HRA41 HRB41

T	Y1	Y3	Y5	Y7	E1	E3	E5	E7
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.262441	0.000847	0.000064	0.000000	-0.019618	-0.000391	-0.000064	-0.000001
20	0.405062	0.017277	0.000078	0.000004	0.003529	-0.004088	-0.000042	-0.000004
30	0.493095	0.047231	0.000969	0.000007	0.006659	-0.004255	-0.000225	-0.000005
40	0.552739	0.082575	0.004046	0.000072	0.006079	-0.002967	-0.000559	-0.000029
50	0.595994	0.118568	0.009773	0.000354	0.004974	-0.001707	-0.000798	-0.000102
60	0.628984	0.153041	0.017941	0.001031	0.003978	-0.000796	-0.000884	-0.000209
70	0.655122	0.185164	0.028055	0.002234	0.003185	-0.000202	-0.000860	-0.000315
80	0.676448	0.214738	0.039578	0.004018	0.002572	0.000163	-0.000777	-0.000394
90	0.694257	0.241843	0.052037	0.006371	0.002101	0.000377	-0.000670	-0.000435
100	0.709409	0.266667	0.065048	0.009235	0.001736	0.000493	-0.000560	-0.000441
110	0.722500	0.289428	0.078314	0.012528	0.001450	0.000548	-0.000456	-0.000420
120	0.733956	0.310344	0.091610	0.016156	0.001223	0.000565	-0.000361	-0.000380
130	0.744088	0.329614	0.104772	0.020030	0.001041	0.000558	-0.000278	-0.000329
140	0.753131	0.347415	0.117680	0.024067	0.000893	0.000539	-0.000205	-0.000274
150	0.761266	0.363902	0.130249	0.028196	0.000772	0.000512	-0.000142	-0.000219
160	0.768634	0.379210	0.142418	0.032357	0.000672	0.000483	-0.000088	-0.000167
170	0.775345	0.393454	0.154151	0.036502	0.000588	0.000453	-0.000041	-0.000120
180	0.781489	0.406736	0.165421	0.040592	0.000519	0.000424	-0.000001	-0.000078
190	0.787139	0.419143	0.176218	0.044597	0.000460	0.000396	0.000033	-0.000041
200	0.792354	0.430750	0.186536	0.048496	0.000410	0.000371	0.000062	-0.000010
210	0.797184	0.441625	0.196379	0.052272	0.000367	0.000348	0.000086	0.000016
220	0.801669	0.451826	0.205754	0.055913	0.000331	0.000327	0.000105	0.000038
230	0.805845	0.461405	0.214672	0.059414	0.000300	0.000309	0.000121	0.000056
240	0.809741	0.470408	0.223145	0.062769	0.000273	0.000292	0.000134	0.000070
250	0.813382	0.478875	0.231189	0.065978	0.000250	0.000277	0.000144	0.000081
260	0.816791	0.486845	0.238819	0.069040	0.000229	0.000263	0.000152	0.000090
270	0.819987	0.494350	0.246053	0.071959	0.000212	0.000250	0.000158	0.000096
280	0.822986	0.501422	0.252907	0.074736	0.000196	0.000239	0.000162	0.000101
290	0.825804	0.508087	0.259399	0.077376	0.000183	0.000229	0.000165	0.000104
300	0.828453	0.514372	0.265545	0.079882	0.000171	0.000220	0.000166	0.000106

STEGSVAR HRA22 HRB22

T	Y1	Y3	Y5	Y7	E1	E3	E5	E7
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
60	0.632628	0.152254	0.017134	0.000831	0.000334	-0.000009	-0.000077	-0.000009
120	0.735103	0.310856	0.091268	0.015799	0.000076	0.000053	-0.000019	-0.000023
180	0.781978	0.407127	0.165415	0.040518	0.000030	0.000033	0.000006	-0.000004
240	0.809998	0.470678	0.223267	0.062835	0.000015	0.000021	0.000012	0.000004
300	0.828614	0.514577	0.265699	0.079982	0.000009	0.000015	0.000012	0.000006
360	0.841517	0.545457	0.296201	0.092507	0.000007	0.000012	0.000011	0.000005
420	0.850604	0.567327	0.317974	0.101497	0.000005	0.000009	0.000009	0.000004
480	0.857044	0.582855	0.333477	0.107911	0.000004	0.000007	0.000007	0.000003
540	0.861617	0.593889	0.344505	0.112478	0.000003	0.000006	0.000006	0.000003
600	0.864867	0.601734	0.352347	0.115726	0.000002	0.000005	0.000005	0.000002
660	0.867177	0.607311	0.357924	0.118036	0.000002	0.000004	0.000004	0.000002
720	0.868819	0.611276	0.361889	0.119678	0.000001	0.000003	0.000003	0.000001
780	0.869987	0.614095	0.364708	0.120846	0.000001	0.000002	0.000002	0.000001
840	0.870818	0.616099	0.366713	0.121676	0.000001	0.000002	0.000002	0.000001
900	0.871408	0.617525	0.368138	0.122267	0.000001	0.000002	0.000002	0.000000
960	0.871828	0.618538	0.369151	0.122687	0.000001	0.000001	0.000001	0.000000
1020	0.872126	0.619258	0.369872	0.122985	0.000000	0.000001	0.000001	0.000000
1080	0.872338	0.619771	0.370384	0.123197	0.000000	0.000001	0.000001	0.000000
1140	0.872489	0.620135	0.370748	0.123348	0.000000	0.000001	0.000001	0.000000
1200	0.872596	0.620394	0.371007	0.123455	0.000000	0.000001	0.000001	0.000000
1260	0.872673	0.620578	0.371191	0.123532	0.000000	0.000001	0.000001	0.000000
1320	0.872727	0.620709	0.371322	0.123586	0.000000	0.000001	0.000001	0.000000
1380	0.872766	0.620802	0.371415	0.123624	0.000000	0.000001	0.000001	0.000000
1440	0.872793	0.620868	0.371482	0.123652	0.000000	0.000001	0.000001	0.000000
1500	0.872812	0.620915	0.371529	0.123671	0.000000	0.000001	0.000001	0.000000
1560	0.872826	0.620949	0.371562	0.123685	0.000000	0.000001	0.000001	0.000000
1620	0.872836	0.620973	0.371586	0.123695	0.000000	0.000001	0.000001	0.000000
1680	0.872843	0.620989	0.371603	0.123702	0.000000	0.000001	0.000001	0.000000
1740	0.872848	0.621002	0.371615	0.123707	0.000000	0.000001	0.000001	0.000000
1800	0.872852	0.621010	0.371623	0.123711	0.000000	0.000001	0.000000	0.000000

## STEGSVAR HRA22 HRB22

T	Y1	Y3	Y5	Y7	E1	E3	E5	E7
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.242918	0.000401	-0.000000	-0.000000	-0.000295	0.000055	0.000000	-0.000000
20	0.406965	0.013656	0.000024	-0.000000	0.001626	-0.000466	0.000012	-0.000000
30	0.498656	0.043375	0.000751	0.000001	0.001097	-0.000399	-0.000007	0.000001
40	0.558112	0.079819	0.003535	0.000041	0.000706	-0.000212	-0.000048	0.000002
50	0.600493	0.116943	0.009048	0.000253	0.000474	-0.000082	-0.000072	-0.000002
60	0.632628	0.152254	0.017134	0.000831	0.000334	-0.000009	-0.000077	-0.000009
70	0.658062	0.184933	0.027265	0.001936	0.000244	0.000029	-0.000070	-0.000017
80	0.678835	0.214854	0.038860	0.003647	0.000185	0.000047	-0.000059	-0.000023
90	0.696214	0.242165	0.051414	0.005962	0.000143	0.000055	-0.000047	-0.000026
100	0.711031	0.267103	0.064525	0.008821	0.000114	0.000057	-0.000036	-0.000027
110	0.723858	0.289921	0.077885	0.012133	0.000092	0.000055	-0.000027	-0.000025
120	0.735103	0.310856	0.091268	0.015799	0.000076	0.000053	-0.000019	-0.000023
130	0.745065	0.330124	0.104507	0.019720	0.000063	0.000049	-0.000013	-0.000019
140	0.753971	0.347909	0.117482	0.023808	0.000053	0.000045	-0.000007	-0.000016
150	0.761993	0.364373	0.130110	0.027989	0.000045	0.000042	-0.000003	-0.000012
160	0.769267	0.379654	0.142330	0.032199	0.000039	0.000038	0.000000	-0.000009
170	0.775900	0.393872	0.154107	0.036388	0.000034	0.000035	0.000003	-0.000006
180	0.781978	0.407127	0.165415	0.040518	0.000030	0.000032	0.000006	-0.000004
190	0.787573	0.419509	0.176243	0.044558	0.000026	0.000030	0.000007	-0.000002
200	0.792741	0.431094	0.186589	0.048486	0.000023	0.000028	0.000009	-0.000000
210	0.797530	0.441948	0.196455	0.052287	0.000021	0.000026	0.000010	0.000001
220	0.801982	0.452130	0.205849	0.055949	0.000018	0.000024	0.000011	0.000003
230	0.806128	0.461691	0.214782	0.059466	0.000017	0.000022	0.000011	0.000003
240	0.809999	0.470678	0.223268	0.062835	0.000015	0.000021	0.000012	0.000004
250	0.813618	0.479132	0.231321	0.066054	0.000014	0.000020	0.000012	0.000005
260	0.817008	0.487089	0.238960	0.069125	0.000013	0.000019	0.000012	0.000005
270	0.820187	0.494583	0.246199	0.072050	0.000012	0.000018	0.000012	0.000005
280	0.823172	0.501645	0.253057	0.074831	0.000011	0.000017	0.000012	0.000006
290	0.825977	0.508301	0.259552	0.077474	0.000010	0.000016	0.000012	0.000006
300	0.828614	0.514577	0.265699	0.079982	0.000009	0.000015	0.000012	0.000006



## STEGSVAR HRA42 HRB42

T	Y1	Y3	Y5	Y7	E1	E3	E5	E7
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.242920	0.000576	0.000000	0.000000	-0.000297	-0.000119	-0.000000	-0.000000
20	0.406911	0.013679	0.000044	0.000000	0.001680	-0.000489	-0.000008	-0.000000
30	0.498622	0.043335	0.000783	0.000003	0.001131	-0.000360	-0.000039	-0.000001
40	0.558093	0.079777	0.003558	0.000046	0.000725	-0.000170	-0.000071	-0.000003
50	0.600483	0.116910	0.009059	0.000260	0.000485	-0.000049	-0.000083	-0.000009
60	0.632623	0.152230	0.017135	0.000838	0.000339	0.000016	-0.000078	-0.000016
70	0.658059	0.184915	0.027260	0.001941	0.000247	0.000047	-0.000066	-0.000022
80	0.678834	0.214841	0.038853	0.003650	0.000186	0.000060	-0.000052	-0.000026
90	0.696214	0.242156	0.051405	0.005963	0.000144	0.000064	-0.000038	-0.000027
100	0.711031	0.267096	0.064515	0.008820	0.000114	0.000064	-0.000027	-0.000026
110	0.723858	0.289916	0.077876	0.012131	0.000092	0.000061	-0.000018	-0.000023
120	0.735103	0.310852	0.091259	0.015796	0.000075	0.000057	-0.000010	-0.000020
130	0.745066	0.330120	0.104499	0.019717	0.000062	0.000052	-0.000005	-0.000016
140	0.753971	0.347906	0.117475	0.023805	0.000053	0.000048	0.000000	-0.000013
150	0.761993	0.364371	0.130103	0.027986	0.000045	0.000044	0.000004	-0.000009
160	0.769267	0.379653	0.142324	0.032196	0.000038	0.000040	0.000006	-0.000006
170	0.775900	0.393870	0.154101	0.036385	0.000033	0.000037	0.000009	-0.000004
180	0.781979	0.407126	0.165410	0.040515	0.000029	0.000034	0.000010	-0.000001
190	0.787573	0.419508	0.176239	0.044556	0.000026	0.000031	0.000012	0.000001
200	0.792741	0.431093	0.186585	0.048484	0.000023	0.000029	0.000012	0.000002
210	0.797531	0.441947	0.196452	0.052285	0.000020	0.000027	0.000013	0.000003
220	0.801982	0.452129	0.205846	0.055947	0.000018	0.000025	0.000014	0.000004
230	0.806128	0.461691	0.214779	0.059465	0.000016	0.000023	0.000014	0.000005
240	0.809999	0.470678	0.223266	0.062834	0.000015	0.000021	0.000014	0.000005
250	0.813618	0.479132	0.231320	0.066053	0.000014	0.000020	0.000014	0.000006
260	0.817008	0.487089	0.238958	0.069124	0.000012	0.000019	0.000014	0.000006
270	0.820187	0.494583	0.246198	0.072049	0.000011	0.000018	0.000014	0.000006
280	0.823172	0.501644	0.253056	0.074830	0.000011	0.000017	0.000013	0.000006
290	0.825977	0.508301	0.259551	0.077473	0.000010	0.000016	0.000013	0.000006
300	0.828664	0.514577	0.265698	0.079982	0.000009	0.000015	0.000013	0.000006

STEGSVAR HRA42 HRB42

T	Y1	Y3	Y5	Y7	E1	E3	E5	E7
0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
60	0.632623	0.152230	0.017135	0.000838	0.000339	0.000016	0.000000	-0.000016
120	0.735103	0.310852	0.091259	0.015796	0.000075	0.000057	-0.000010	-0.000020
180	0.781979	0.407126	0.165410	0.040515	0.000029	0.000034	0.000010	-0.000001
240	0.809999	0.470678	0.223265	0.062834	0.000015	0.000022	0.000014	0.000005
300	0.828614	0.514577	0.265698	0.079982	0.000009	0.000015	0.000013	0.000006
360	0.841517	0.545457	0.296201	0.092506	0.000006	0.000011	0.000011	0.000005
420	0.850605	0.567327	0.317974	0.101497	0.000005	0.000009	0.000008	0.000004
480	0.857044	0.582855	0.333477	0.107912	0.000004	0.000007	0.000007	0.000003
540	0.861617	0.593890	0.344505	0.112478	0.000003	0.000005	0.000005	0.000002
600	0.864867	0.601734	0.352348	0.115726	0.000002	0.000004	0.000004	0.000002
660	0.867177	0.607311	0.357924	0.118036	0.000002	0.000003	0.000003	0.000001
720	0.868820	0.611276	0.361889	0.119678	0.000001	0.000003	0.000003	0.000001
780	0.869987	0.614095	0.364708	0.120846	0.000001	0.000002	0.000002	0.000001
840	0.870818	0.616100	0.366713	0.121676	0.000001	0.000002	0.000002	0.000001
900	0.871408	0.617525	0.368138	0.122267	0.000001	0.000001	0.000001	0.000000
960	0.871828	0.618538	0.369151	0.122687	0.000001	0.000001	0.000001	0.000000
1020	0.872126	0.619259	0.369872	0.122985	0.000000	0.000001	0.000001	0.000000
1080	0.872338	0.619771	0.370384	0.123197	0.000000	0.000001	0.000001	0.000000
1140	0.872489	0.620135	0.370748	0.123348	0.000000	0.000001	0.000001	0.000000
1200	0.872596	0.620394	0.371007	0.123455	0.000000	0.000001	0.000001	0.000000
1260	0.872673	0.620578	0.371191	0.123532	0.000000	0.000001	0.000001	0.000000
1320	0.872727	0.620709	0.371322	0.123586	0.000000	0.000001	0.000001	0.000000
1380	0.872766	0.620802	0.371415	0.123624	0.000000	0.000001	0.000001	0.000000
1440	0.872793	0.620868	0.371482	0.123652	0.000000	0.000001	0.000001	0.000000
1500	0.872812	0.620915	0.371529	0.123671	0.000000	0.000001	0.000001	0.000000
1560	0.872826	0.620949	0.371562	0.123685	0.000000	0.000001	0.000001	0.000000
1620	0.872836	0.620973	0.371586	0.123695	0.000000	0.000001	0.000001	0.000000
1680	0.872843	0.620989	0.371603	0.123702	0.000000	0.000001	0.000001	0.000000
1740	0.872848	0.621002	0.371615	0.123707	0.000000	0.000001	0.000001	0.000000
1800	0.872852	0.621010	0.371623	0.123711	0.000000	0.000001	0.000000	0.000000

APPENDIX B

C PROGRAM RODPAC

C  
C  
C  
C  
C  
C INTERACTIVE PROGRAM FOR GENERATING MODELS AND  
C ANALYSING/COMPUTING KALMAN FILTERS OF A  
C DIFFUSION PROCESS  
C  
C  
C  
C  
C  
C  
C  
C

C DESCRIPTION OF COMMAND STRUCTURE:

C \*\*\*\*\*

C RODI NX NU HNM1 HNM2 (I=1,2,3,4,5,6,7,8,9)

C NX - NUMBER OF STATES (MAX 20)  
C NU - NUMBER OF INPUTS (MAX 2)  
C HNM1 - NAME OF CONTINUOUS 'A'-MATRIX  
C HNM2 - NAME OF CONTINUOUS 'B'-MATRIX  
C  
C

C SAMP HNM1 HNM2 TSAMP HNM3 HNM4

C HNM1 - NAME OF CONTINUOUS 'A'-MATRIX  
C HNM2 - NAME OF CONTINUOUS 'B'-MATRIX  
C TSAMP - SAMPLINGINTERVAL (SEC)  
C HNM3 - NAME OF DISCRETE 'FI'-MATRIX  
C HNM4 - NAME OF DISCRETE 'GAMMA'-MATRIX  
C  
C

C THETA HNM NY NX I1 I2 ,,,INY

C HNM - NAME OF DISCRETE 'THETA'-MATRIX  
C NY - NUMBER OF MEASURABLE STATES (MAX 9)  
C NX - NUMBER OF STATES (MAX 20)  
C I1,I2 ,,,INY- SPECIFIES MEASURABLE STATES  
C  
C

C CMTX HNM NX D1 D2 ,,,DNX

C HNM - NAME OF DIAGONAL COVARIANCE MATRIX  
C NX - ORDER OF COVARIANCE MATRIX (MAX 20)  
C D1 D2 ,,,DNX- VALUE OF EACH DIAGONAL ELEMENT  
C  
C

C TRANS CSYST CCOV TSAMP DCOV

C CSYST - NAME OF CONTINUOUS SYSTEM (A NUL C NUL)  
C CCOV - NAME OF CONTINUOUS COVARIANCE MATRICES  
C (NUL R1 R12/NUL R2)  
C TSAMP - SAMPLINGINTERVAL (SEC)  
C DCOV - NAME OF DISCRETE COVARIANCE MATRICES  
C (NUL R1D R12D R2D)  
C



## SUBROUTINES REQUIRED

ROD

ROD1

ROD2

ROD3

ROD4

ROD5

ROD6

ROD7

ROD8

ROD9

SAMP

COSA1

EXPAN1

THETA

CMTX

TRANS

INRIC1

RESYS

MATRE

FILES

STRANS

KALFI

INRIC1

RESYS

MATRE

FILES

DRIC1

EIGP1

SCAL

HESQR

REALVE

COMPVE

EIGEN

EIGP1

SCAL

HESQR

REALVE

COMPVE

SYST

ZEROM

EXTEN

MATOP

DEFI

```

DIMENSION COM(22)
C
COMMON /CMD/ IP,IND,BUFF(16),ITC
COMMON /DEF/ NMAX,EPS
COMMON /DEFHR/ RL,TD,THL,T1,T2,G1,G2,ZETA1,ZETA2,
*W01,W02
C
DATA COM /4HR0D1,4HR0D2,4HR0D3,4HR0D4,4HR0D5,4HR0D6,
*4HR0D7,4HR0D8,4HR0D9,4HCNTX,5HTHETA,4HSAMP,5HTRANS,
*5HKALFI,5HEIGEN,4HSYST,5HZEROM,5HEXTEN,5HMATOP,
*4HDEFI,4HSTOP,1H,/
C
C INITIATE
C
WRITE(9,500)
ITC=1
NMAX=200
EPS=1.0E-04
RL=45.
TD=1.159
THL=0.36E-04
T1=1.5
T2=1.5
G1=0.45
G2=0.45
ZETA1=1.3
ZETA2=1.3
W01=0.19
W02=0.19
C
C READ A COMMAND FROM TT1
C
2 IERR=0
CALL RLINE(9,8,BUFF,1)
IP=1
C
CALL RIFF(IP,RRES,IRES,IND)
IF(IND.EQ.1.OR.IND.EQ.6) GO TO 4
IF(IND-7) 8,2,8
4 IF(RRES.EQ.COM(22)) RRES=COM(ITC)
DO 6 ITC=1,21
IF(RRES.EQ.COM(ITC)) GO TO(10,10,10,10,10,10,10,10,10,
*20,30,40,50,60,70,80,90,100,110,120,130),ITC
6 CONTINUE
C
8 IERR=1
GO TO 200
C
C ROD
C
10 CALL ROD(IERR)
IF(IERR.EQ.0) GO TO 2
GO TO 200
C
C CMTX
C
20 CALL CMTX(IERR)
IF(IERR.EQ.0) GO TO 2
GO TO 200
C
C THETA
C
30 CALL THETA(IERR)
IF(IERR.EQ.0) GO TO 2

```

```
GO TO 200
C
C   SAMP
C
40  CALL SAMP(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 200
C
C   TRANS
C
50  CALL TRANS(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 200
C
C   KALFI
C
60  CALL KALFI(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 200
C
C   EIGEN
C
70  CALL EIGEN(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 200
C
C   SYST
C
80  CALL SYST(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 200
C
C   ZEROM
C
90  CALL ZEROM(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 200
C
C   EXTEN
C
100 CALL EXTEN(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 200
C
C   MATOP
C
110 CALL MATOP(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 200
C
C   DEFI
C
120 CALL DEFI(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 200
C
C   STOP
C
130 CONTINUE
    STOP
C
C   ERROR CHECK
C
200 NERR=11
```



```
DO 210 J=1,NERR  
IF(J.EQ.IERR) GO TO(301,302,303,304,305,306,  
*307,308,309,310,311),J
```

```
210 CONTINUE
```

```
C
```

```
301 WRITE(9,401)
```

```
GO TO 2
```

```
302 WRITE(9,402)
```

```
GO TO 2
```

```
303 WRITE(9,403)
```

```
GO TO 2
```

```
304 WRITE(9,404)
```

```
GO TO 2
```

```
305 WRITE(9,405)
```

```
GO TO 2
```

```
306 WRITE(9,406)
```

```
GO TO 2
```

```
307 WRITE(9,407)
```

```
GO TO 2
```

```
308 WRITE(9,408)
```

```
GO TO 2
```

```
309 WRITE(9,409)
```

```
GO TO 2
```

```
310 WRITE(9,410)
```

```
GO TO 2
```

```
311 WRITE(9,411)
```

```
GO TO 2
```

```
C
```

```
401 FORMAT(16H ILLEGAL COMMAND)
```

```
402 FORMAT(18H TOO FEW ARGUMENTS)
```

```
403 FORMAT(19H TOO MANY ARGUMENTS)
```

```
404 FORMAT(12H BAD INTEGER)
```

```
405 FORMAT(11H BAD NUMBER)
```

```
406 FORMAT(14H BAD FILE NAME)
```

```
407 FORMAT(28H BAD SYSTEM DESCRIPTION NAME)
```

```
408 FORMAT(15H FILE NOT FOUND)
```

```
409 FORMAT(22H EIGENVALUES NOT FOUND)
```

```
410 FORMAT(11H S SINGULAR)
```

```
411 FORMAT(19H BAD PARAMETER NAME)
```

```
500 FORMAT(7H RODPAC)
```

```
C
```

```
END
```

SUBROUTINE ROD(IERR)

MAIN SUBROUTINE FOR ROD1,ROD2,ROD3,ROD4,  
ROD5,ROD6,ROD7,ROD8,ROD9

AUTHOR: TORSTEN CROMNOW 1974-06-15

SUBROUTINES REQUIRED

ROD1  
ROD2  
ROD3  
ROD4  
ROD5  
ROD6  
ROD7  
ROD8  
ROD9

DIMENSION A(20,20),B(20,2),KODE(10),  
\*HNM1(2),HNM2(2),SLASK(20,20)

COMMON /CMD/ IP,IND,BUFF(16),ITC  
COMMON SLASK,A,B

DATA HNM1/5H ,4H BIN/  
DATA HNM2/5H ,4H BIN/

READ ARGUMENTS

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND-3) 80,12,80  
12 IF(IRES.LT.(ITC+1)) GO TO 84  
IF(IRES.GT.20) GO TO 84  
NX=IRES

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND-3) 80,14,80  
14 IF(IRES.LE.0.OR.IRES.GT.2) GO TO 84  
NU=IRES

DO 17 J=1,3,2  
CALL RIFF(IP,RRES,IRES,IND)  
IF(IND-1) 86,15,86  
15 IF(J.EQ.1) GO TO 16  
HNM2(1)=RRES  
GO TO 17  
16 HNM1(1)=RRES  
17 CONTINUE

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND.EQ.7) GO TO 18  
IERR=3  
RETURN

18 KODE(5)=9

DO 20 J=1,9  
IF(ITC.EQ.J) GO TO (21,22,23,24,25,26,27,28,29),J  
20 CONTINUE

```
21 CALL ROD1(NX,NU,A,B)
   KODE(5)=7
   KODE(6)=127
   IF(NX.EQ.7) GO TO 30
   KODE(6)=10922
   GO TO 30
C
22 CALL ROD2(NX,NU,A,B)
   KODE(5)=7
   KODE(6)=127
   IF(NX.EQ.7) GO TO 30
   KODE(6)=10922
   GO TO 30
C
23 CALL ROD3(NX,NU,A,B)
   KODE(6)=511
   IF(NX.EQ.9) GO TO 30
   KODE(6)=87381
   GO TO 30
C
24 CALL ROD4(NX,NU,A,B)
   KODE(5)=7
   KODE(6)=127
   IF(NX.EQ.7) GO TO 30
   KODE(6)=10922
   GO TO 30
C
25 CALL ROD5(NX,NU,A,B)
   KODE(6)=511
   IF(NX.EQ.9) GO TO 30
   KODE(6)=87381
   GO TO 30
C
26 CALL ROD6(NX,NU,A,B)
   KODE(6)=1022
   IF(NX.EQ.11) GO TO 30
   KODE(6)=174762
   GO TO 30
C
27 CALL ROD7(A,B)
   NX=11
   NU=2
   KODE(5)=7
   KODE(6)=762
   GO TO 30
C
28 CALL ROD8(A,B)
   NX=13
   NU=2
   KODE(6)=5621
   GO TO 30
C
29 CALL ROD9(A,B)
   NX=15
   NU=2
   KODE(6)=11242
C
C PRINT SYSTEM MATRICES
C
30 WRITE(6,200) HNM1(1)
   CALL MATUT(A,NX,NX,8)
   WRITE(6,200) HNM2(1)
   CALL MATUT(B,NX,NU,8)
C
```

```
C      TRANSFER SYSTEM MATRICES TO DISK
C
      DO 50 K=1,3,2
      KODE(1)=NX
      KODE(2)=NX
      IF(K.EQ.3) KODE(2)=NU
      IF(K-1) 44,40,44
40     CALL ENTER(1,HNM1)
      WRITE(1) (KODE(J),J=1,10)
      DO 42 II=1,NX
42     WRITE(1) (A(II,J),J=1,NX)
      GO TO 50
44     CALL ENTER(1,HNM2)
      WRITE(1) (KODE(J),J=1,10)
      DO 46 II=1,NX
46     WRITE(1) (B(II,J),J=1,NU)
50     CALL CLOSE(1)
      RETURN

C
C      ERROR CHECK
C
80     IF(IND-7) 84,82,84
82     IERR=2
      RETURN

C
84     IERR=4
      RETURN

C
86     IF(IND-7) 88,82,88
88     IERR=6
      RETURN

C
200    FORMAT(///8H MATRIX ,A5///)
C
      END
```

```
      SUBROUTINE ROD1(NX,NU,A,B)
C
C   GENERATES A CONTINUOUS MODEL OF A DIFFUSION PROCESS,
C   USING FINITE DIFFERENCES OF ORDER H**2
C   THE DYNAMICS OF THE SERVOS ARE NEGLECTED
C
C   AUTHOR:      TORSTEN CROMNOW 1974-05-16
C
C   SUBROUTINE REQUIRED
C       NONE
C
C   DIMENSION A(1,1),B(1,1)
C
C   COMMON /DEFHR/ RL,TD,THL,T1,T2,G1,G2,ZETA1,ZETA2,
*W01,W02
C
C   INITIATE
C
C   HZ=RL/(NX+1)
C   BETA=TD/(HZ*HZ)
C   ALFA=2*BETA+THL
C
C   SET UP SYSTEM MATRICES OF PROCESS
C
C   DO 12 I=1,NX
C   DO 10 J=1,NX
10    A(I,J)=0.
C   DO 12 J=1,NU
12    B(I,J)=0.
C
C   NI=NX-1
C   DO 20 I=1,NI
C   I1=I+1
20    A(I,I1)=BETA
C   DO 22 I=1,NX
22    A(I,I)=-ALFA
C   B(1,1)=BETA
C   IF(NU.EQ.2) B(NX,NU)=BETA
C
C   RETURN
C
C   END
```

```

SUBROUTINE ROD2(NX,NU,A,B)
C
C   GENERATES A CONTINUOUS MODEL OF A DIFFUSION PROCESS,
C   USING FINITE DIFFERENCES OF ORDER H**2 AND H**4
C   THE DYNAMICS OF THE SERVOS ARE NEGLECTED
C
C   AUTHOR:      TORSTEN CROMNOW 1974-05-16
C
C   SUBROUTINE REQUIRED
C       NONE
C
C   DIMENSION A(1,1),B(1,1)
C
C   COMMON /DEFHR/ RL,TD,THL,T1,T2,G1,G2,ZETA1,ZETA2,
*W01,W02
C
C   INITIATE
C
C   HZ=RL/(NX+1)
C   BETA=TD/(HZ*HZ)
C   ALFA=2*BETA+THL
C   EPS=TD/(12.*HZ*HZ)
C   GAM=30.*EPS+THL
C   DEL=16.*EPS
C
C   SET UP SYSTEM MATRICES OF PROCESS
C
C   DO 12 I=1,NX
C   DO 10 J=1,NX
10  A(I,J)=0.
C   DO 12 J=1,NU
12  B(I,J)=0.
C
C   NI=NX-1
C   DO 20 I=1,NI
C   I1=I+1
20  A(I,I1)=DEL
C   A(I1,I)=DEL
C
C   NI=NX-2
C   DO 22 I=1,NI
C   I1=I+2
22  A(I,I1)=-EPS
C   A(I1,I)=-EPS
C
C   DO 24 I=1,NX
24  A(I,I)=-GAM
C
C   A(1,1)=-ALFA
C   A(1,2)=BETA
C   A(1,3)=0.
C   A(NX,NX-2)=0.
C   A(NX,NX-1)=BETA
C   A(NX,NX)=-ALFA
C   B(1,1)=BETA
C   B(2,1)=-EPS
C   IF(NU.EQ.1) RETURN
C   B(NX-1,NU)=-EPS
C   B(NX,NU)=BETA
C
C   RETURN
C
C   END

```

```

SUBROUTINE ROD3(NX,NU,A,B)
C
C   GENERATES AN EXTENDED CONTINUOUS MODEL OF A
C   DIFFUSION PROCESS,USING FINITE DIFFERENCES
C   OF ORDER H**2 AND H**4,
C   THE DYNAMICS OF THE SERVOS ARE OF FIRST ORDER
C
C   AUTHOR:      TORSTEN CROMNOW 1974-05-16
C
C   SUBROUTINE REQUIRED
C       NONE
C
C   DIMENSION A(1,1),B(1,1)
C
C   COMMON /DEFHR/ RL,TD,THL,T1,T2,G1,G2,ZETA1,ZETA2,
*W01,W02
C
C   INITIATE
C
C   HZ=RL/(NX-1)
C   BETA=TD/(HZ*HZ)
C   ALFA=2*BETA+THL
C   EPS=TD/(12.*HZ*HZ)
C   GAM=30.*EPS+THL
C   DEL=16.*EPS
C
C   SET UP SYSTEM MATRICES OF PROCESS
C
C   DO 12 I=1,NX
C   DO 10 J=1,NX
10  A(I,J)=0.
C   DO 12 J=1,NU
12  B(I,J)=0.
C   NI=NX-1
C   DO 20 I=1,NI
C   I1=I+1
20  A(I,I1)=DEL
C   NI=NX-2
C   DO 22 I=1,NI
C   I1=I+2
22  A(I,I1)=-EPS
C   DO 24 I=1,NX
24  A(I,I)=-GAM
C   A(1,2)=0.
C   A(2,3)=BETA
C   A(2,1)=BETA
C   A(NX,NX-1)=0.
C   A(NX-1,NX-2)=BETA
C   A(NX-1,NX)=BETA
C   A(1,1)=-1/T1
C   A(2,2)=-ALFA
C   A(NX-1,NX-1)=-ALFA
C   A(NX,NX)=-1/T2
C   A(1,3)=0.
C   A(2,4)=0.
C   A(NX-1,NX-3)=0.
C   A(NX,NX-2)=0.
C   B(1,1)=1/T1
C   IF(NU.EQ.2) B(NX,NU)=1/T2
C   RETURN
C   END

```

```

SUBROUTINE ROD4(NX,NU,A,B)
C
C   GENERATES A CONTINUOUS MODEL OF A DIFFUSION PROCESS
C   USING FINITE DIFFERENCES OF ORDER H**2,H**4 AND H**6
C   THE DYNAMICS OF THE SERVOS ARE NEGLECTED
C
C   AUTHOR:      TORSTEN CROMNOW 1974-07-12
C
C   SUBROUTINE REQUIRED
C       NONE
C
C   DIMENSION A(1,1),B(1,1)
C
C   COMMON /DEFHR/ RL,TD,THL,T1,T2,G1,G2,ZETA1,ZETA2,
*W01,W02
C
C   INITIATE
C
C   HZ=RL/(NX+1)
C   HA=TD/(HZ*HZ)
C   A1=-2.*HA-THL
C   A2=HA
C
C   HB=TD/(12.*HZ*HZ)
C   B1=-30.*HB-THL
C   B2=16.*HB
C   B3=-HB
C
C   HC=TD/(180.*HZ*HZ)
C   C1=-490.*HC-THL
C   C2=270.*HC
C   C3=-27.*HC
C   C4=2.*HC
C
C   SET UP SYSTEM MATRICES OF PROCESS
C
C   DO 12 I=1,NX
C   DO 10 J=1,NX
10  A(I,J)=0.
C   DO 12 J=1,NU
12  B(I,J)=0.
C
C   DO 20 I=1,NX
20  A(I,I)=C1
C   A(1,1)=A1
C   A(2,2)=B1
C   A(NX,NX)=A1
C   A(NX-1,NX-1)=B1
C
C   NI=NX-1
C   DO 22 I=1,NI
C   I1=I+1
22  A(I,I1)=C2
C   A(I1,I)=C2
C   A(1,2)=A2
C   A(2,1)=B2
C   A(2,3)=B2
C   A(NX,NX-1)=A2
C   A(NX-1,NX-2)=B2
C   A(NX-1,NX)=B2
C
C   NI=NX-2

```



```
DO 24 I=1,NI
  I1=I+2
  A(I,I1)=C3
24  A(I1,I)=C3
  A(1,3)=0.
  A(2,4)=B3
  A(NX,NX-2)=0.
  A(NX-1,NX-3)=B3
C
  NI=NX-3
  DO 26 I=1,NI
  I1=I+3
  A(I,I1)=C4
26  A(I1,I)=C4
  A(1,4)=0.
  A(2,5)=0.
  A(NX,NX-3)=0.
  A(NX-1,NX-4)=0.
C
  B(1,1)=A2
  B(2,1)=B3
  B(3,1)=C4
  IF(NU.EQ.1) RETURN
  B(NX,2)=A2
  B(NX-1,2)=B3
  B(NX-2,2)=C4
C
  RETURN
C
  END
```

```

SUBROUTINE ROD5(NX,NU,A,B)
C
C   GENERATES AN EXTENDED CONTINUOUS MODEL OF A
C   DIFFUSION PROCESS USING FINITE DIFFERENCES
C   OF ORDER H**2,H**4 AND H**6
C   THE DYNAMICS OF THE SERVOS ARE OF FIRST ORDER
C
C   AUTHOR:      TORSTEN CROMNOW 1974-09-01
C
C   SUBROUTINE REQUIRED
C     ROD4
C
C   DIMENSION A(1,1),B(1,1),AA(20,20)
C
C   COMMON /DEFHR/ RL,TD,THL,T1,T2,G1,G2,ZETA1,ZETA2,
*W01,W02
COMMON AA
C
C   INITIATE
C
C   DO 10 K=1,NX
C   DO 10 L=1,NX
10  A(K,L)=0.
C
C   SET UP SYSTEM MATRICES OF PROCESS
C
C   NX1=NX-2
C   CALL ROD4(NX1,NU,AA,B)
C   DO 20 K=1,NX1
C   DO 20 L=1,NX1
20  A(K+1,L+1)=AA(K,L)
C
C   A(1,1)=-1./T1
C   A(2,1)=B(1,1)
C   A(3,1)=B(2,1)
C   A(4,1)=B(3,1)
C   A(NX,NX)=-1./T2
C   A(NX-1,NX)=B(1,1)
C   A(NX-2,NX)=B(2,1)
C   A(NX-3,NX)=B(3,1)
C
C   DO 30 K=1,NX
C   DO 30 L=1,NU
30  B(K,L)=0.
C
C   B(1,1)=1./T1
C   IF(NU.EQ.1) RETURN
C   B(NX,2)=1./T2
C
C   RETURN
C
C   END

```

```

SUBROUTINE ROD6(NX,NU,A,B)
C
C   GENERATES AN EXTENDED CONTINUOUS MODEL OF A
C   DIFFUSION PROCESS USING FINITE DIFFERENCES
C   OF ORDER H**2,H**4,AND H**6.
C   THE DYNAMICS OF THE SERVOS ARE OF SECOND ORDER
C
C   AUTHOR:      TORSTEN CROMNOW 1974-09-15
C
C   SUBROUTINE REQUIRED
C     ROD4
C
C   DIMENSION A(1,1),B(1,1),AA(20,20)
C
C   COMMON /DEFHR/ RL,TD,THL,T1,T2,G1,G2,ZETA1,ZETA2,
*   W01,W02
C   COMMON AA
C
C   INITIATE
C
C   DO 10 K=1,NX
C   DO 10 L=1,NX
10  A(K,L)=0.
C
C   SET UP SYSTEM MATRICES OF PROCESS
C
C   NX1=NX-4
C   CALL ROD4(NX1,NU,AA,B)
C   DO 20 K=1,NX1
C   DO 20 L=1,NX1
20  A(K+2,L+2)=AA(K,L)
C
C   A(2,1)=1.
C
C   A(1,2)=-W01*W01
C   A(2,2)=-2.*ZETA1*W01
C   A(3,2)=B(1,1)
C   A(4,2)=B(2,1)
C   A(5,2)=B(3,1)
C
C   A(NX-1,NX)=1.
C
C   A(NX,NX-1)=-W02*W02
C   A(NX-1,NX-1)=-2.*ZETA2*W02
C   A(NX-2,NX-1)=B(1,1)
C   A(NX-3,NX-1)=B(2,1)
C   A(NX-4,NX-1)=B(3,1)
C
C   DO 30 K=1,NX
C   DO 30 L=1,NU
30  B(K,L)=0.
C
C   B(1,1)=W01*W01
C   B(2,1)=G1
C   IF(NU.EQ.1) RETURN
C   B(NX,2)=W02*W02
C   B(NX-1,2)=G2
C
C   RETURN
C
C   END

```

```

SUBROUTINE ROD7(A,B)
C
C   GENERATES A CONTINUOUS MODEL OF A DIFFUSION PROCESS,
C   USING FINITE DIFFERENCES OF ORDER H**2,H**4,H**6 AND H**8
C   THE DYNAMICS OF THE SERVOS ARE NEGLECTED
C
C   AUTHOR:      TORSTEN CROMNOW 1974-08-16
C
C   SUBROUTINE REQUIRED
C       NONE
C
C   DIMENSION A(1,1),B(1,1)
C
C   COMMON /DEFHR/ RL,TD,THL,T1,T2,G1,G2,ZETA1,ZETA2,
*W01,W02
C
C   INITIATE
C
C   HZ1=RL/8.
C   HZ2=HZ1/2.
C
C   HA=TD/(HZ2*HZ2)
C   A1=-2.*HA-THL
C   A2=HA
C
C   HB=TD/(12.*HZ2*HZ2)
C   B1=-30.*HB-THL
C   B2=16.*HB
C   B3=-HB
C
C   HC=TD/(12.*HZ1*HZ1)
C   C1=-30.*HC-THL
C   C2=16.*HC
C   C3=-HC
C
C   HD=TD/(180.*HZ1*HZ1)
C   D1=-490.*HD-THL
C   D2=270*HD
C   D3=-27.*HD
C   D4=2.*HD
C
C   HE=TD/(5040.*HZ1*HZ1)
C   E1=-13090.*HE-THL
C   E2=7056.*HE
C   E3=-504.*HE
C   E4=-16.*HE
C   E5=9.*HE
C
C   SET UP SYSTEM MATRICES OF PROCESS
C
C   DO 12 I=1,11
C   DO 10 J=1,11
10  A(I,J)=0.
C   DO 12 J=1,2
12  B(I,J)=0.
C
C   A(1,1)=A1
C   A(1,2)=A2
C
C   A(2,1)=B2
C   A(2,2)=B1
C   A(2,3)=B2

```

C A(2,4)=B3  
A(3,2)=A2  
A(3,3)=A1  
A(3,4)=A2  
C A(4,2)=C2  
A(4,4)=C1  
A(4,5)=C2  
A(4,6)=C3  
C A(5,2)=D3  
A(5,4)=D2  
A(5,5)=D1  
A(5,6)=D2  
A(5,7)=D3  
A(5,8)=D4  
C A(6,2)=E4  
A(6,4)=E3  
A(6,5)=E2  
A(6,6)=E1  
A(6,7)=E2  
A(6,8)=E3  
A(6,10)=E4  
C A(7,4)=D4  
A(7,5)=D3  
A(7,6)=D2  
A(7,7)=D1  
A(7,8)=D2  
A(7,10)=D3  
C A(8,6)=C3  
A(8,7)=C2  
A(8,8)=C1  
A(8,10)=C2  
C A(9,8)=A2  
A(9,9)=A1  
A(9,10)=A2  
C A(10,8)=B3  
A(10,9)=B2  
A(10,10)=B1  
A(10,11)=B2  
C A(11,10)=A2  
A(11,11)=A1  
C B(1,1)=A2  
B(2,1)=B3  
B(4,1)=C3  
B(5,1)=D4  
B(6,1)=E5  
B(6,2)=E5  
B(7,2)=D4  
B(8,2)=C3  
B(10,2)=B3  
B(11,2)=A2  
C  
C RETURN  
END

```

SUBROUTINE ROD8(A,B)
C
C   GENERATES AN EXTENDED CONTINUOUS MODEL OF A
C   DIFFUSION PROCESS USING FINITE DIFFERENCES
C   OF ORDER H**2,H**4,H**6 AND H**8
C   THE DYNAMICS OF THE SERVOS ARE OF FIRST ORDER
C
C   AUTHOR:      TORSTEN CROMNOW 1974-09-01
C
C   SUBROUTINE REQUIRED
C       ROD7
C
C   DIMENSION A(1,1),B(1,1),AA(11,11)
C
C   COMMON /DEFHR/ RL,TD,THL,T1,T2,G1,G2,ZETA1,ZETA2,
*W01,W02
COMMON AA
C
C   INITIATE
C
C   DO 10 K=1,13
C   DO 10 L=1,13
10  A(K,L)=0.
C
C   SET UP SYSTEM MATRICES OF PROCESS
C
C   CALL ROD7(AA,B)
C   DO 20 K=1,11
C   DO 20 L=1,11
20  A(K+1,L+1)=AA(K,L)
C
C   A(1,1)=-1./T1
C   A(2,1)=B(1,1)
C   A(3,1)=B(2,1)
C   A(4,1)=B(3,1)
C   A(5,1)=B(4,1)
C   A(6,1)=B(5,1)
C   A(7,1)=B(6,1)
C
C   A(7,13)=B(6,2)
C   A(8,13)=B(7,2)
C   A(9,13)=B(8,2)
C   A(10,13)=B(9,2)
C   A(11,13)=B(10,2)
C   A(12,13)=B(11,2)
C   A(13,13)=-1./T2
C
C   DO 30 K=1,13
C   DO 30 L=1,2
30  B(K,L)=0.
C
C   B(1,1)=1./T1
C   B(13,2)=1./T2
C
C   RETURN
C
C   END

```

```

SUBROUTINE ROD9(A,B)
C
C   GENERATES AN EXTENDED CONTINUOUS MODEL OF A
C   DIFFUSION PROCESS USING FINITE DIFFERENCES OF
C   ORDER H**2,H**4,H**6 AND H**8
C   THE DYNAMICS OF THE SERVOS ARE OF SECOND ORDER
C
C   AUTHOR:      TORSTEN GROMNOW 1974-09-15
C
C   SUBROUTINE REQUIRED
C     ROD7
C
C   DIMENSION A(1,1),B(1,1),AA(11,11)
C
C   COMMON /DEFHR/ RL,TD,THL,T1,T2,G1,G2,ZETA1,ZETA2,
*W01,W02
COMMON AA
C
C   INITIATE
C
DO 10 I=1,15
DO 10 J=1,15
10  A(I,J)=0.
C
C   SET UP SYSTEM MATRICES OF PROCESS
C
CALL ROD7(AA,B)
DO 20 I=1,11
DO 20 J=1,11
20  A(I+2,J+2)=AA(I,J)
C
A(2,1)=1.
A(1,2)=-W01*W01
A(2,2)=-2.*ZETA1*W01
A(3,2)=B(1,1)
A(4,2)=B(2,1)
A(5,2)=B(3,1)
A(6,2)=B(4,1)
A(7,2)=B(5,1)
A(8,2)=B(6,1)
A(8,14)=B(6,2)
A(9,14)=B(7,2)
A(10,14)=B(8,2)
A(11,14)=B(9,2)
A(12,14)=B(10,2)
A(13,14)=B(11,2)
A(14,14)=-2.*ZETA2*W02
A(15,14)=-W02*W02
A(14,15)=1.
C
DO 30 I=1,15
DO 30 J=1,2
30  B(I,J)=0.
C
B(1,1)=W01*W01
B(2,1)=G1
B(14,2)=G2
B(15,2)=W02*W02
C
RETURN
C
END

```

SUBROUTINE SAMP(IERR)

COMPUTES THE DISCRETE SYSTEM HNM3(FI),HNM4(GAM)  
CORRESPONDING TO THE CONTINUOUS SYSTEM HNM1(A),HNM2(B)  
AND THE SAMPLINGINTERVAL TSAMP

AUTHOR: TORSTEN CROMNOW 1974-06-04

SUBROUTINES REQUIRED

COSA1  
EXPAN1

LOGICAL IFOUND

DIMENSION A(20,20),B(20,20),KODE(10)  
DIMENSION HNM1(2),HNM2(2),HNM3(2),HNM4(2)

COMMON /CMD/ IP,IND,BUFF(16)

DATA HNM1/5H ,4H BIN/  
DATA HNM2/5H ,4H BIN/  
DATA HNM3/5H ,4H BIN/  
DATA HNM4/5H ,4H BIN/

READ ARGUMENTS

DO 8 I=1,2  
CALL RIFF(IP,RRES,IRES,IND)  
IF(IND-1) 76,2,76  
GO TO (4,6),I  
HNM1(1)=RRES  
GO TO 8  
HNM2(1)=RRES  
CONTINUE

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND-4) 70,10,70  
IF(RRES.LE.0.) GO TO 74  
TSAMP=RRES

DO 20 I=1,2  
CALL RIFF(IP,RRES,IRES,IND)  
IF(IND-1) 76,12,76  
GO TO(14,16),I  
HNM3(1)=RRES  
GO TO 20  
HNM4(1)=RRES  
CONTINUE

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND.NE.7) GO TO 80

TEST IF INPUT FILES ARE PRESENT ON DISK

CALL FSTAT(1,HNM1,IFOUND)  
IF(.NOT, IFOUND) GO TO 82  
CALL FSTAT(1,HNM2,IFOUND)  
IF(.NOT, IFOUND) GO TO 82

GET HNM1 AND HNM2 FROM DISK



```

C
  CALL SEEK(1,HNM1)
  READ(1) (KODE(1),I=1,10)
  NX=KODE(1)
  IF(NX.GT.20) GO TO 78
  IF(NX.NE.KODE(2)) GO TO 78
C
  DO 30 I=1,NX
30  READ(1) (A(I,K),K=1,NX)
  CALL CLOSE(1)
C
  CALL SEEK(1,HNM2)
  READ(1) (KODE(1),I=1,10)
  IF(NX.NE.KODE(1)) GO TO 78
  NU=KODE(2)
  IF(NU.GT.20) GO TO 78
C
  DO 34 I=1,NX
34  READ(1) (B(I,K),K=1,NU)
  CALL CLOSE(1)
C
C   COMPUTE DISCRETE SYSTEM
C
  DO 44 I=1,NX
  DO 40 J=1,NX
40  A(I,J)=A(I,J)*TSAMP
  DO 42 J=1,NU
42  B(I,J)=B(I,J)*TSAMP
44  CONTINUE
  CALL COSA1(A,B,NX,NU)
C
C   PRINT FI AND GAMMA
C
  WRITE(6,100) HNM1(1),HNM2(1),TSAMP,HNM3(1),HNM4(1)
  WRITE(6,200) HNM3(1)
  CALL MATUT(A,NX,NX,8)
  WRITE(6,200) HNM4(1)
  CALL MATUT(B,NX,NU,8)
C
C   WRITE FI AND GAMMA ON DISK
C
  KODE(1)=NX
  KODE(2)=NX
  KODE(4)=TSAMP*50,
  CALL ENTER(1,HNM3)
  WRITE(1) (KODE(I),I=1,10)
  DO 50 I=1,NX
50  WRITE(1) (A(I,J),J=1,NX)
  CALL CLOSE(1)
C
  KODE(2)=NU
  CALL ENTER(1,HNM4)
  WRITE(1) (KODE(I),I=1,10)
  DO 60 I=1,NX
60  WRITE(1) (B(I,J),J=1,NU)
  CALL CLOSE(1)
C
  RETURN
C
70  IF(IND-7) 74,72,74
72  IERR=2
  RETURN
C
74  IERR=5

```

```
      RETURN
C
76    IF(IND-7) 78,72,78
78    IERR=6
      RETURN
C
80    IERR=3
      RETURN
C
82    IERR=8
      RETURN
C
100   FORMAT(///5H SAMP,2(1X,A5),1X,1F8.3,2(1X,A5)///)
200   FORMAT(///8H MATRIX ,A5///)
C
      END
```

```

SUBROUTINE THETA(IERR)
C
C
C   GENERATES A 'THETA'-MATRIX FOR
C   THE KALMAN ESTIMATION PROBLEM
C
C   AUTHOR:      TORSTEN CROMNOW 1974-06-05
C
C   SUBROUTINES REQUIRED
C     NONE
C
C   DIMENSION KODE(10),TH(9,20),HNM(2)
C
C   COMMON /CMD/ IP,IND,BUFF(16)
C
C   DATA HNM/5H      ,4H BIN/
C
C   READ ARGUMENTS
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-1) 60,4,60
4   HNM(1)=RRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-3) 66,6,66
6   IF(IRES.LT.1.OR.IRES.GT.9) GO TO 68
   NY=IRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-3) 66,8,66
8   IF(IRES.LT.1.OR.IRES.LT.NY.OR.IRES.GT.20) GO TO 68
   NX=IRES
C
C   GENERATE OUTPUT MATRIX
C
C   DO 12 I=1,NY
C   DO 12 J=1,NX
12  TH(I,J)=0.
   IDUM=0
   DO 20 I=1,NY
   CALL RIFF(IP,RRES,IRES,IND)
   IF(IND.NE.3.OR.IRES.LE.IDUM.OR.IRES.GT.NX) GO TO 70
   IDUM=IRES
20  TH(I,IRES)=1.
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.7) GO TO 74
C
C   COMPUTE KODE(5)
C
C   KODE(5)=0
C   DO 28 I=1,NX
C   TOT=0.
C   DO 26 J=1,NY
26  TOT=TH(J,I)+TOT
   K=0
   IF(TOT.EQ.1.) K=1
   KODE(5)=KODE(5)+K
28  KODE(5)=KODE(5)+K
C
C   PRINT THETA
C

```

```
30  WRITE(6,200) HNM(1)
    CALL MATUT(TH,NY,NX,8)
C
C  WRITE THETA ON DISK
C
    KODE(1)=NY
    KODE(2)=NX
    CALL ENTER(1,HNM)
    WRITE(1) (KODE(I),I=1,10)
    DO 40 I=1,NY
40  WRITE(1) (TH(I,J),J=1,NX)
    CALL CLOSE(1)
C
    RETURN
C
C  ERROR CHECK
C
60  IF(IND-7) 64,62,64
62  IERR=2
    RETURN
C
64  IERR=6
    RETURN
C
66  IF(IND-7) 68,62,68
68  IERR=4
    RETURN
C
70  IF(IND-7) 68,72,68
72  IERR=2
    RETURN
74  IERR=3
    RETURN
C
200 FORMAT(///8H MATRIX ,A5///)
C
    END
```

```

SUBROUTINE CMTX(IERR)
C
C
C   GENERATES A DIAGONAL COVARIANCE MATRIX
C
C
C   AUTHOR:      TORSTEN CROMNOW 1974-06-04
C
C   SUBROUTINES REQUIRED
C       NONE
C
C
C   DIMENSION KODE(10),R(20,20),HNM(2)
C
C   COMMON /CMD/ IP,IND,BUFF(16)
C
C   DATA HNM/5H      ,4H BIN/
C
C   READ ARGUMENTS AND GENERATE COVARIANCE MATRIX
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-1) 68,4,68
C   HNM(1)=RRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-3) 50,10,50
C   IF(IRES.LE.0) GO TO 54
C   NX=IRES
C
C   DO 14 I=1,NX
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.EQ.4.OR.IND.EQ.7) GO TO 11
C   GO TO 64
C 11  IF(IND-4) 12,14,12
C 12  CALL RLINE(9,8,BUFF,2)
C   IP=1
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-4) 60,14,60
C 14  R(I,I)=RRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.7) GO TO 66
C
C   PRINT COVARIANCE MATRIX
C
C   WRITE(6,200) HNM(1)
C   CALL MATUT(R,NX,NX,8)
C
C   TRANSFER COVARIANCE MATRIX MATRIX TO DISK
C
C   KODE(1)=NX
C   KODE(2)=NX
C   CALL ENTER(1,HNM)
C   WRITE(1) (KODE(I),I=1,10)
C   DO 20 I=1,NX
C 20  WRITE(1) (R(I,J),J=1,NX)
C   CALL CLOSE(1)
C   RETURN
C
C   ERROR CHECK
C
C 50  IF(IND-7) 54,52,54
C 52  IERR=2

```

```
      RETURN
C
  54  IERR=4
      RETURN
C
  60  IF(IND-7) 64,52,64
  64  IERR=5
      RETURN
C
  66  IERR=3
      RETURN
C
  68  IF(IND-7) 70,52,70
  70  IERR=6
      RETURN
C
 200  FORMAT(///8H MATRIX ,A5///)
C
      END
```

SUBROUTINE TRANS(IERR)

COMPUTES THE DISCRETE COVARIANCE MATRICES R1D,R12D,R2D  
CORRESPONDING TO THE CONTINUOUS COVARIANCE MATRICES  
R1,R12,R2 FOR THE KALMAN ESTIMATION PROBLEM

AUTHOR: TORSTEN CROMNOW 1974-07-01

SUBROUTINES REQUIRED

STRANS  
INRIC1  
RESYS  
MATRE  
FILES

LOGICAL DLOG,KLOG

DIMENSION A(20,20),B(20,20),Q11(20,20),Q12(20,20),  
\*TEMP1(20,20),TEMP2(20,20),TEMP4(20,20),TEMP3(20,20),  
\*TEMP5(20,20),SNAME(2),Q22(20,20),IT(10),Q0(20,20),  
\*QNAM1(2),QNAM2(2),QNAM3(2),QDNAM(4)

COMMON /CMD/ IP,IND,BUFF(16)  
COMMON /DEF/ NMAX  
COMMON /SYCO/ ISYS(11)  
COMMON A,B,Q0,Q11,Q12,Q22,TEMP1,TEMP2,TEMP3,TEMP4,TEMP5

EQUIVALENCE (IERR1,ISYS(2))

DATA MAX /20/  
DATA KLOG /,TRUE./  
DATA IS /20/  
DATA QNAM1 /5H ,4H BIN/  
DATA QNAM2 /5H ,4H BIN/  
DATA QNAM3 /5H ,4H BIN/

INITIATE

DLOG=.FALSE.  
IERR1=0

READ ARGUMENTS

DO 2 I=1,2  
CALL RIFF(IP,RRES,IRES,IND)  
IF(IND.NE.1) GO TO 40  
SNAME(I)=RRES

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND.NE.4) GO TO 46  
IF(RRES.EQ.0.) GO TO 46  
TSAMP=RRES

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND.NE.1) GO TO 40  
DNAME=RRES

CALL RIFF(IP,RRES,IRES,IND)

```

      IF(IND,NE.7) GO TO 48
C
      WRITE(6,200) SNAME(1),SNAME(2),TSAMP,DNAME
C
C      READ SYSTEM MATRICES AND LOSS MATRICES
C      A=AT,B=CT,Q11=R1,Q12=R12,Q22=R2
C
      CALL INRIC1(SNAME(1),SNAME(2),A,B,Q0,Q11,Q12,Q22,
*NX,NU,DLOG,KLOG,MAX,IT,TEMP1,TEMP2,TEMP3)
      IF(DLOG.EQ..TRUE.) GO TO 50
      IF(IERR1.NE.0) GO TO 50
C
C      COMPUTE R1D,R12D AND R2D
C
      CALL STRANS(A,B,Q11,Q12,Q22,NX,NU,TSAMP,NMAX,NOCONV,
*TEMP1,TEMP2,TEMP3,TEMP4,TEMP5,IS)
      IF(NOCONV.EQ.1) GO TO 52
C
C      COMPUTE SYMMETRI ERROR IN R1D AND SYMMETRIZE R1D
C      (SYMMETRI ERROR=NORM(R1D-R1DT)/NORM(R1D)
C
      DO 10 I=1,NX
10      CALL RMOVE(Q11(I,1),MAX,TEMP1(1,I),1,NX)
      CALL MADD(Q11,TEMP1,TEMP2,NX,NX,0,0,1)
      CALL NORM(TEMP2,NX,MAX,DNORM)
      CALL NORM(Q11,NX,MAX,QNORM)
      SERR=DNORM/QNORM
C
      DO 12 I=1,NX
      DO 12 J=1,NX
12      Q11(I,J)=(Q11(I,J)+TEMP1(I,J))/2.
C
C      READ OUTPUT SYSTEM
C
      CALL RESYS(DNAME,QDNAM,4)
      IF(IERR1.NE.0) GO TO 50
      QNAM1(1)=QDNAM(2)
      QNAM2(1)=QDNAM(3)
      QNAM3(1)=QDNAM(4)
C
C      PRINT RESULTS
C
      WRITE(6,100) QNAM1(1)
      CALL MATUT(Q11,NX,NX,8)
      WRITE(6,100) QNAM2(1)
      CALL MATUT(Q12,NX,NU,8)
      WRITE(6,100) QNAM3(1)
      CALL MATUT(Q22,NU,NU,8)
      WRITE(6,300) QNAM1(1),SERR
C
C      WRITE Q11,Q12 AND Q22 ON DISK
C
      IT(1)=NX
      IT(2)=NX
      IT(4)=TSAMP*50.
      CALL ENTER(1,QNAM1)
      WRITE(1) (IT(I),I=1,10)
      DO 20 I=1,NX
20      WRITE(1) (Q11(I,J),J=1,NX)
      CALL CLOSE(1)
C
      IT(2)=NU
      CALL ENTER(1,QNAM2)
      WRITE(1) (IT(I),I=1,10)

```



```
DO 25 I=1,NX
25 WRITE(1) (Q12(I,J),J=1,NU)
CALL CLOSE(1)
C
IT(1)=NU
CALL ENTER(1,QNAM3)
WRITE(1) (IT(I),I=1,10)
DO 30 I=1,NU
30 WRITE(1) (Q22(I,J),J=1,NU)
CALL CLOSE(1)
C
RETURN
C
C ERROR CHECK
C
40 IF(IND-7) 44,42,44
42 IERR=2
RETURN
C
44 IERR=6
RETURN
C
46 IERR=5
RETURN
C
48 IERR=3
RETURN
C
50 IERR=7
52 RETURN
C
100 FORMAT(///27H DISCRETE COVARIANCE MATRIX,5X,A5///)
200 FORMAT(///6H TRANS,1X,2(A5,1X),1X,1F8.3,1X,A5///)
300 FORMAT(///18H SYMMETRI ERROR IN,1X,A5,5X,1E10.3///)
C
END
```

SUBROUTINE KALFI(IERR)

COMPUTES THE KALMANGAIN HNMK OF A DISCRETE SYSTEM  
DESCRIBED BY THE SYSTEM NAMES DSYST AND DCOV

AUTHOR: TORSTEN CROMNOW 1974-06-28

SUBROUTINES REQUIRED

INRIC1  
RESYS  
MATRE  
FILES  
DRIC1  
EIGP1  
HESQR  
SCAL  
REALVE  
COMPVE

LOGICAL DLOG,KLOG

DIMENSION A(20,20),B(20,20),Q1(20,20),Q2(20,20),EVI(20),  
1 TEMP4(20,20),Q12(20,20),COM(3),TEMP1(20,20),  
2 TEMP2(20,20),TEMP3(20,20),SNAME(2),EVR(20),  
3 S(20,20),HNMK(2),IT(10),RL(20,20),INDIC(20),  
4 HNMP(2)

COMMON /CMD/ IP,IND,BUFF(16)  
COMMON /SYCO/ ISYS(11)  
COMMON A,B,Q1,Q12,Q2,S,TEMP1,TEMP2,TEMP3,TEMP4,RL

EQUIVALENCE (IERR1,ISYS(2))

DATA COM/2HNO,4HYES1,4HYES2/  
DATA MAX/20/  
DATA KLOG/.TRUE./  
DATA HNMK /5H ,4H BIN/  
DATA HNMP /5HPSTAT,4H BIN/

INITIATE

ITER=0  
IERR1=0

READ ARGUMENTS

DO 2 I=1,2  
CALL RIFF(IP,RRES,IRES,IND)  
IF(IND.NE.1) GO TO 78  
SNAME(I)=RRES

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND.NE.1)GO TO 78  
HNMK(1)=RRES

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND.NE.7) GO TO 84

WRITE(6,100) (SNAME(I),I=1,2),HNMK(1)

```

C      READ SYSTEM AND LOSS MATRICES
C      A=FIT,B=THT,S=R0,Q1=R1,Q12=R12,Q2=R2
C
C      CALL INRIC1(SNAME(1),SNAME(2),A,B,S,Q1,Q12,Q2,
*      NX,NU,DLOG,KLOG,MAX,IT,TEMP1,TEMP2,TEMP3)
C      IF(.NOT. DLOG) GO TO 80
C      IF(IERR1.NE.0) GO TO 80
C
C      ITERATE DISCRETE RICCATI EQUATION
C      (S STATIONARY SOLUTION OF RICCATI EQ. AND K=RLT)
C
C      11 CALL DRIC1(A,B,Q1,Q12,Q2,S,RL,NX,NU,MAX,IERR,
*      TEMP1,TEMP2,TEMP3,TEMP4)
C
C      IF(IERR.EQ.2) GO TO 88
C      IF(IERR.EQ.0) GO TO 22
C
C      200 ITERATIONS DID NOT LEAD TO CONVERGENS
C
C      IERR=0
C      ITER=ITER+200
C      WRITE(9,900) ITER
C      WRITE(9,1000)
C
C      READ NEW COMMAND FROM TT
C
C      12 CALL RLINE(9,8,BUFF,1)
C      IP=1
C      CALL RIFF(IP,RRES,IRES,IND)
C      DO 14 I=1,3
C      IF(RRES.EQ.COM(I)) GO TO(16,17,18),I
C      14 CONTINUE
C      WRITE(9,600)
C      GO TO 12
C
C      NO
C
C      16 IERR=0
C      RETURN
C
C      YES1 (CONTINUE WITHOUT OUTPUT)
C
C      17 GO TO 11
C
C      YES2 (CONTINUE WITH OUTPUT OF S AND RLT)
C
C      18 WRITE(6,700) ITER
C      CALL MATUT(S,NX,NX,8)
C      DO 20 I=1,NX
C      20 CALL RMOVE(RL(1,I),1,TEMP3(I,1),MAX,NU)
C      WRITE(6,800) ITER
C      CALL MATUT(TEMP3,NX,NU,8)
C      GO TO 11
C
C      READY COMPUTE  $TEMP1=FI-K*TH=PSI$ 
C
C      22 CALL MULT(RL,B,TEMP2,NX,NU,NX,1,1)
C      CALL MADD(A,TEMP2,TEMP1,NX,NX,1,0,1)
C
C      PRINT RESULTS AND COMPUTE EIGENVALUES OF PSI
C
C      DO 24 I=1,NX
C      24 CALL RMOVE(RL(1,I),1,TEMP3(I,1),MAX,NU)
C      WRITE(6,200)

```

```

CALL MATUT(S,NX,NX,8)
C
IT(1)=NX
IT(2)=NX
CALL ENTER(1,HNMP)
WRITE(1) (IT(I),I=1,10)
DO 25 I=1,NX
25 WRITE(1) (S(I,J),J=1,NX)
CALL CLOSE(1)
C
WRITE(6,300) HNMK(1)
CALL MATUT(TEMP3,NX,NU,8)
WRITE(6,400)
CALL MATUT(TEMP1,NX,NX,8)
CALL EIGP1(TEMP1,NX,26,,EVR,EVI,TEMP2,TEMP4,INDIC)
WRITE(6,500)
NFOUND=0
DO 26 I=1,NX
IF(INDIC(I).NE.0) GO TO 26
WRITE(6,1200) I
NFOUND=1
26 CONTINUE
C
IF(NFOUND.EQ.0) GO TO 27
WRITE(6,1300)
27 DO 28 I=1,NX
TEMP1(I,1)=EVR(I)
28 TEMP1(I,2)=EVI(I)
CALL MATUT(TEMP1,NX,2,8)
C
C COMPUTE AND PRINT TEMP1=TH*PSTAT*THT+R2
C
CALL MULT(B,S,TEMP1,NU,NX,NX,1,0)
CALL MULT(TEMP1,B,TEMP2,NU,NX,NU,0,0)
CALL MADD(TEMP2,Q2,TEMP1,NU,NU,0,0,0)
WRITE(6,1100)
CALL MATUT(TEMP1,NU,NU,8)
C
C WRITE K ON DISK
C
IT(1)=NX
IT(2)=NU
CALL ENTER(1,HNMK)
WRITE(1) (IT(I),I=1,10)
DO 30 I=1,NX
30 WRITE(1) (TEMP3(I,J),J=1,NU)
CALL CLOSE(1)
IF(NFOUND.EQ.1) IERR=9
C
RETURN
C
C ERROR CHECK
C
78 IF(IND-7) 80,82,80
80 IERR=7
RETURN
C
82 IERR=2
RETURN
C
84 IERR=3
RETURN
C
88 IERR=10

```

```
      RETURN
C
100  FORMAT(///6H KALFI,1X,3(A5,1X)///)
200  FORMAT(///7H P-STAT//)
300  FORMAT(///11H KALMANGAIN,4X,A5///)
400  FORMAT(//8H FI-K*TH//)
500  FORMAT(//23H EIGENVALUES OF FI-K*TH///)
600  FORMAT(2H ?)
700  FORMAT(///9H MATRIX P,5X,27HNOT THE STATIONARY SOLUTION,5X,
*    15,11H ITERATIONS///)
800  FORMAT(///9H MATRIX K,5X,27HNOT THE STATIONARY SOLUTION,5X,
*    15,11H ITERATIONS///)
900  FORMAT(1X,15,11H ITERATIONS/1X,22HTYPE YES1 TO CONTINUE,)
1000 FORMAT(37H YES2 TO CONTINUE WITH OUTPUT ELSE NO)
1100 FORMAT(///16H TH*PSTAT*THT+R2///)
1200 FORMAT(1X,10HEIGENVALUE,1X,12,1X,3HNOT,1X,5HFOUND)
1300 FORMAT(//1X//)
C
      END
```

SUBROUTINE EIGEN(IERR)

```

C
C
C COMPUTES AND PRINTS THE EIGENVALUES OF A MATRIX
C MAX (20*20) ON LINE PRINTER
C
C
C AUTHOR:      TORSTEN CROMNOW 1974-07-01
C
C SUBROUTINES REQUIRED
C   EIGP1
C   SCAL
C   HESOR
C   REALVE
C   COMPVE
C
C LOGICAL IFOUND
C
C   DIMENSION A(20,20),INDIC(20),EVR(20),EVI(20),
C *TEMP1(20,20),TEMP2(20,20),HNMA(2),KODE(10)
C
C
C   COMMON /CMD/ IP,IND,BUFF(16)
C   COMMON TEMP1,TEMP2,A
C
C   DATA HNMA /5H      ,4H BIN/
C
C   READ ARGUMENTS
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.1) GO TO 30
C   HNMA(1)=RRES
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.7) GO TO 36
C
C   READ MATRIX A
C
C   CALL FSTAT(1,HNMA,IFOUND)
C   IF(.NOT. IFOUND) GO TO 38
C   CALL SEEK(1,HNMA)
C   READ(1) (KODE(I),I=1,10)
C   NX=KODE(1)
C   IF(NX.NE.KODE(2)) GO TO 34
C   DO 10 I=1,NX
10  READ(1) (A(I,J),J=1,NX)
C   CALL CLOSE(1)
C
C   WRITE(6,100) HNMA(1)
C
C   FIND EIGENVALUES OF A
C
C   CALL EIGP1(A,NX,26.,EVR,EVI,TEMP1,TEMP2,INDIC)
C   NFOUND=0
C   DO 20 I=1,NX
C   IF(INDIC(I).NE.0) GO TO 20
C   WRITE(6,200) I
C   NFOUND=1
20  CONTINUE
C
C   WRITE EIGENVALUES ON LP
C
C   DO 22 I=1,NX

```

```
TEMP1(1,1)=EVR(1)
22 TEMP1(1,2)=EVI(1)
   IF(NFOUND.EQ.0) GO TO 24
   WRITE(6,300)
24 CALL MATUT(TEMP1,NX,2,8)
   IF(NFOUND.EQ.1) IERR=9
C
   RETURN
C
C   ERROR CHECK
C
30 IF(IND-7) 34,32,34
32 IERR=2
   RETURN
C
34 IERR=6
   RETURN
C
36 IERR=3
   RETURN
C
38 IERR=8
   RETURN
C
100 FORMAT(//15H EIGENVALUES OF,1X,A5//)
200 FORMAT(1X,10HEIGENVALUE,1X,12,1X,3HNOT,1X,5HFOUND)
300 FORMAT(//1X//)
C
   END
```

```

SUBROUTINE SYST(IERR)
C
C   GENERATES A SYSTEM INCLUDING 4 FILENAMES
C
C   AUTHOR:      TORSTEN CROMNOW 1974-07-03
C
C   SUBROUTINE REQUIRED
C     NONE
C
C   DIMENSION HNM(4),SNAME(2)
C
C   COMMON /CMD/ IP,IND,BUFF(16)
C
C   DATA SNAME /5H      ,4H SRC/
C
C   READ ARGUMENTS
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.1) GO TO 30
C   SNAME(1)=RRES
C
C   DO 10 I=1,4
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.1) GO TO 36
10  HNM(I)=RRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.7) GO TO 40
C
C   WRITE SYSTEM ON DISK
C
C   CALL ENTER(1,SNAME)
C   WRITE(1,100) (HNM(I),I=1,4)
C   CALL CLOSE(1)
C
C   WRITE SYSTEM ON LP
C
C   WRITE(6,200) SNAME(1)
C   WRITE(6,300) (HNM(I),I=1,4)
C
C   RETURN
C
C   ERROR CHECK
C
30  IF(IND-7) 34,32,34
32  IERR=2
C   RETURN
C
34  IERR=7
C   RETURN
C
36  IF(IND-7) 38,32,38
38  IERR=6
C   RETURN
C
40  IERR=3
C   RETURN
C
100  FORMAT(1X,A5)
200  FORMAT(///7H SYSTEM,1X,A5/)
300  FORMAT(8X,A5)
C
END

```



```

SUBROUTINE ZEROM(IERR)
C
C
C   GENERATES A ZERO-MATRIX (MAX 20*20)
C
C   AUTHOR:      TORSTEN CROMNOW 1974-07-06
C
C   SUBROUTINE REQUIRED
C     NONE
C
C   DIMENSION HNMZ(2),Z(20,20),IT(10)
C
C   COMMON /CMD/ IP,RRES,IRES,IND
C   COMMON Z
C
C   DATA HNMZ /5H      ,4H BIN/
C
C   READ ARGUMENTS
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.1) GO TO 30
C   HNHZ(1)=RRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.3) GO TO 36
C   IF(IRES.GT.20) GO TO 36
C   NR=IRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.3) GO TO 36
C   IF(IRES.GT.20) GO TO 36
C   NC=IRES
C
C   CALL RIFF(IP,RES,IRES,IND)
C   IF(IND.NE.7) GO TO 38
C
C   GENERATE ZERO-MATRIX
C
C   DO 10 I=1,NR
C   DO 10 J=1,NC
10  Z(I,J)=0.
C
C   PRINT ZERO-MATRIX ON LP
C
C   WRITE(6,100) HNMZ(1)
C   CALL MATUT(Z,NR,NC,8)
C
C   WRITE ZERO-MATRIX ON DISK
C
C   IT(1)=NR
C   IT(2)=NC
C   CALL ENTER(1,HNMZ)
C   WRITE(1) (IT(I),I=1,10)
C   DO 20 I=1,NR
20  WRITE(1) (Z(I,J),J=1,NC)
C   CALL CLOSE(1)
C
C   RETURN
C
C   ERROR CHECK

```

```
C
 30  IF(IND-7) 34,32,34
 32  IERR=2
     RETURN
C
 34  IERR=6
     RETURN
C
 36  IERR=4
     RETURN
C
 38  IERR=3
     RETURN
C
100  FORMAT(///7H MATRIX,1X,A5///)
C
     END
```

```

SUBROUTINE EXTEN(IERR)
C
C
C
C EXTENDS A GIVEN FEEDBACK MATRIX
C
C
C AUTHOR:      TORSTEN CROMNOW 1974-07-08
C
C SUBROUTINE REQUIRED
C   NONE
C
C LOGICAL IFOUND
C
C DIMENSION FBI(20,20),FBO(20,20),IT(10),HNM1(2),HNM2(2)
C
C COMMON /CMD/ IP,IND,BUFF(16)
C COMMON FBI,FBO
C
C DATA HNM1 /5H      ,4H BIN/
C DATA HNM2 /5H      ,4H BIN/
C
C READ ARGUMENTS
C
C CALL RIFF(IP,RRES,IRES,IND)
C IF(IND.NE.1) GO TO 30
C HNM1(1)=RRES
C
C CALL RIFF(IP,RRES,IRES,IND)
C IF(IND.NE.1) GO TO 30
C HNM2(1)=RRES
C
C CALL RIFF(IP,RRES,IRES,IND)
C IF(IND.NE.7) GO TO 36
C
C READ INPUT MATRIX
C
C CALL FSTAT(1,HNM1,IFOUND)
C IF(.NOT. IFOUND) GO TO 38
C CALL SEEK(1,HNM1)
C READ(1) (IT(1),I=1,10)
C NR=IT(1)
C IF(NR.GT.20) GO TO 38
C NC=IT(2)
C IF(NC.GT.18) GO TO 38
C DO 10 I=1,NR
10 READ(1) (FBI(I,J),J=1,NC)
C CALL CLOSE(1)
C
C WRITE(6,200) HNM1(1),HNM2(1)
C
C WRITE INPUT MATRIX ON DISK AND LP WITH IT(3)=2
C
C IT(3)=2
C CALL ENTER(1,HNM1)
C WRITE(1) (IT(1),I=1,10)
C DO 11 I=1,NR
11 WRITE(1) (FBI(I,J),J=1,NC)
C CALL CLOSE(1)
C
C WRITE(6,100) HNM1(1)
C CALL MATUT(FBI,NR,NC,8)

```

```

C
C   GENERATE OUTPUT MATRIX
C
      DO 12 I=1, NR
12    FBO(I,1)=0.
        J=NC+2
      DO 14 I=1, NR
14    FBO(I,J)=0.
        K=NC+1
      DO 16 I=1, NR
      DO 16 L=2, K
        N=L-1
16    FBO(I,L)=FBI(I,N)
C
C   PRINT OUTPUT MATRIX
C
      WRITE(6,100) HNM2(1)
      CALL MATUT(FBO, NR, J, 8)
C
C   WRITE OUTPUT MATRIX ON DISK
C
      IT(2)=J
      CALL ENTER(1, HNM2)
      WRITE(1) (IT(I), I=1, 10)
      DO 20 I=1, NR
20    WRITE(1) (FBO(I,K), K=1, J)
      CALL CLOSE(1)
C
      RETURN
C
C   ERROR CHECK
C
30    IF(IND-7) 34, 32, 34
32    IERR=2
      RETURN
C
34    IERR=6
      RETURN
C
36    IERR=3
      RETURN
C
38    CALL CLOSE(1)
      IERR=8
      RETURN
C
100   FORMAT(////7H MATRIX, 1X, A5////)
200   FORMAT(////6H EXTEN, 1X, A5, 1X, A5////)
C
      END

```

```

SUBROUTINE MATOP(IERR)
C
C PERFORMS ARITHMETIC OPERATIONS ON MATRICES
C IF OPERATION IS + OR - HNM1 MAY BE CALLED NUL
C
C
C AUTHOR:      TORSTEN CROMNOW 1974-08-14
C
C SUBROUTINE REQUIRED
C   NONE
C
C
C REAL NUL
C LOGICAL IFOUND
C
C
C DIMENSION HNM1(2),HNM2(2),HNM3(2),OP(4),
C *KODE(10),A(20,20),B(20,20),C(20,20)
C
C COMMON /CMD/ IP,IND,BUFF(16)
C COMMON A,B,C
C
C DATA HNM1 /5H      ,4H BIN/
C DATA HNM2 /5H      ,4H BIN/
C DATA HNM3 /5H      ,4H BIN/
C DATA OP /1H*,1H+,1H-,2HTR/
C DATA NUL /3HNUL/
C
C READ ARGUMENTS
C
C CALL RIFF(IP,RRES,IRES,IND)
C IF(IND.NE.1) GO TO 80
C HNM1(1)=RRES
C
C CALL RIFF(IP,RRES,IRES,IND)
C IF(IND.NE.1) GO TO 88
C OPER=RRES
C
C IF(OPER.EQ.OP(4)) GO TO 10
C HNM2(1)=RRES
C
C CALL RIFF(IP,RRES,IRES,IND)
C IF(IND.NE.6) GO TO 92
C OPER=RRES
C
C 10 CALL RIFF(IP,RRES,IRES,IND)
C IF(IND.NE.1) GO TO 80
C HNM3(1)=RRES
C
C CALL RIFF(IP,RRES,IRES,IND)
C IF(IND.NE.7) GO TO 90
C
C FIND SPECIFIED OPERATION
C
C DO 12 I=1,4
C IF(OPER.EQ.OP(I)) GO TO (20,30,32,50),I
C 12 CONTINUE
C IERR=1
C RETURN
C
C MULT
C
C 20 CALL FSTAT(1,HNM1,IFOUND)
C IF(.NOT. IFOUND) GO TO 86

```

```

CALL FSTAT(1,HNM2,IFOUND)
IF(.NOT. IFOUND) GO TO 86
C
C   GET HNM1 AND HNM2 FROM DISK
C
CALL SEEK(1,HNM1)
READ(1) (KODE(1),I=1,10)
NX1=KODE(1)
NU1=KODE(2)
IF(NX1.GT.20.OR.NU1.GT.20) GO TO 84
DO 22 I=1,NX1
22  READ(1) (A(I,J),J=1,NU1)
CALL CLOSE(1)
C
CALL SEEK(1,HNM2)
READ(1) (KODE(1),I=1,10)
IF(KODE(1).NE.NU1) GO TO 84
NX2=KODE(2)
IF(KODE(2).GT.20) GO TO 84
DO 24 I=1,NU1
24  READ(1) (B(I,J),J=1,NX2)
CALL CLOSE(1)
C
C   C=A*B
C
CALL MULT(A,B,C,NX1,NU1,NX2,0,0)
NU=NX2
NX=NX1
KODE(1)=NX
KODE(2)=NU
GO TO 60
C
C   ADD AND SUB
C
30  IOPKOD=0
GO TO 34
32  IOPKOD=1
C
34  CALL FSTAT(1,HNM2,IFOUND)
IF(.NOT. IFOUND) GO TO 86
C
C   GET HNM2 FROM DISK
C
CALL SEEK(1,HNM2)
READ(1) (KODE(1),I=1,10)
NX=KODE(1)
NU=KODE(2)
IF(NX.GT.20.OR.NU.GT.20) GO TO 84
DO 36 I=1,NX
36  READ(1) (B(I,J),J=1,NU)
CALL CLOSE(1)
C
C   GET HNM1 FROM DISK
C
IF(HNM1(1).NE.NUL) GO TO 40
DO 38 I=1,NX
DO 38 J=1,NU
38  A(I,J)=0.
GO TO 44
C
40  CALL FSTAT(1,HNM1,IFOUND)
IF(.NOT. IFOUND) GO TO 86
C
CALL SEEK(1,HNM1)

```

```

      READ(1) (KODE(1),I=1,10)
      IF(KODE(1).NE.NX.OR.KODE(2).NE.NU) GO TO 84
      DO 42 I=1,NX
42    READ(1) (A(I,J),J=1,NU)
      CALL CLOSE(1)
C
C    C=A+-B
C
44    CALL MADD(A,B,C,NX,NU,0,0,IOPKOD)
      GO TO 60
C
C    TR
C
50    CALL FSTAT(1,HNM1,IFOUND)
      IF(.NOT.IFOUND) GO TO 86
C
C    GET HNM1 FROM DISK
C
      CALL SEEK(1,HNM1)
      READ(1) (KODE(1),I=1,10)
      NX=KODE(1)
      NU=KODE(2)
      IF(NX.GT.20.OR.NU.GT.20) GO TO 84
      DO 52 I=1,NX
52    READ(1) (A(I,J),J=1,NU)
      CALL CLOSE(1)
C
C    TRANSPOSE A    C=TR A
C
      DO 54 I=1,NX
54    CALL RMOVE(A(1,I),1,C(1,1),20,NX)
      NX1=NX
      NX=NU
      NU=NX1
      KODE(1)=NX
      KODE(2)=NU
C
C    WRITE COMMAND ON LP
C
      WRITE(6,100) HNM1(1),HNM3(1)
      GO TO 62
60    WRITE(6,200) HNM1(1),HNM2(1),OPER,HNM3(1)
C
C    WRITE C ON DISK
C
62    CALL ENTER(1,HNM3)
      WRITE(1) (KODE(1),I=1,10)
      DO 64 I=1,NX
64    WRITE(1) (C(I,J),J=1,NU)
      CALL CLOSE(1)
C
C    WRITE C ON LP
C
      WRITE(6,300) HNM3(1)
      CALL MATUT(C,NX,NU,8)
C
      RETURN
C
C    ERROR CHECK
C
80    IF(IND-7) 84,82,84
82    IERR=2
      RETURN
C

```

```
84  IERR=6  
    RETURN  
C  
86  IERR=8  
    RETURN  
C  
88  IERR=1  
    RETURN  
C  
90  IERR=3  
    RETURN  
C  
92  IERR=1  
    RETURN  
C  
100 FORMAT(////1X,5HMATOP,1X,A5,1X,2HTR,1X,A5////)  
200  FORMAT(////1X,5HMATOP,1X,2(A5,1X),A2,1X,A5////)  
300  FORMAT(////1X,6HMATRIX,1X,A5////)  
C  
    END
```



```

SUBROUTINE DEFI(IERR)
C
C   CHANGES THE VALUE OF A SPECIFIED PARAMETER
C
C   AUTHOR:      TORSTEN CROMNOW 1974-08-14
C
C   SUBROUTINE REQUIRED
C       NONE
C
C   DIMENSION PAR(13),HNM(11)
C
C   COMMON /CMD/ IP,IND,BUFF(16)
C   COMMON /DEF/ NMAX,EPS
C   COMMON /DEFHR/ RL,TD,THL,T1,T2,G1,G2,ZETA1,ZETA2,
C   *W01,W02
C
C   EQUIVALENCE (RL,HNM(1))
C
C   DATA PAR /4HNM,3HEPS,2HRL,2HTD,3HHTL,2HT1,2HT2,
C   *2HG1,2HG2,5HZETA1,5HZETA2,3HW01,3HW02/
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.1) GO TO 50
C   PARAM=RRES
C
C   DO 10 I=1,13
C   IF(PARAM.EQ.PAR(I)) GO TO (20,30,40,40,40,40,40,
C   *40,40,40,40,40,40),I
10  CONTINUE
C   GO TO 54
C
C   CHANGE NMAX
C
C   20  CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.3) GO TO 56
C   IF(IRES.LE.0) GO TO 62
C   IVALUE=IRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.7) GO TO 60
C   NMAX=IVALUE
C   WRITE(6,100) IVALUE
C   RETURN
C
C   CHANGE EPS
C
C   30  CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.4) GO TO 58
C   IF(RRES.LE.0.) GO TO 66
C   VALUE=RRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.7) GO TO 60
C   EPS=VALUE
C   WRITE(6,200) VALUE
C   RETURN
C
C   CHANGE PARAMETER OF HROD
C
C   40  CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.4) GO TO 58
C   IF(RRES.LE.0) GO TO 66

```

```
VALUE=RRES
C
CALL RIFF(IP,RRES,IRES,IND)
IF(IND.NE.7) GO TO 60
HNM(1-2)=VALUE
WRITE(6,300) PAR(1),VALUE
RETURN
C
C ERROR CHECK
C
50 IF(IND-7) 54,52,54
52 IERR=2
RETURN
C
54 IERR=11
RETURN
C
56 IF(IND-7) 62,64,62
C
58 IF(IND-7) 66,64,66
C
60 IERR=3
RETURN
C
62 IERR=4
RETURN
C
64 IERR=2
RETURN
C
66 IERR=5
RETURN
C
100 FORMAT(////1X,4HDEFI,1X,4HNMAX,1X,1I6////)
200 FORMAT(////1X,4HDEFI,1X,3HEPS,1X,1E10,3////)
300 FORMAT(////1X,4HDEFI,1X,A5,1X,1E15,6////)
C
END
```

APPENDIX C

TASK DB

\*\*\*\*\*

INTERACTIVE PROGRAM FOR DEAD-BEAT CONTROL  
OF A DIFFUSION PROCESS

DESCRIPTION OF COMMAND STRUCTURE:

\*\*\*\*\*

INIT NUSA GAIN U1 U2

NUSA - NUMBER OF SAMPLING POINTS (MAX 46)  
GAIN - GAIN OF INTERFACE (ANALOGUE COMPUTER)  
U1,U2 - INITIAL BOUNDARY VALUES OF PROFILE (MAX+-5)

FBCK FNAME

FNAME - NAME OF FEEDBACK MATRIX

AXIS IPD YMIN YMAX (REZERO REGAIN)

IPD - PLOT DENSITY (POINTS/SAMPLINGINTERVAL)  
(MAX 2 POINTS/SEC)

YMIN,YMAX - MIN-AND MAX-VALUE OF Y-AXIS

REZERO - ZERO POINT OF X-AXIS FOR RESIDUALS

REGAIN - GAIN FACTOR OF ERRORS WHEN PLOTTED

EXEC U1 U2

U1,U2 - TERMINAL BOUNDARY VALUES OF PROFILE (MAX+-5)

PRINT

STOP

NB. REZERO AND REGAIN SHOULD ONLY BE SPECIFIED WHEN A  
KALMAN FILTER IS USED TO ESTIMATE THE STATE OF THE  
DIFFUSION PROCESS

SUBROUTINES REQUIRED

INIDB

```

C                (RIFF)                C2.
C          FBCK
C          RDMTX
C          (RIFF)
C          AXEDB
C          (AXIS)
C          (RSXPLT)
C          (SCALE)
C          EXEDB
C          DBR
C          (ISENRX)
C          PLTDB
C          (LINE)
C          (SCALE)
C          (RIFF)
C          (RSXPLT)
C          PRTDB
C          ERRDB
C
C          DIMENSION COM(6)
C
C          COMMON /CMD/  IP,IND,BUFF(16)
C          COMMON /DAT/  ITIME(6),AMONTH
C          COMMON /PAR/  NUSA,UI(2),UR(2),XR(17)
C          COMMON /FBK/  NR,NC,NZ,ITICK,FNAME,F(17,17)
C          COMMON /PLT/  IPLT,IPD,YMIN,YMAX,REZERO,REGAIN,IX(921)
C          COMMON /UY/  UP(2),YP(9)
C          COMMON /IOCOM/  ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),
*IFLOVA(48),USER(36)
C          COMMON /RTCOM/  IO(512)
C
C          DATA COM/4HINIT,4HFBCK,4HAXIS,4HEXEC,5HPRINT,4HSTOP/
C
C          INITIATE
C
C          LUI=12
C          LUO=13
C          LUN=40
C          IPLT=0
C          NUSA=0
C
C          READ A COMMAND FROM TT0
C
C          IERR=0
C          CALL RLINE(LUO,LUI,BUFF,1)
C          IP=1
C
C          CALL RIFF(IP,RRES,IRES,IND)
C          IF(IND.EQ.1) GO TO 4
C          IF(IND-7) 8,2,8
C
C          DO 6 I=1,6
C          IF(RRES.EQ.COM(I)) GO TO(10,20,30,40,50,60),I
C          CONTINUE
C
C          IERR=1
C          GO TO 90
C
C          INIT
C
C          CALL INIDB(IERR)
C          IF(IERR.EQ.0) GO TO 2
C          GO TO 90

```

```
C
C   FBCK
C
20  CALL FBCK(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 90
C
C   AXIS
C
30  CALL AXEDB(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 90
C
C   EXEC
C
40  CALL EXEDB(IERR)
    IF(IERR.EQ.0) GO TO 2
    GO TO 90
C
C   PRINT
C
50  CALL RIFF(IP,RRES,IRES,IND)
    IF(IND.EQ.7) GO TO 52
    IERR=3
    GO TO 90
52  CALL PRIDB
    GO TO 2
C
C   STOP
C
60  IF(IPLT) 62,64,62
62  CALL PLOTS(LUN,2,IEV)
64  CALL EXIT
C
C   ERROR CHECK
C
90  IF(IERR) 92,2,92
92  CALL ERRDB(IERR)
    GO TO 2
C
    END
```

```
SUBROUTINE ERRDB(IERR)
```

```
C  
C  
C  
C  
C  
C  
C  
C  
C  
C
```

```
PRINTS OUT ERROR INFORMATION  
IERR - ERROR SWITCH
```

```
AUTHOR: TORSTEN CROMNOW 1974-09-22
```

```
SUBROUTINE REQUIRED  
NONE
```

```
LU0=13  
NERR=14  
DO 10 I=1,NERR  
IF(I.EQ.IERR) GO TO(101,102,103,104,105,106,  
*107,108,109,110,111,112,113,114),I
```

```
10 CONTINUE  
RETURN
```

```
C
```

```
101 WRITE(LU0,201)  
RETURN
```

```
102 WRITE(LU0,202)  
RETURN
```

```
103 WRITE(LU0,203)  
RETURN
```

```
104 WRITE(LU0,204)  
RETURN
```

```
105 WRITE(LU0,205)  
RETURN
```

```
106 WRITE(LU0,206)  
RETURN
```

```
107 WRITE(LU0,207)  
RETURN
```

```
108 WRITE(LU0,208)  
RETURN
```

```
109 WRITE(LU0,209)  
RETURN
```

```
110 WRITE(LU0,210)  
RETURN
```

```
111 WRITE(LU0,211)  
RETURN
```

```
112 WRITE(LU0,212)  
RETURN
```

```
113 WRITE(LU0,213)  
RETURN
```

```
114 WRITE(LU0,214)  
RETURN
```

```
C
```

```
201 FORMAT(16H ILLEGAL COMMAND)  
202 FORMAT(18H TOO FEW ARGUMENTS)  
203 FORMAT(19H TOO MANY ARGUMENTS)  
204 FORMAT(12H BAD INTEGER)  
205 FORMAT(11H BAD NUMBER)  
206 FORMAT(14H BAD FILE NAME)  
207 FORMAT(29H BAD VALUE OF SAMPLING POINTS)  
208 FORMAT(12H BAD PROFILE)  
209 FORMAT(15H FILE NOT FOUND)  
210 FORMAT(17H BAD PLOT DENSITY)  
211 FORMAT(19H BAD SCALE LIMIT(S))  
212 FORMAT(25H NO FEEDBACK MATRIX GIVEN)  
213 FORMAT(17H PLEASE DRAW AXIS)  
214 FORMAT(18H BAD VALUE OF GAIN)
```

```
C
```

```
END
```

```

SUBROUTINE INIDB(IERR)
C
C   INITIALIZES THE PROFILE OF THE DIFFUSION PROCESS
C
C   IERR      - ERROR SWITCH
C
C   AUTHOR:   TORSTEN CROMNOW  1974-04-24
C
C   SUBROUTINE REQUIRED
C     (RIFF)
C
C   DIMENSION U(2)
C
C   COMMON /CMD/ IP,IND,BUFF(16)
C   COMMON /PAR/ NUSA,UI(2),UR(2),XR(17)
C   COMMON /UY/ UP(2),YP(9)
C   COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),
*IFLOVA(48),USER(36)
C
C   EQUIVALENCE (NVAI,ISYSA(5)),(NVA0,ISYSA(6)),(U(1),IFLOVA(33)),
*(GAIN,USER(1)),(ISWK,ISYSA(23))
C
C   INITIATE
C
C   NVAI=9
C   NVA0=2
C
C   READ ARGUMENTS
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-3) 40,10,40
10  NUSA=IRES
C   IF(NUSA.GE.0.AND.NUSA.LT.47) GO TO 12
C   IERR=7
C   RETURN
C
C   12  CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-4) 50,14,50
14  GAIN=RRES
C   IF(GAIN.NE.0) GO TO 15
C   IERR=14
C   RETURN
C
C   15  DO 20 I=1,2
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-4) 50,16,50
16  UI(I)=RRES
C   IF(ABS(UI(I)-U(I)/GAIN).LE.5..AND.ABS(UI(I)).LE.5.) GO TO 18
C   IERR=8
C   RETURN
18  U(I)=GAIN*UI(I)
20  UP(I)=UI(I)
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.EQ.7) GO TO 30
C   IERR=3
C   RETURN
C

```



```
C      INITIALIZE PROFILE OF DIFFUSION PROCESS
C
C 30     IF(ISWK.EQ.1) RETURN
        CALL REQST(4HALIO,150)
        RETURN
C
C 40     IF(IND-7)44,42,44
C 42     IERR=2
        RETURN
C
C 44     IERR=4
        RETURN
C
C 50     IF(IND-7)52,42,52
C 52     IERR=5
        RETURN
C
C      END
```

C7.

```

SUBROUTINE FBCK(IERR)
C
C
C   READS A FEEDBACK MATRIX FROM DT1
C
C   IERR      - ERROR SWITCH
C
C   AUTHOR:   TORSTEN CROMNOW 1974-04-24
C
C   SUBROUTINES REQUIRED
C     RDMTX
C     (RIFF)
C
C   DIMENSION KODE(10)
C
C   COMMON /CMD/ IP,IND,BUFF(16)
C   COMMON /FBK/ NR,NC,NZ,ITICK,FNAME,F(17,17)
C
C   READ ARGUMENTS
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-1) 30,10,30
10  FNAME=RRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.EQ.7) GO TO 15
C   IERR=3
C   RETURN
C
C   READS A FEEDBACK MATRIX FROM DT1
C
C   15  LUN=33
C   CALL RDMTX(FNAME,LUN,KODE,F,IERR)
C   IF(IERR) 22,20,22
20  NR=KODE(1)
C   NC=KODE(2)
C   NZ=KODE(3)
C   ITICK=KODE(4)
C   RETURN
C
C   22  IERR=9
C   RETURN
C
C   30  IF(IND-7) 34,32,34
32  IERR=2
C   RETURN
C
C   34  IERR=6
C   RETURN
C
C   END
```



```

SUBROUTINE AXEDB(IERR)
C
C READS PLOTTING ARGUMENTS AND DRAWS AXIS ON VP
C
C IERR      - ERROR SWITCH
C
C AUTHOR:    TORSTEN CROMNOW  1974-05-04
C
C SUBROUTINE REQUIRED
C   RSXPLT
C   (AXIS)
C   (SCALE)
C
C DIMENSION X(921),FNAMK(4),IT(4),AMON(12)
C
C COMMON /CMD/  IP,IND,BUFF(16)
C COMMON /DAT/  ITIME(6),AMONTH
C COMMON /PAR/  NUSA,UI(2),UR(2),XR(17)
C COMMON /FBK/  NR,NC,NZ,ITICK,FNAME,F(17,17)
C COMMON /PLT/  IPLT,IPD,YMIN,YMAX,REZERO,REGAIN,IX(921)
C COMMON /IOCOM/ IDUM(184),USER(36)
C
C EQUIVALENCE (ITICKK,IDUM(19)),(IPER,IDUM(20)),
* (ISWK,IDUM(23)),(FNAMK(1),USER(3))
C
C DATA IT /10,1,0,1/
C DATA AMON /3HJAN,3HFEB,4HMARS,5HAPRIL,3HMAJ,
* 4HJUNI,4HJULY,3HAUG,4HSEPT,3HOCT,3HNOV,3HDEC/
C
C IF(NUSA-1) 2,2,4
C 2 IERR=7
C RETURN
C 4 IF(ITICK.GE.25.OR.ITICK.LE.45000) GO TO 10
C IERR=12
C RETURN
C
C READ ARGUMENTS
C
C 10 CALL RIFF(IP,RRES,IRES,IND)
C IF(IND-3) 60,11,60
C 11 IPD=IRES
C
C IF(IPD.LT.1.OR.IPD*NUSA.GT.920) GO TO 12
C IF(ITICK/IPD.LT.25.OR.ITICK.NE.IPD*(ITICK/IPD)) GO TO 12
C IF(ISWK.EQ.0.OR.ITICKK.EQ.(ITICK/IPD)*
* (ITICKK/(ITICK/IPD))) GO TO 14
C
C 12 IERR=10
C RETURN
C
C 14 IF(ISWK) 16,20,16
C 16 CALL UNFIX(4HALIO)
C CALL CANCEL(4HALIO)
C IT(3)=ITICK/IPD
C IPER=ITICKK/(ITICK/IPD)
C CALL FIX(4HALIO)
C CALL RUN(4HALIO,IT,150)
C
C 20 CALL RIFF(IP,RRES,IRES,IND)
C IF(IND-4) 70,22,70
C 22 YMIN=RRES
C IF(ABS(YMIN).LE.10.) GO TO 24
C IERR=11

```

```

RETURN
C
24 CALL RIFF(IP,RRES,IRES,IND)
   IF(IND-4) 70,26,70
26 YMAX=RRES
   IF(ABS(YMAX).LE.10.,.AND.(YMAX-YMIN).GT.0.) GO TO 30
   IERR=11
   RETURN
C
30 IF(ISWK) 40,40,32
32 CALL RIFF(IP,RRES,IRES,IND)
   IF(IND-4) 70,34,70
34 REZERO=RRES
   IF(ABS(REZERO).LE.10.) GO TO 36
   IERR=11
   RETURN
C
36 CALL RIFF(IP,RRES,IRES,IND)
   IF(IND-4) 70,38,70
38 REGAIN=RRES
C
40 CALL RIFF(IP,RRES,IRES,IND)
   IF(IND-7) 42,50,42
42 IERR=3
   RETURN
C
C DRAW AXIS ON VP
C
50 LUN=40
   IPLT=1
   TSAMP=(ITICK/IPD)/50.
   XMAX=(NUSA-1)*ITICK/50.
   CALL PLOTS(LUN,0,IEV)
   CALL DATE(ETIME)
   ISL=ETIME(1)
   ETIME(1)=ETIME(2)
   AMONTH=AMON(ISL)
   WRITE(LUN,300) ETIME(1),AMONTH,(ETIME(1),I=3,6),FNAME
   IF(ISWK.EQ.0) GO TO 52
   WRITE(LUN,400) (FNAMK(I),I=1,4)
52 NI=(NUSA-1)*IPD+1
   DO 54 I=1,NI
54 X(I)=(I-1)*TSAMP
C
   CALL SCALE(X(1),IX(1),NI,22.,0.,XMAX,SMINX,DSX,1)
   ASSIGN 100 TO IFMT
   CALL AXIS(0,0,22.,IFMT,8,0.,SMINX,DSX)
C
   CALL SCALE(REZERO,IZERO,1,17.,YMIN,YMAX,SMINY,DSY,1)
   IF(ISWK.EQ.0) GO TO 56
   CALL AXIS(0,IZERO,22.,IFMT,8,0.,SMINX,DSX)
56 ASSIGN 200 TO IFMT
   CALL AXIS(0,0,-17.,IFMT,7,90.,SMINY,DSY)
C
RETURN
C
60 IF(IND-7) 64,62,64
62 IERR=2
   RETURN
C
64 IERR=4
   RETURN
C
70 IF(IND-7) 72,62,72

```

C11.

SUBROUTINE EXEDB(IERR)

PERFORMS DEAD-BEAT CONTROL, PLOTS INPUTS AND OUTPUTS ON  
DISPLAY AND STORES PROFILE IN RTCOM

IERR - ERROR SWITCH

AUTHOR: TORSTEN CROMNOW 1974-05-10

SUBROUTINES REQUIRED

PLTDB  
(LINE)  
(SCALE)  
DBR  
RSXPLT  
ISENRX  
(RIFF)

DIMENSION IT(4),U(8),Y(9)

COMMON /CMD/ IP,IND,BUFF(16)  
COMMON /PAR/ NUSA,UI(2),UR(2),XR(17)  
COMMON /FBK/ NR,NC,NZ,ITICK,FNAME,F(17,17)  
COMMON /PLT/ IPLT,IPD,YMIN,YMAX,REZERO,REGAIN,IX(921)  
COMMON /UY/ UP(2),YP(9)  
COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),  
\*IFLOVA(48),USER(36)

EQUIVALENCE (NE,ISYSA(18)),(ISWD,ISYSA(22)),(ISWK,ISYSA(23)),  
\*(IEV1,ICOMVA(41)),(IEV3,ICOMVA(43)),(IEV4,ICOMVA(44)),  
\*(IEV5,ICOMVA(45)),(U(1),IFLOVA(33)),(Y(1),IFLOVA(1)),  
\*(GAIN,USER(1))

DATA IT/10,1,0,1/

IF(IPLT.NE.0) GO TO 2  
IERR=13  
RETURN

INITIATE

LUN=40  
IPER=IPD  
NS=0  
IC1=0  
IC2=0  
IC3=IPER  
UI(1)=UP(1)  
UI(2)=UP(2)  
ISW1=0  
IF(ISWK.EQ.1) GO TO 6  
IEV1=0  
IEV4=0  
IT(3)=ITICK/IPD

READ ARGUMENTS

DO 12 I=1,2  
CALL RIFF(IP,RRES,IRES,IND)  
IF(IND=4) 20,10,20  
UR(1)=RRES

```

IF (ABS(UR(1)-UI(1)),LE,5.C12 AND ABS(UR(1)),LE,5.) GO TO 12
IERR=8
RETURN
12 CONTINUE
C
CALL RIFF(IP,RRES,IRES,IND)
IF(IND.EQ.7) GO TO 30
IERR=3
RETURN
C
20 IF(IND-7) 24,22,24
22 IERR=2
RETURN
C
24 IERR=5
RETURN
C
30 IF(ISWK) 34,32,34
32 CALL FIX(4HALIO)
CALL RUN(4HALIO,IT,150)
GO TO 40
34 CALL WAITFR(IEV4)
ISWD=1
C
C MAIN LOOP
C
40 IF(NS-NUSA) 42,90,90
42 CALL WAITFR(IEV1)
IEV1=0
C
C SCALE PROCESS OUTPUTS
C
DO 44 I=1,9
44 YP(I)=Y(I)/GAIN
C
IF(IC3-IPER) 70,50,70
C
C GENERATE REFERENCE TRAJECTORY OF PROCESS
C
50 IF(ISWK) 52,56,52
52 IF(NE.EQ.7) GO TO 56
DO 54 I=1,NC
54 XR(I)=UR(1)+(I-1)*(UR(2)-UR(1))/(NC-1)
GO TO 60
56 DO 58 I=1,NC
58 XR(I)=UR(1)+I*(UR(2)-UR(1))/(NC+1)
C
C COMPUTE OUTPUTS OF DEAD-BEAT REGULATOR
C AND STORE PROCESS INPUTS AND OUTPUTS IN RTCOM
C
60 CALL DBR(IC1,IC2)
NS=NS+1
IC3=0
IF(IC1,EQ.NR) IC1=0
C
C LET TASK SKEST RUN
C
70 IF(ISWK) 72,76,72
72 IEV3=1
IF(NS.EQ.1) IEV1=1
74 CALL WAIT
IF(IEV3,EQ.1) GO TO 74
C
76 CALL PLTDB(IC3,ISW1)

```

IC3=IC3+1

C13.

C

C GET NEW BOUNDARY VALUES OF PROFILE  
C STORED IN FLA0 7 AND 8

C

IF(IEV5) 80,82,80  
80 IEV5=0  
UR(1)=U(7)  
UR(2)=U(8)

C

C IF DATASWITCH 0 IS SET THE EXPERIMENT IS STOPPED

C

82 IF(ISENSW(0)) GO TO 90

C

GO TO 40

C

C READY

C

90 IF(ISWK) 94,92,94  
92 CALL UNFIX(4HAL10)  
CALL CANCEL(4HAL10)  
94 CALL PLOTS(LUN,2,IEV)  
IPLT=0  
IEV3=1  
ISWD=0

C

RETURN

C

END



SUBROUTINE DBR(IC1,IC2)

PERFORMS DEAD-BEAT CONTROL AND STORES PROCESS  
INPUTS AND OUTPUTS IN RTCOM

AUTHOR: TORSTEN CROMNOW 1974-06-30

SUBROUTINES REQUIRED  
NONE

DIMENSION U(2),XE(17)

COMMON /PAR/ NUSA,UI(2),UR(2),XR(17)  
COMMON /FBK/ NR,NC,NZ,ITICK,FNAME,F(17,17)  
COMMON /UY/ UP(2),YP(9)  
COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),  
\*IFLOVA(48),USER(36)  
COMMON /RTCOM/ IO(512)

EQUIVALENCE (ISWK,ISYSA(23)),(U(1),IFLOVA(33)),  
\*(GAIN,USER(1)),(XE(1),USER(7))

INITIATE

NC1=9  
NC2=9+NZ

COMPUTE DEAD-BEAT CONTROL

IF (ISWK) 20,10,20  
10 DO 12 I=1,NZ  
I1=IC1+1  
UP(I)=SCAPRO(F(I1,1),17,YP(1),1,NC)-  
\*SCAPRO(F(I1,1),17,XR(1),1,NC)+UR(I)  
12 U(I)=GAIN\*UP(I)  
GO TO 30

20 DO 22 I=1,NZ  
I1=IC1+1  
UP(I)=SCAPRO(F(I1,1),17,XE(1),1,NC)-  
\*SCAPRO(F(I1,1),17,XR(1),1,NC)+UR(I)  
22 U(I)=GAIN\*UP(I)

STORE PROCESS INPUTS AND OUTPUTS IN RTCOM

30 DO 32 I=1,NZ  
I1=IC2+1  
32 IO(I1)=25000.\*UP(I)  
DO 34 I=1,NC1  
I1=I+IC2+NZ  
34 IO(I1)=25000.\*YP(I)

INCREMENT COUNTERS

IC1=IC1+NZ  
IC2=IC2+NC2

RETURN

END

```

SUBROUTINE PLTDB(ICNT,ISW)
C
C PLOTS INPUTS AND OUTPUTS AND IF TASK SKEST IS ACTIVE
C ALSO THE ESTIMATION ERRORS OF A PROCESS ON VP
C
C ICNT - COUNTER
C ISW - PUT ISW=0 TO INITIALIZE THE PLOTTING
C
C AUTHOR: TORSTEN CROMNOW 1974-05-04
C
C SUBROUTINES REQUIRED
C (SCALE)
C (LINE)
C
C DIMENSION E(9),ES(9),IXE(2),UYE(20),IUYE(20),IUYEP(2,20)
C
C COMMON /FBK/ NR,NC,NZ,ITICK,FNAME,F(17,17)
C COMMON /PLT/ IPLT,IPD,YMIN,YMAX,REZERO,REGAIN,IX(921)
C COMMON /UY/ UP(2),YP(9)
C COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),
*IFLOVA(48),USER(36)
C
C EQUIVALENCE (NE,ISYSA(18)),(ISWK,ISYSA(23)),(IPLE,ISYSA(24)),
*(E(1),USER(25))
C
C SCALE INPUTS,OUTPUTS AND ERRORS
C
C NUY=NZ+9
C NUYE=NZ+NE+9
C ICNTRL=0
C IF(IPD.EQ.1) ICNTRL=1
C IF(ICNT.EQ.1) ICNTRL=2
C
C DO 2 I=1,NE
2 ES(I)=REZERO+REGAIN*E(I)
C
C CALL MOVE(UP(1),UYE(1),NZ+NZ)
C CALL MOVE(YP(1),UYE(NZ+1),18)
C CALL MOVE(ES(1),UYE(NZ+10),NE+NE)
C CALL SCALE(UYE(1),IUYE(1),NUYE,17.,YMIN,YMAX,SMIN,DS,1)
C
C IF(ISW) 20,10,20
C
C INITIATE PLOTTING PROCEDURE
C
C 10 ISW=1
C IC1=0
C IXE(2)=IX(1)
C DO 12 I=1,NUYE
12 IUYEP(2,I)=IUYE(I)
C GO TO 50
C
C PLOT INPUTS AND OUTPUTS
C
C 20 DO 22 I=1,NUY
C I1=I-1
C IF(I1.GT.7) I1=I1-8
C IF(ICNT.EQ.0.AND.I.LE.NZ) GO TO 22
C IUYEP(1,I)=IUYEP(2,I)
C IUYEP(2,I)=IUYE(I)
22 CALL LINE(IX(IC1),IUYEP(1,I),2,1,ICNTRL,I1)
C

```

C16.

```
      IF(ICNT) 30,24,30
24    DO 26 I=1,NZ
26    IUYEP(2,I)=IUYE(I)
C
30    IF(ISWK) 32,50,32
32    IF(IPL-1) 50,40,50
C
C    PLOT ERRORS
C
40    ICNTRL=1
      IXE(1)=IXE(2)
      IXE(2)=IX(IC1+1)
      DO 42 I=1,NE
        I1=I-1
        IF(I1.GT.7) I1=I1-8
        I2=NUY+I
        IUYEP(1,I2)=IUYEP(2,I2)
        IUYEP(2,I2)=IUYE(I2)
42    CALL LINE(IXE(1),IUYEP(1,I2),2,1,ICNTRL,I1)
C
50    IC1=IC1+1
C
      RETURN
C
      END
```

## SUBROUTINE PRITDB

```

C
C
C PRINTS RESULTS OF A DEAD-BEAT CONTROL EXPERIMENT
C
C AUTHOR:      TORSTEN CROMNOW    1974-04-26
C
C SUBROUTINE REQUIRED
C      NONE
C
C DIMENSION RIO(11),FNAMK(4)
C
C COMMON /DAT/  ITIME(6),AMONTH
C COMMON /PAR/  NUSA,UI(2),UR(2),XR(17)
C COMMON /FBK/  NR,NC,NZ,ITICK,FNAME,F(17,17)
C COMMON /IOCOM/ IDUM(184),USER(36)
C COMMON /RTCOM/ IO(512)
C
C EQUIVALENCE (ISWK,IDUM(23)),(FNAMK(1),USER(3))
C
C LUO=36
C NC1=9+NZ
C TSAMP=ITICK/50.
C
C WRITE(LUO,100) ITIME(1),AMONTH,(ITIME(I),I=3,6)
C WRITE(LUO,102) TSAMP,NC
C WRITE(LUO,110) FNAME
C
C NK=NR/NZ
C DO 10 K=1,NK
C WRITE(LUO,112) K
C DO 10 I=1,NZ
C I1=(K-1)*NZ+I
10 WRITE(LUO,114) (F(I1,J),J=1,NC)
C
C WRITE(LUO,116)
C WRITE(LUO,118) (UI(I),I=1,NZ)
C
C IF(ISWK.EQ.0) GO TO 16
C WRITE(LUO,120) (FNAMK(I),I=1,4)
C
C 16 WRITE(LUO,122)
C
C DO 22 K=1,NUSA
C T=(K-1)*TSAMP
C DO 20 I=1,NC1
C I1=(K-1)*NC1+I
20 RIO(I)=IO(I1)/25000.
22 WRITE(LUO,124) T,(RIO(I),I=1,NC1)
C
C 100 FORMAT(21H1DEAD-BEAT EXPERIMENT,10X,I2,1X,A5,
C *1X,2H19,I2,5X,1H(,2(I2,1H:),I2,1H)/21(1H*))
C 102 FORMAT(///18H SAMPLING PERIOD =,F7.2,4H SEC
C *///19H NUMBER OF STATES =,I2)
C 110 FORMAT(////16H FEEDBACK MATRIX,2X,A5)
C 112 FORMAT(/5X,1HL,I2)
C 114 FORMAT(1X,7F15.6/7F15.6/3F15.6)
C 116 FORMAT(29H1PROFILE OF DIFFUSION PROCESS/29(1H*))
C 118 FORMAT(////10X,15HINITIAL PROFILE,3X,3HU1=,1F7.3,
C *3X,3HU2=,1F7.3)
C 120 FORMAT(///10X,13HKALMAN FILTER,2X,4(A5,1X)///)
C 122 FORMAT(/4X,4HTIME,6X,2HU1,6X,2HU2,6X,2HY1,6X,2HY2,
C *6X,2HY3,6X,2HY4,6X,2HY5,6X,2HY6,6X,2HY7,5X,3HYE1,

```

C18.

\*5X,3HYE2/1X)

124 FORMAT(1X,16F8,3)

C

RETURN

C

END

APPENDIX D

TASK KEST  
 \*\*\*\*\*

INTERACTIVE PROGRAM FOR KALMAN ESTIMATION OF  
 THE PROFILE OF A DIFFUSION PROCESS

DESCRIPTION OF COMMAND STRUCTURE:

\*\*\*\*\*

INIT NUSA GAIN U1 U2

NUSA - NUMBER OF SAMPLING POINTS (MAX 512)  
 GAIN - GAIN OF INTERFACE (ANALOGUE COMPUTER)  
 U1,U2 - INITIAL BOUNDARY VALUES OF PROFILE (MEASURED)

KALF FNAME1 FNAME2 FNAME3 FNAME4

FNAME1 - NAME OF 'FI'-MATRIX  
 FNAME2 - NAME OF 'GAMMA'-MATRIX  
 FNAME3 - NAME OF 'THETA'-MATRIX  
 FNAME4 - NAME OF 'KALMAN GAIN'

AXIS IPD YMIN YMAX REZERO REGAIN

IPD - PLOT DENSITY (POINTS/SAMPLING INTERVAL)  
 (MAX 2 POINTS/SEC)  
 YMIN,YMAX - MIN AND MAX VALUE OF Y-AXIS  
 REZERO - ZERO POINT OF X-AXIS FOR RESIDUALS  
 REGAIN - GAIN FACTOR OF ERRORS/RESIDUALS WHEN PLOTTED

EXEC

PRINT

STOP

NB. IN EXEC MODE INPUTS, OUTPUTS AND ERRORS OF PROCESS ARE  
 PLOTTED, IF REGAIN IS POSITIVE. OTHERWISE ONLY INPUTS,  
 OUTPUTS CORRESPONDING TO MEASURED SIGNALS AND RESIDUALS  
 ARE PLOTTED.

SUBROUTINES REQUIRED

(FRMTOT)

```

C      INIKE
C      (RIFF)
C      KALF
C      RDMTX
C      (RIFF)
C      AXEKE
C      (AXIS)
C      (RSXPLT)
C      (SCALE)
C      EXEKE
C      EST
C      (ISENRX)
C      PLTKE
C      (LINE)
C      (SCALE)
C      (RIFF)
C      (RSXPLT)
C      PRTKE
C      (FRMTDT)
C      REST
C      (ACOF1)
C      (CSGFT1)
C      (FINO)
C
C      DIMENSION COM(6)
C
C      COMMON /CMD/ IP,IND,BUFF(16)
C      COMMON /DAT/ ITIME(6),AMONTH
C      COMMON /INDI/ IPLT,ISWP,KTH,KXE,KDTH(17),KDXE(17)
C      COMMON /KALM/ FNAME(4),FI(17,17),GA(17,2),
*TH(9,17),AK(17,9)
C      COMMON /PLT/ NUSA,IPD,YMIN,YMAX,REZERO,REGAIN,
*IX(1025)
C      COMMON /UYRE/ UI(2),UP(2),YP(9),RE(9),YRE(9)
C      COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),
*IFLOVA(48),USER(36)
C      COMMON /RTCOM/ IO(512)
C
C      DATA COM /4HINIT,4HKALF,4HAXIS,4HEXEC,5HPRINT,4HSTOP/
C
C      INITIATE
C
C      LUI=12
C      LUO=13
C      LUN=40
C      IPLT=0
C      NUSA=0
C
C      READ A COMMAND FROM TTO
C
C      2      IERR=0
C      CALL RLINE(LUO,LUI,BUFF,1)
C      IP=1
C
C      CALL RIFF(IP,RRES,IRES,IND)
C      IF(IND.EQ.1) GO TO 4
C      IF(IND-7) 8,2,8
C
C      4      DO 6 I=1,6
C      IF(RRES.EQ.COM(I)) GO TO(10,20,30,40,50,60),I
C      6      CONTINUE
C
C      8      IERR=1

```



```

                                D3.
C      GO TO 90
C      INIT
C
10     CALL INIKE(IERR)
       IF(IERR.EQ.0) GO TO 2
       GO TO 90
C
C      KALF
C
20     CALL KALF(IERR)
       IF(IERR.EQ.0) GO TO 2
       GO TO 90
C
C      AXIS
C
30     CALL AXEKE(IERR)
       IF(IERR.EQ.0) GO TO 2
       GO TO 90
C
C      EXEC
C
40     CALL RIFF(IP,RRES,IRES,IND)
       IF(IND.NE.7) GO TO 103
       CALL EXEKE(IERR)
       IF(IERR.EQ.0) GO TO 2
       GO TO 90
C
C      PRINT
C
50     CALL RIFF(IP,RRES,IRES,IND)
       IF(IND.NE.7) GO TO 103
       CALL PRTKE
       GO TO 2
C
C      STOP
C
60     IF(IPLT) 62,64,62
62     CALL PLOTS(LUN,2,IEV)
64     CALL EXIT
C
C      ERROR CHECK
C
90     NERR=14
       DO 92 I=1,NERR
       IF(I.EQ.IERR) GO TO(101,102,103,104,105,106,
*107,108,109,110,111,112,113,114),I
92     CONTINUE
       GO TO 2
C
C
101    WRITE(LU0,201)
       GO TO 2
102    WRITE(LU0,202)
       GO TO 2
103    WRITE(LU0,203)
       GO TO 2
104    WRITE(LU0,204)
       GO TO 2
105    WRITE(LU0,205)
       GO TO 2
106    WRITE(LU0,206)
       GO TO 2
107    WRITE(LU0,207)

```

D4.

```
GO TO 2
108 WRITE(LU0,208)
GO TO 2
109 WRITE(LU0,209)
GO TO 2
110 WRITE(LU0,210)
GO TO 2
111 WRITE(LU0,211)
GO TO 2
112 WRITE(LU0,212)
GO TO 2
113 WRITE(LU0,213)
GO TO 2
114 WRITE(LU0,214)
GO TO 2
```

C

C

```
201 FORMAT(' ILLEGAL COMMAND')
202 FORMAT(' TOO FEW ARGUMENTS')
203 FORMAT(' TOO MANY ARGUMENTS')
204 FORMAT(' BAD INTEGER')
205 FORMAT(' BAD NUMBER')
206 FORMAT(' BAD FILE NAME')
207 FORMAT(' BAD VALUE OF SAMPLING POINTS')
208 FORMAT(' BAD ESTIMATE OF PROFILE')
209 FORMAT(' FILE NOT FOUND')
210 FORMAT(' BAD PLOT DENSITY')
211 FORMAT(' BAD SCALE LIMIT(S)')
212 FORMAT(' NO SYSTEM MATRICES GIVEN')
213 FORMAT(' PLEASE DRAW AXIS')
214 FORMAT(' BAD VALUE OF GAIN')
```

C

C

END

SUBROUTINE INIKE(IERR)

INITIALIZES ESTIMATION OF PROFILE  
OF THE DIFFUSION PROCESS

IERR - ERROR SWITCH

AUTHOR: TORSTEN CROMNOW 1974-06-10

SUBROUTINE REQUIRED  
(RIFF)

COMMON /CMD/ IP,IND,BUFF(16)  
COMMON /PLT/ NUSA,IPD,YMIN,YMAX,REZERO,REGAIN,  
\*IX(1025)  
COMMON /UYRE/ UI(2),UP(2),YP(9),RE(9),YRE(9)  
COMMON /I0COM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),  
\*IFLOVA(48),USER(36)

EQUIVALENCE (NVAI,ISYSA(5)),(NVA0,ISYSA(6)),(GAIN,USER(1))

INITIATE

NVAI=9  
NVA0=2

READ ARGUMENTS

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND-3) 40,10,40  
IF(IRES.LE.0.OR.IRES.GT.512) GO TO 46  
NUSA=IRES

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND-4) 50,12,50  
IF(RRES.EQ.0.) GO TO 56  
GAIN=RRES

DO 16 I=1,2  
CALL RIFF(IP,RRES,IRES,IND)  
IF(IND-4) 48,14,48  
IF(ABS(RRES).GE.5.) GO TO 54  
UI(I)=RRES

CALL RIFF(IP,RRES,IRES,IND)  
IF(IND.NE.7) GO TO 52

RETURN

ERROR CHECK

40 IF(IND-7) 44,42,44  
42 IERR=2  
RETURN

44 IERR=4  
RETURN

D6.

46 IERR=7  
RETURN

C

48 IF(IND-7) 50,42,50  
50 IERR=5  
RETURN

C

52 IERR=3  
RETURN

C

54 IERR=8  
RETURN

C

56 IERR=14  
RETURN

C

END

D7.

```
      SUBROUTINE KALF(IERR)
C
C
C      READS SYSTEM MATRICES AND KALMANFILTER FROM DT1
C
C      IERR      - ERROR SWITCH
C
C
C      AUTHOR:      TORSTEN CROMNOW 1974-06-09
C
C      SUBROUTINES REQUIRED
C          RDMTX
C          (RIFF)
C
C      DIMENSION IT(10),HNM(4)
C
C      COMMON /CMD/ IP,IND,BUFF(16)
C      COMMON /INDI/ IPLT,ISWP,KTH,KXE,KDTH(17),KDXE(17)
C      COMMON /KALM/ FNAME(4),FI(17,17),GA(17,2),
C      *TH(9,17),AK(17,9)
C      COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),
C      *IFLOVA(48),USER(36)
C
C      EQUIVALENCE (NX,ISYSA(18)),(NU,ISYSA(19)),(NY,ISYSA(20)),
C      *(NE,ISYSA(21)),(ITICK,ISYSA(22))
C
C      READ ARGUMENTS
C
C      DO 2 I=1,4
C      CALL RIFF(IP,RRES,IRES,IND)
C      IF(IND-1) 30,2,30
C      FNAME(I)=RRES
C
C      CALL RIFF(IP,RRES,IRES,IND)
C      IF(IND.EQ.7) GO TO 10
C      IERR=3
C      RETURN
C
C      READ KALMANFILTER FROM DT1
C
C      10  LUN=33
C      CALL RDMTX(FNAME(1),LUN,IT,FI,IERR)
C      IF(IERR.NE.0) GO TO 40
C      NX=IT(1)
C      ITICK=IT(4)
C      NE=IT(5)
C      KXE=IT(6)
C
C      CALL RDMTX(FNAME(2),LUN,IT,GA,IERR)
C      IF(IERR.NE.0) GO TO 40
C      NU=IT(2)
C
C      CALL RDMTX(FNAME(3),LUN,IT,TH,IERR)
C      IF(IERR.NE.0) GO TO 40
C      NY=IT(1)
C      KTH=IT(5)
C
C      CALL RDMTX(FNAME(4),LUN,IT,AK,IERR)
C      IF(IERR.NE.0) GO TO 40
C
C      DO 20 I=1,NX
```

D8.

```
IC1=1
IC2=1
IF((KTH/2)*2, EQ, KTH) IC1=0
IF((KXE/2)*2, EQ, KXE) IC2=0
I1=NX+1-1
KDTH(I1)=IC1
KDXE(I1)=IC2
KTH=KTH/2
KXE=KXE/2
```

20

C

```
RETURN
```

C

C

```
ERROR CHECK
```

C

```
30 IF(IND-7) 34,32,34
```

32

```
IERR=2
```

```
RETURN
```

C

34

```
IERR=6
```

```
RETURN
```

C

40

```
IERR=9
```

```
RETURN
```

C

```
END
```

```
SUBROUTINE RDMTX(FNAME,LUN,KODE,A,IERR)
```

```
READS A SYNPAK MATRIX FROM LOGICAL UNIT LUN
```

```
FNAME - FILENAME  
LUN    - LOGICAL UNIT  
KODE   - FILE HEAD OF MATRIX A  
A      - MATRIX OF ORDER NR*NC  
IERR   - ERROR SWITCH
```

```
AUTHOR: TORSTEN CROMNOW 1974-04-24
```

```
SUBROUTINE REQUIRED  
NONE
```

```
DIMENSION KODE(1),A(1,1)
```

```
IERR=0
```

```
CALL SEEK(LUN,FNAME,3HBIN,IEV)  
CALL WAITFR(IEV)  
IF(IEV.LT.0) GO TO 20
```

```
READ(LUN)(KODE(I),I=1,10)  
NR=KODE(1)  
NC=KODE(2)  
DO 10 I=1,NR  
10 READ(LUN)(A(I,J),J=1,NC)
```

```
CALL CLOSE(LUN,FNAME,3HBIN,IEV)  
CALL WAITFR(IEV)  
IF(IEV.LT.0) GO TO 20  
RETURN
```

```
20 IERR=1  
RETURN
```

```
END
```

D10.

SUBROUTINE AXEKE(IERR)

READS PLOTTING ARGUMENTS AND DRAWS AXIS ON VP

IERR - ERROR SWITCH

AUTHOR: TORSTEN CROMNOW 1974-06-04

SUBROUTINES REQUIRED

RSXPLT

(AXIS)

(SCALE)

DIMENSION AMON(12),X(921)

COMMON /CMD/ IP,IND,BUFF(16)

COMMON /DAT/ ITIME(6),AMONTH

COMMON /INDI/ IPLT,ISWP,KTH,KXE,KDTH(17),KDXE(17)

COMMON /KALM/ FNAME(4),FI(17,17),GA(17,2),

\*TH(9,17),AK(17,9)

COMMON /PLT/ NUSA,IPD,YMIN,YMAX,REZERO,REGAIN,

\*IX(1025)

COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),

\*IFLOVA(48),USER(36)

EQUIVALENCE (NY,ISYSA(20)),(ITICK,ISYSA(22))

DATA AMON /3HJAN,3HFEB,4HMARS,5HAPRIL,3HMAJ,  
\*4HJUNI,4HJULY,3HAUG,4HSEPT,3HOCT,3HNOV,3HDEC/

ISWP=0

IF(NUSA.GT.1.AND,NY\*NUSA.LT.512) GO TO 2

IERR=7

RETURN

2 IF(ITICK) 8,6,8

6 IERR=12

RETURN

READ ARGUMENTS

8 CALL RIFF(IP,RRES,IRES,IND)

IF(IND-3) 40,10,40

10 IPD=IRES

IF(IPD.LT.1.OR,IPD\*NUSA.GT.920) GO TO 11

IF(ITICK/IPD.GE.25.AND,ITICK.EQ,IPD\*(ITICK/IPD)) GO TO 12

11 IERR=10

RETURN

12 CALL RIFF(IP,RRES,IRES,IND)

IF(IND-4) 50,14,50

14 YMIN=RRES

IF(ABS(YMIN).LE.10.) GO TO 16

IERR=11

RETURN

16 CALL RIFF(IP,RRES,IRES,IND)

IF(IND-4) 50,18,50

18 YMAX=RRES

IF(ABS(YMAX).LE.10..AND,(YMAX-YMIN).GT.0.) GO TO 20

IERR=11

RETURN

20 CALL RIFF(IP,RRES,IRES,IND)



D11.

```
IF(IND-4) 50,22,50
22 REZERO=RRES
   IF(ABS(REZERO),LE,10.) GO TO 24
   IERR=11
   RETURN
C
24 CALL RIFF(IP,RRES,IRES,IND)
   IF(IND-4) 50,26,50
26 IF(RRES,LT,0.) ISWP=1
   REGAIN=ABS(RRES)
C
   CALL RIFF(IP,RRES,IRES,IND)
   IF(IND.EQ,7) GO TO 30
   IERR=3
   RETURN
C
C DRAW AXIS ON VP
C
30 LUN=40
   TSAMP=ITICK/50./IPD
   XMAX=(NUSA-1)*ITICK/50.
   CALL PLOTS(LUN,0,IEV)
   CALL DATE(ETIME)
   ISL=ETIME(1)
   ITIME(1)=ETIME(2)
   AMONTH=AMON(ISL)
   WRITE(LUN,300) ITIME(1),AMONTH,ITIME(3),(FNAME(I),I=1,4)
   NI=(NUSA-1)*IPD+1
   DO 32 I=1,NI
32 X(I)=(I-1)*TSAMP
C
   CALL SCALE(X(1),IX(1),NI,22.,0.,XMAX,SMINX,DSX,1)
   ASSIGN 100 TO IFMT
   CALL AXIS(0,0,22.,IFMT,8,0.,SMINX,DSX)
C
   CALL SCALE(REZERO,IZERO,1,17.,YMIN,YMAX,SMINY,DSY,1)
   CALL AXIS(0,IZERO,22.,IFMT,8,0.,SMINX,DSX)
C
   ASSIGN 200 TO IFMT
   CALL AXIS(0,0,-17.,IFMT,7,90.,SMINY,DSY)
C
   IPLT=1
C
   RETURN
C
40 IF(IND-7) 44,42,44
42 IERR=2
   RETURN
C
44 IERR=4
   RETURN
C
50 IF(IND-7) 52,42,52
52 IERR=5
   RETURN
C
100 FORMAT(8HTIME SEC)
200 FORMAT(7HVOLT(S))
300 FORMAT(9X,'KALMAN ESTIMATION',I8,1X,A5,I8,
*//9X,'KALMAN FILTER',1X,A5,1X,A5,1X,A5,1X,A5)
C
   END
```

SUBROUTINE EXEKE(IERR)

```

C
C
C PERFORMS KALMAN ESTIMATION AND PLOTS INPUTS,OUTPUTS
C AND ESTIMATION ERRORS ON VP
C
C IERR      - ERROR SWITCH
C
C
C AUTHOR:    TORSTEN CROMNOW  1974-06-10
C
C SUBROUTINES REQUIRED
C   PLTKE
C   (LINE)
C   (SCALE)
C   EST
C   RSXPLT
C   ISENRX
C
C DIMENSION IT(4),Y(9),U(8),XE(17)
C
C COMMON /INDI/ IPLT,ISWP,KTH,KXE,KDTH(17),KDXE(17)
C COMMON /PLT/  NUSA,IPD,YMIN,YMAX,REZERO,REGAIN,
C *IX(1025)
C COMMON /UYRE/ UI(2),UP(2),YP(9),RE(9),YRE(9)
C COMMON /IDCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),
C *IFLOVA(48),USER(36)
C
C EQUIVALENC (NX,ISYSA(18)),(NE,ISYSA(21)),(ITICK,ISYSA(22)),
C *(ISWN,ISYSA(23)),(IEV1,ICOMVA(41)),(IEV5,ICOMVA(45)),
C *(Y(1),IFLOVA(1)),(U(1),IFLOVA(33)),(GAIN,USER(1)),
C *(XE(1),USER(7))
C
C DATA IT/10,1,0,1/
C
C TEST IF AXIS IS DRAWN
C
C IF(IPLT) 4,2,4
C 2 IERR=13
C RETURN
C
C INITIATE
C
C 4 LUN=40
C   IPER=IPD
C   NS=0
C   IC1=IPER
C   IC2=0
C   ISW1=0
C   IEV1=0
C   ISWN=1
C   IT(3)=ITICK/IPD
C
C UP(1)=U(1)/GAIN
C UP(2)=U(2)/GAIN
C
C CALL FIX(4HALIO)
C CALL RUN(4HALIO,IT,150)
C
C COMPUTE INITIAL VALUE OF ESTIMATE
C
C IF(NE.EQ.7) GO TO 8
C DO 6 I=1,NX

```

```

6   XE(1)=UI(1)+(1-1)*(UI(2)-UI(1))/(NX-1)
   GO TO 10
   DO 8 I=1,NX
8   XE(1)=UI(1)+I*(UI(2)-UI(1))/(NX+1)
C
C   MAIN LOOP
C
10  IF(NS-NUSA) 12,70,70
12  CALL WAITFR(IEV1)
   IEV1=0
C
C   SCALE PROCESS OUTPUTS AND CHANGE ORDER
C   OF PROCESS OUTPUTS IF NE.EQ.9
C
   IF(NE-7) 20,24,20
20  DO 22 I=1,7
22  YP(I+1)=Y(I)/GAIN
   YP(1)=Y(8)/GAIN
   YP(9)=Y(9)/GAIN
   GO TO 30
24  DO 26 I=1,7
26  YP(I)=Y(I)/GAIN
C
C   MASK OUT MEASURABLE OUTPUTS
C
30  I1=0
   I2=0
   DO 32 I=1,NX
   IF(KDXE(I).EQ.1) I1=I1+1
   IF(KDTH(I).EQ.0) GO TO 32
   I2=I2+1
   YRE(I2)=YP(I1)
32  CONTINUE
C
   IF(IC1-IPER) 60,40,60
C
C   SET OUT NEW PROCESS INPUTS STORED IN FLA0 7,8 IF IEV5.EQ.1
C
40  IF(IEV5-1) 50,42,50
42  IEV5=0
   UP(1)=U(7)
   U(1)=GAIN*U(7)
   UP(2)=U(8)
   U(2)=GAIN*U(8)
C
C   UPDATE ESTIMATE OF STATE VARIABLES
C
50  CALL EST(IC2)
   IC1=0
   NS=NS+1
C
C   PLOT INPUTS,OUTPUTS OF PROCESS AND ESTIMATION ERRORS
C
60  CALL PLTKE(IC1,ISW1)
   IC1=IC1+1
C
C   IF DATASWITCH 1 IS SET THE EXPERIMENT IS STOPPED
C
   IF(ISENSW(1)) GO TO 70
C
   GO TO 10
C
C   READY
C

```

```
70 CALL UNFIX(4HAL10)          D14.  
   CALL CANCEL(4HAL10)  
   CALL PLOTS(LUN,2,IEV)  
   IPLT=0  
C  
   RETURN  
C  
   END
```

```

SUBROUTINE EST(IC)
C
C PERFORMS KALMAN ESTIMATION
C
C IC - COUNTER
C
C AUTHOR: TORSTEN CROMNOW 1974-06-10
C
C SUBROUTINE REQUIRED
C NONE
C
C DIMENSION E(9),XE0(17),XE(17)
C
C COMMON /INDI/ IPLT,ISWP,KTH,KXE,KDTH(17),KDXE(17)
C COMMON /KALM/ FNAME(4),FI(17,17),GA(17,2),
*TH(9,17),AK(17,9)
C COMMON /UYRE/ UI(2),UP(2),YP(9),RE(9),YRE(9)
C COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),
*IFLOVA(48),USER(36)
C COMMON /RTCOM/ IO(512)
C
C EQUIVALENCE (NX,ISYSA(18)),(NU,ISYSA(19)),(NY,ISYSA(20)),
*(XE(1),USER(7)),(E(1),USER(25))
C
C COMPUTE ESTIMATION ERRORS
C
C I1=0
C DO 10 I=1,NX
C IF(KDXE(I).EQ.0) GO TO 10
C I1=I1+1
C E(I1)=YP(I1)-XE(I)
10 CONTINUE
C
C STORE RESIDUALS AND COMPUTE NEW ESTIMATION
C
C CALL MOVE(XE(1),XE0(1),NX+NX)
C DO 20 I=1,NY
C I1=IC+I
C RE(I)=YRE(I)-SCAPRO(TH(I,1),9,XE(1),1,NX)
20 IO(I1)=25000.*RE(I)
C DO 22 I=1,NX
22 XE(I)=SCAPRO(FI(I,1),17,XE0(1),1,NX)+
*SCAPRO(GA(I,1),17,UP(1),1,NU)+
*SCAPRO(AK(I,1),17,RE(1),1,NY)
C
C INCREMENT COUNTER
C
C IC=IC+NY
C
C RETURN

```

```

SUBROUTINE PLTKE(ICNT,ISW)
C
C PLOTS INPUTS, OUTPUTS AND ESTIMATION ERRORS OF A PROCESS ON VP
C
C ISW      - PUT ISW=0 TO INITIALIZE THE PLOTTING
C
C AUTHOR:   TORSTEN CROMNOW  1974-06-24
C
C SUBROUTINES REQUIRED
C          RSXPLT
C          (SCALE)
C          (LINE)
C
C          DIMENSION E(9),ES(9),IXE(2),UYE(20),IUYE(20),IUYEP(2,20)
C
C          COMMON /INDI/ IPLT,ISWP,KTH,KXE,KDTH(17),KDXE(17)
C          COMMON /PLT/ NUSA,IPD,YMIN,YMAX,REZERO,REGAIN,
C          *IX(1025)
C          COMMON /UYRE/ UI(2),UP(2),YP(9),RE(9),YRE(9)
C          COMMON /ILOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),
C          *IFLOVA(48),USER(36)
C
C          EQUIVALENCE (NX,ISYSA(18)),(NU,ISYSA(19)),(NY,ISYSA(20)),
C          *(NE,ISYSA(21)),(E(1),USER(25))
C
C          INITIATE
C
C          NEP=NE
C          IF(ISWP.EQ.1) NEP=NY
C          NYP=9
C          IF(ISWP.EQ.1) NYP=NY
C          NUY=NU+NYP
C          NUYE=NU+NYP+NEP
C          ICNTRL=0
C          IF(IPD.EQ.1) ICNTRL=1
C          IF(ICNT.EQ.1) ICNTRL=2
C
C          SCALE INPUTS,OUTPUTS AND ERRORS
C
C          IF(ISWP-1) 2,6,2
C          DO 4 I=1,NE
C          ES(I)=REZERO+REGAIN*E(I)
C          GO TO 10
C          DO 8 I=1,NY
C          ES(I)=REZERO+REGAIN*RE(I)
C
C          CALL MOVE(UP(1),UYE(1),NU+NU)
C          IF(ISWP-1) 12,14,12
C          CALL MOVE(YP(1),UYE(NU+1),18)
C          GO TO 16
C          CALL MOVE(YRE(1),UYE(NU+1),NY+NY)
C          CALL MOVE(ES(1),UYE(NUY+1),NEP+NEP)
C          CALL SCALE(UYE(1),IUYE(1),NUYE,17.,YMIN,YMAX,SMIN,DS,1)
C
C          IF(ISW) 30,20,30
C
C          INITIATE PLOTTING PROCEDURE
C
C          ISW=1
C          IC1=0
C          IXE(2)=IX(1)
C          DO 22 I=1,NUYE

```

```
22  IUYEP(2,1)=IUYE(1)
    GO TO 40
C
C  PLOT INPUTS AND OUTPUTS
C
30  DO 32 I=1,NUY
    I1=I-1
    IF(I1.GT.7) I1=I1-8
    IUYEP(1,I)=IUYEP(2,I)
    IUYEP(2,I)=IUYE(I)
32  CALL LINE(IX(IC1),IUYEP(1,I),2,1,ICNTRL,I1)
C
    IF(ICNT) 40,34,40
C
C  PLOT RESIDUALS RE
C
34  ICNTRL=1
    IXE(1)=IXE(2)
    IXE(2)=IX(IC1+1)
    DO 36 I=1,NEP
    I1=I-1
    IF(I1.GT.7) I1=I1-8
    I2=I+NUY
    IUYEP(1,I2)=IUYEP(2,I2)
    IUYEP(2,I2)=IUYE(I2)
36  CALL LINE(IXE(1),IUYEP(1,I2),2,1,ICNTRL,I1)
C
40  IC1=IC1+1
C
    RETURN
C
    END
```

SUBROUTINE PRTKE

PRINTS DESCRIPTION OF A KALMANFILTER

AUTHOR: TORSTEN CROMNOW 1974-06-26

SUBROUTINE REQUIRED  
REST

COMMON /DAT/ ITIME(6),AMONTH  
COMMON /INDI/ IPLT,ISWP,KTH,KXE,KDTH(17),KDXE(17)  
COMMON /KALM/ FNAME(4),FI(17,17),GA(17,2),  
\*TH(9,17),AK(17,9)  
COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),  
\*IFLOVA(48),USER(36)

EQUIVALENCE (NX,ISYSA(18)),(NU,ISYSA(19)),(NY,ISYSA(20)),  
\*(NE,ISYSA(21)),(ITICK,ISYSA(22))

INITIATE

LUO=36

PRINT KALMAN FILTER

WRITE(LUO,100) ITIME(1),AMONTH,ITIME(3)  
WRITE(LUO,102) NX,NU,NY  
WRITE(LUO,104) NE,ITICK  
WRITE(LUO,106) (KDTH(I),I=1,NX)  
WRITE(LUO,112) (KDXE(I),I=1,NX)

WRITE(LUO,108) FNAME(1)  
DO 10 I=1,NX  
WRITE(LUO,110) (FI(I,J),J=1,NX)

WRITE(LUO,108) FNAME(2)  
DO 20 I=1,NX  
WRITE(LUO,110) (GA(I,J),J=1,NU)

WRITE(LUO,108) FNAME(3)  
DO 30 I=1,NY  
WRITE(LUO,110) (TH(I,J),J=1,NX)

WRITE(LUO,108) FNAME(4)  
DO 40 I=1,NX  
WRITE(LUO,110) (AK(I,J),J=1,NY)

CHECK STATISTICAL PROPERTIES OF RESIDUALS

CALL REST

100 FORMAT('1DESCRIPTION OF KALMAN FILTER',9X,  
\*18,1X,A5,18/1X,9('\*'),9('\*'),9('\*'),'\*')  
102 FORMAT(///// 'NX=',18,5X,'NU=',18,5X,'NY=',18//)  
104 FORMAT(// 'NE=',18,4X,'ITICK=',18)  
106 FORMAT(//9X,9X,'VECTOR KDTH'//918/818)  
108 FORMAT(///// 'MATRIX '\*,A5//)  
110 FORMAT(1X,7E15.6/1X,7E15.6/1X,3E15.6)



D19.

```
112  FORMAT(//9X,9X,'VECTOR KDXE'//9I8/8I8)  
C  
      RETURN  
C  
      END
```

## SUBROUTINE REST

```

C
C COMPUTES AUTOCOVARIANCE FUNCTION AND COVARIANCE
C MATRIX OF RESIDUALS.
C CORRELATION AND NORMALITY OF THE RESIDUALS IS
C ALSO TESTED.
C
C AUTHOR:      TORSTEN CROMNOW 1974-08-16
C
C SUBROUTINES REQUIRED
C   ACOF
C   CSGFT
C   FINO
C
C DIMENSION ACF(20),RES(512),R(9,9)
C
C COMMON /PLT/ NUSA,IPD,YMIN,YMAX,REZERO,REGAIN,
*IX(1025)
C COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),
*ICOMVA(45),IFLOVA(48),USER(36)
C COMMON /RTCOM/ IO(512)
C
C EQUIVALENCE (NY,ISYSA(20)),(RES(1),IX(1))
C
C INITIATE
C
C LU0=36
C NOL=21
C NLAG=5
C
C PERFORM STATISTICAL TESTS
C
C RNUSA=NUSA
C WRITE(LU0,100)
C
C DO 18 K=1,NY
C   IC=K
C   DO 10 I=1,NUSA
C     RES(I)=IO(IC)/25000.
10   IC=IC+NY
C
C WRITE(LU0,102) K,(RES(I),I=1,NUSA)
C
C TEST RESIDUALS FOR NORMALITY
C
C CALL CSGFT1(RES,NUSA,2,LU0,CHI2,CORR)
C
C COMPUTE AUTOCOVARIANCE FUNCTION AND TEST
C QUANTITIES FOR NO CORRELATION OF RESIDUALS
C
C CALL ACOF1(RES,RES,ACF,NUSA,NOL,0,LU0)
C SL=ACF(1)
C DO 12 I=1,NOL
12   ACF(I)=ACF(I)/SL
C   CI=1.96/SQRT(RNUSA)
C
C WRITE(LU0,106) (ACF(I),I=1,NOL)
C WRITE(LU0,108) CI
C
C NI=NLAG+1
C TQ=0.
C DO 14 I=2,NI
14   TQ=TQ+RNUSA*ACF(I)*ACF(I)

```

```

WRITE(LUO,110) TQ,NLAG
C
C   COMPUTE NUMBER OF CHANGES OF SIGN OF RESIDUALS
C
   ICS=0
   DO 16 I=2,NUSA
   SL=RES(I)
16  IF(SL*RES(I-1).LT.0.) ICS=ICS+1
   ISL=IFIX(1.96*SQRT(RNUSA-1.)/2.+0.5)
   I1=(NUSA-1)/2-ISL
   I2=(NUSA-1)/2+ISL
C
18  WRITE(LUO,104) ICS,I1,I2
C
C   COMPUTE COVARIANCE MATRIX OF RESIDUALS
C
   DO 22 K=1,NY
   IC=K
   S=0.
   DO 20 I=1,NUSA
   S=I0(IC)/25000.+S
20  IC=IC+NY
   S=S/RNUSA
   IC=K
   DO 22 I=1,NUSA
   RES(IC)=I0(IC)/25000.-S
22  IC=IC+NY
C
   DO 24 I=1,NY
   DO 24 J=1,NY
   R(I,J)=SCAPRO(RES(I),NY,RES(J),NY,NUSA)/(RNUSA-1.)
24  R(J,I)=R(I,J)
C
   WRITE(LUO,120)
   DO 26 I=1,NY
26  WRITE(LUO,122) (R(I,J),J=1,NY)
C
100  FORMAT('1STATISTICAL TEST OF RESIDUALS'/1X,
*9('*'),9('*'),9('*'),2('*'))
102  FORMAT(/////////' RESIDUAL NO=',I8/(1X,8E15.6))
104  FORMAT(//' NUMBER OF CHANGES OF SIGN OF RESIDUALS',
*18//' CONFIDENCE INTERVAL 95 PERCENT',2I8)
106  FORMAT(//' AUTO-COVARIANCE FUNCTION'/(1X,8E15.6))
108  FORMAT(//' CONFIDENCE INTERVAL 95 PERCENT',E15.6)
110  FORMAT(//' TEST QUANTITY FOR NO CORRELATION OF RESIDUALS',
*E15.6//' DEGREE OF FREEDOM',I8)
120  FORMAT(/////////' COVARIANCE MATRIX OF RESIDUALS')
122  FORMAT(1X,8E15.6)
C
   RETURN
C
   END

```

```
C      TASK NOISE
C
C      GENERATES TWO SEQUENCIES OF INDEPENDENT PSEUDO-
C      RANDOM NUMBERS, EACH DISTRIBUTED ACCORDING TO A
C      NORMAL (0,SIGMA) DISTRIBUTION. (SEQUENCIES
C      ALSO INDEPENDENT)
C
C      AUTHOR:      TORSTEN CROMNOW 1974-08-20
C
C      SUBROUTINE REQUIRED
C      MCNODI
C
C      NB.THE NOISE GENERATOR IS INITIATED BY PUTTING
C      THE CELL IEXT 6 IN COMMON BLOCK IOCOM EQUAL
C      TO 1. THE STANDARD DEVIATIONS SIGMA1 AND SIGMA2
C      OF THE SEQUENCIES SHOULD BE PLACED IN THE COMMON
C      BLOCK IOCOM,CELL USER 5-6.
C
C      DIMENSION U(8)
C
C      COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),
C      *IFLOVA(48),USER(36)
C
C      EQUIVALENCE (ISW,ISYSA(23)),(NU1,ISYSA(24)),
C      *(NU2,ISYSA(25)),(IEV5,ICOMVA(45)),(U(1),IFLOVA(33)),
C      *(SIGMA1,USER(5)),(SIGMA2,USER(6))
C
C      INITIATE
C
C      IF (ISW) 20,20,10
10      NU1=19
C      NU2=21
C      ISW=0
C
C      GENERATE RANDOM SEQUENCIES
C
C      20      CALL MCNODI(NU1,GS1)
C      CALL MCNODI(NU2,GS2)
C      IEV5=1
C      U(7)=SIGMA1*GS1
C      U(8)=SIGMA2*GS2
C
C      CALL EXIT
C
C      END
```

APPENDIX E



```

C
  DIMENSION COM(5)
C
  COMMON /CMD/ IP,IND,BUFF(16)
  COMMON /IND1/ IDUM1(7)
  COMMON /KALM/ IDUM2(1261)
  COMMON /UY/ IDUM3(22)
  COMMON /IOCOM/ IDUM4(256)
C
  DATA COM /4HINIT,4HKALF,4HEXEC,5HPRINT,4HSTOP/
C
  C
  C   INITIATE
  C
  C   LUI=38
  C   LUO=39
C
  C   READ A COMMAND FROM TT1
  C
  C
  2   IERR=0
  CALL RLINE(LUO,LUI,BUFF,1)
  IP=1
C
  CALL RIFF(IP,RRES,IRES,IND)
  IF(IND-1) 3,4,3
  3   IF(IND-7) 8,2,8
C
  4   DO 6 I=1,5
  IF(RRES.EQ.COM(I)) GO TO(10,20,30,40,50),I
  6   CONTINUE
C
  8   IERR=1
  GO TO 70
C
  C   INIT
  C
  C
  10  CALL INISKE(IERR)
  GO TO 60
C
  C   KALF
  C
  C
  20  CALL KALFS(IERR)
  GO TO 60
C
  C   EXEC
  C
  C
  30  CALL RIFF(IP,RRES,IRES,IND)
  IF(IND-7) 32,34,32
  32  IERR=3
  GO TO 70
  34  CALL EXESKE(IERR)
  GO TO 60
C
  C   PRINT
  C
  C
  40  CALL PRTSKE
  GO TO 2
C
  C   STOP
  C
  C
  50  CALL EXIT
C
  C   ERROR CHECK
  C
  60  IF(IERR) 70,2,70

```

E3.

70 CALL ERRSKE(IERR)  
GO TO 2

C

END



```
      SUBROUTINE ERRSKE(IERR)
C
C   PRINTS OUT ERROR INFORMATION
C
C   IERR - ERROR SWITCH
C
C   AUTHOR:      TORSTEN CROMNOW 1974-07-05
C
C   SUBROUTINE REQUIRED
C       NONE
C
C       LU0=39
C       NERR=11
C       DO 10 I=1,NERR
C         IF(I.EQ.IERR) GO TO(101,102,103,104,105,106,
10          *107,108,109,110,111),I
C       CONTINUE
C
C       RETURN
C
101  WRITE(LU0,201)
      RETURN
102  WRITE(LU0,202)
      RETURN
103  WRITE(LU0,203)
      RETURN
104  WRITE(LU0,204)
      RETURN
105  WRITE(LU0,205)
      RETURN
106  WRITE(LU0,206)
      RETURN
107  WRITE(LU0,207)
      RETURN
108  WRITE(LU0,208)
      RETURN
109  WRITE(LU0,209)
      RETURN
110  WRITE(LU0,210)
      RETURN
111  WRITE(LU0,211)
      RETURN
C
201  FORMAT(' ILLEGAL COMMAND')
202  FORMAT(' TOO FEW ARGUMENTS')
203  FORMAT(' TOO MANY ARGUMENTS')
204  FORMAT(' BAD INTEGER')
205  FORMAT(' BAD NUMBER')
206  FORMAT(' BAD FILE NAME')
207  FORMAT(' BAD VALUE OF SAMPLING POINTS')
208  FORMAT(' BAD ESTIMATE OF PROFILE')
209  FORMAT(' FILE NOT FOUND')
210  FORMAT(' NO SYSTEM MATRICES GIVEN')
211  FORMAT(' BAD VALUE OF GAIN')
C
      END
```

E5.

```

SUBROUTINE INISKE(IERR)
C
C   INITIALIZES ESTIMATION OF PROFILE OF
C   THE DIFFUSION PROCESS
C
C   IERR      - ERROR SWITCH
C
C   AUTHOR:   TORSTEN CROMNOW  1974-06-10
C
C   SUBROUTINE REQUIRED
C           (RIFF)
C
C
C   COMMON /CMD/  IP,IND,BUFF(16)
C   COMMON /IND1/ NUSA,KXE,KTH,UI(2)
C   COMMON /IOCOM/ ISYSA(25),IOA(26),MPXA(40),ICOMVA(45),
C   *IFLOVA(48),USER(36)
C
C   EQUIVALENCE (NVA1,ISYSA(5)),(NVA0,ISYSA(6)),(GAIN,USER(1))
C
C   INITIATE
C
C   2   NVA1=9
C       NVA0=2
C
C   READ ARGUMENTS
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-3) 40,10,40
C   10  IF(IRES.LE.0.OR.IRES.GT.10000) GO TO 46
C       NUSA=IRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-4) 50,12,50
C   12  IF(RRES.EQ.0.) GO TO 56
C       GAIN=RRES
C
C   DO 16 I=1,2
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-4) 48,14,48
C   14  IF(ABS(RRES).GE.5.) GO TO 54
C   16  UI(I)=RRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND.NE.7) GO TO 52
C
C   RETURN
C
C   ERROR CHECK
C
C   40  IF(IND-7) 44,42,44
C   42  IERR=2
C       RETURN
C
C   44  IERR=4
C       RETURN
C
C   46  IERR=7
C       RETURN

```

E6.

```
C
 48  IF(IND-7) 50,42,50
 50  IERR=5
      RETURN
C
 52  IERR=3
      RETURN
C
 54  IERR=8
      RETURN
C
 56  IERR=11
      RETURN
C
      END
```

## E7.

SUBROUTINE KALFS(IERR)

```

C
C
C   READS SYSTEM MATRICES AND KALMANFILTER FROM DT1
C
C   IERR      - ERROR SWITCH
C
C
C   AUTHOR:    TORSTEN CROMNOW 1974-06-09
C
C   SUBROUTINES REQUIRED
C       RDMTX
C       (RIFF)
C
C   DIMENSION IT(10),FNAME(4)
C
C   COMMON /CMD/  IP,IND,BUFF(16)
C   COMMON /INDI/ NUSA,KXE,KTH,UI(2)
C   COMMON /KALM/ NX,NU,NY,FI(17,17),GA(17,2),
*TH(9,17),AK(17,9)
C   COMMON /IOCOM/ IDUM(184),USER(36)
C
C   EQUIVALENCE (NE,IDUM(18)),(ITICK,IDUM(19)),
* (FNAME(1),USER(3))
C
C   READ ARGUMENTS
C
C   DO 2 I=1,4
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-1) 30,2,30
C   FNAME(I)=RRES
C
C   CALL RIFF(IP,RRES,IRES,IND)
C   IF(IND-7) 4,10,4
C   IERR=3
C   RETURN
C
C   READ KALMANFILTER FROM DT1
C
C   10  CALL RDMTX(FNAME(1),33,IT,FI,IERR)
C   IF(IERR) 40,12,40
C   12  NX=IT(1)
C   ITICK=IT(4)
C   NE=IT(5)
C   KXE=IT(6)
C
C   CALL RDMTX(FNAME(2),33,IT,GA,IERR)
C   IF(IERR) 40,14,40
C   14  NU=IT(2)
C
C   CALL RDMTX(FNAME(3),33,IT,TH,IERR)
C   IF(IERR) 40,16,40
C   16  NY=IT(1)
C   KTH=IT(5)
C
C   CALL RDMTX(FNAME(4),33,IT,AK,IERR)
C   IF(IERR) 40,18,40
C
C   18  RETURN
C
C   ERROR CHECK

```

E8.

```
C
30  IF (IND-7) 34,32,34
32  IERR=2
    RETURN
```

```
C
34  IERR=6
    RETURN
```

```
C
40  IERR=9
    RETURN
```

```
C
    END
```

```
      SUBROUTINE RDMTX(FNAME,LUN,KODE,A,IERR)
C
C
C      READS A SYNPAK MATRIX FROM LOGICAL UNIT LUN
C
C      FNAME  - FILENAME
C      LUN    - LOGICAL UNIT
C      KODE   - FILE HEAD OF MATRIX A
C      A      - MATRIX OF ORDER NR*NC
C      IERR   - ERROR SWITCH
C
C
C      AUTHOR:      TORSTEN CROMNOW 1974-04-24
C
C      SUBROUTINE REQUIRED
C      NONE
C
C      DIMENSION KODE(1),A(1,1)
C
C      IERR=0
C
C      CALL SEEK(LUN,FNAME,3HBIN,IEV)
C      CALL WAITFR(IEV)
C      IF(IEV.LT.0) GO TO 20
C
C      READ(LUN)(KODE(I),I=1,10)
C      NR=KODE(1)
C      NC=KODE(2)
C      DO 10 I=1,NR
10    READ(LUN)(A(I,J),J=1,NC)
C
C      CALL CLOSE(LUN,FNAME,3HBIN,IEV)
C      CALL WAITFR(IEV)
C      IF(IEV.LT.0) GO TO 20
C      RETURN
C
C      20  IERR=1
C          RETURN
C
C      END
```

```

SUBROUTINE EXESKE(IERR)
C
C
C PERFORMS KALMAN ESTIMATION
C
C IERR      - ERROR SWITCH
C
C
C AUTHOR:    TORSTEN CROMNOW  1974-05-10
C
C SUBROUTINES REQUIRED
C   ISENRX
C
C DIMENSION IT(4),KDXE(17),KDBTH(17),U(2),Y(9),YRE(9),
*E(9),RE(9),XE(17),XE0(17)
C
C COMMON /INDI/ NUSA,KXE,KTH,UI(2)
COMMON /KALM/ NX,NU,NY,FI(17,17),GA(17,2),
*TH(9,17),AK(17,9)
COMMON /UY/ UP(2),YP(9)
COMMON /IOCOM/ ISYSA(25),IDUM(66),ICOMVA(45),
*IFLOVA(48),USER(36)
C
C EQUIVALENCE (NE,ISYSA(18)),(ITICK,ISYSA(19)),
*(IPER,ISYSA(20)),(ISWD,ISYSA(22)),(ISWK,ISYSA(23)),
*(IPLI,ISYSA(24)),(IEV1,ICOMVA(41)),(IEV3,ICOMVA(43)),
*(IEV4,ICOMVA(44)),(Y(1),IFLOVA(1)),(U(1),IFLOVA(33)),
*(GAIN,USER(1)),(XE(1),USER(7)),(E(1),USER(25))
C
C DATA IT/10,1,0,1/
C
C TEST IF READY TO GO
C
C IF(NX.GE.1.AND.NX.LE.17) GO TO 2
IERR=10
RETURN
2  IF(GAIN) 6,4,6
4  IERR=8
RETURN
C
C INITIATE
C
6  ISWK=1
IPER=1
ICNT=IPER
NS=0
IEV1=0
IT(3)=ITICK/IPER
C
C CALL FIX(4HALIO)
CALL RUN(4HALIO,IT,150)
C
C COMPUTE INITIAL VALUE OF ESTIMATE
C
C IF(NE=7) 10,14,10
10 DO 12 I=1,NX
12 XE(I)=UI(1)+(I-1)*(UI(2)-UI(1))/(NX-1)
GO TO 18
14 DO 16 I=1,NX
16 XE(I)=UI(1)+I*(UI(2)-UI(1))/(NX+1)
C
C FIND CODE FOR MEASURABLE AND MEASURED OUTPUTS
C

```

```

18  KKXE=KXE
    KKTH=KTH
    DO 20 I=1,NX
      IC1=1
      IC2=1
      IF((KKXE/2)*2.EQ, KKXE) IC1=0
      IF((KKTH/2)*2.EQ, KKTH) IC2=0
      I1=NX+1-I
      KDXE(I1)=IC1
      KDTH(I1)=IC2
      KKXE=KKXE/2
      KKTH=KKTH/2
20
C
C  MAIN LOOP
C
30  IF(NS-NUSA) 32,100,100
32  IF(ISWD) 36,34,36
34  CALL WAITFR(IEV1)
     IEV1=0
     GO TO 38
36  CALL WAITFR(IEV3)
38  IPLE=0
C
     IF(ICNT-IPER) 80,40,40
C
C  SCALE PROCESS INPUTS AND OUTPUTS, CHANGE ORDER
C  OF PROCESS OUTPUTS IF THE MODEL INCLUDES SERVO
C  DYNAMICS
C
40  DO 41 I=1,NU
41  UP(I)=U(I)/GAIN
     IF(NE-7) 42,46,42
42  DO 44 I=1,7
44  YP(I+1)=Y(I)/GAIN
     YP(1)=Y(8)/GAIN
     YP(9)=Y(9)/GAIN
     GO TO 50
C
46  DO 48 I=1,7
48  YP(I)=Y(I)/GAIN
C
C  COMPUTE ESTIMATION ERRORS
C
50  I1=0
     DO 52 I=1,NX
       IF(KDXE(I).EQ.0) GO TO 52
       I1=I1+1
       E(I1)=YP(I1)-XE(I)
52  CONTINUE
C
C  MASK OUT MEASURABLE OUTPUTS
C
     I1=0
     I2=0
     DO 60 I=1,NX
       IF(KDXE(I).EQ.1) I1=I1+1
       IF(KDTH(I).EQ.0) GO TO 60
       I2=I2+1
       YRE(I2)=YP(I1)
60  CONTINUE
C
C  COMPUTE NEW ESTIMATION
C
CALL MOVE(XE(1),XE0(1),NX+NX)

```



```
C      DO 70 I=1,NY
70     RE(I)=YRE(I)-SCAPRO(TH(I,1),9,XE(1),1,NX)
      DO 72 I=1,NX
72     XE(I)=SCAPRO(FI(I,1),17,XE0(1),1,NX)+
*SCAPRO(GA(I,1),17,UP(1),1,NU)+
*SCAPRO(AK(I,1),17,RE(1),1,NY)

C      IPLE=1
      ICNT=0
      NS=NS+1

C      80   ICNT=ICNT+1
      IEV3=0
      IF(ICNT.EQ.IPER) IEV4=1

C      IF DATASWITCH 1 IS SET THE EXPERIMENT IS STOPPED
C      IF(ISENSW(1)) GO TO 100

C      GO TO 30

C      READY

C      100  CALL UNFIX(4HALIO)
      CALL CANCEL(4HALIO)
      ISWK=0
      IPLE=0
      IEV4=0

C      RETURN

C      END
```

## SUBROUTINE PRTSKE

PRINTS DESCRIPTION OF A KALMANFILTER

AUTHOR: TORSTEN CROMNOW 1974-06-26

SUBROUTINE REQUIRED  
NONE

DIMENSION FNAME(4),AMON(12),ITIME(6)

COMMON /INDI/ NUSA,KXE,KTH,UI(2)  
COMMON /KALM/ NX,NU,NY,FI(17,17),GA(17,2),  
\*TH(9,17),AK(17,9)  
COMMON /IOCOM/ IDUM(184),USER(36)

EQUIVALENCE (NE,IDUM(18)),(ITICK,IDUM(19)),  
\*(FNAME(1),USER(3))

DATA AMON /3HJAN,3HFEB,4HMARS,5HAPRIL,3HMAJ,  
\*4HJUNI,4HJULY,3HAUG,4HSEPT,3HOCT,3HNOV,3HDEC/

INITIATE

LUO=36  
CALL DATE(ITIME)  
K=ITIME(1)

PRINT KALMANFILTER

WRITE(LUO,100) ITIME(2),AMON(K),ITIME(3)  
WRITE(LUO,102) NX,NU,NY  
WRITE(LUO,104) NE,ITICK  
WRITE(LUO,106) KXE,KTH

WRITE(LUO,108) FNAME(1)  
DO 10 I=1,NX  
WRITE(LUO,110) (FI(I,J),J=1,NX)

WRITE(LUO,108) FNAME(2)  
DO 20 I=1,NX  
WRITE(LUO,110) (GA(I,J),J=1,NU)

WRITE(LUO,108) FNAME(3)  
DO 30 I=1,NY  
WRITE(LUO,110) (TH(I,J),J=1,NX)

WRITE(LUO,108) FNAME(4)  
DO 40 I=1,NX  
WRITE(LUO,110) (AK(I,J),J=1,NY)

100 FORMAT('1DESCRIPTION OF KALMAN FILTER'2X,18,1X,A5,18,  
\*/1X,9('!\*'),9('!\*'),9('!\*'),!\*')  
102 FORMAT(////' NX=',18,5X,'NU=',18,5X,'NY=',18//)  
104 FORMAT(//' NE=',18,4X,'ITICK=',18)  
106 FORMAT(///' KXE=',18,5X,6X,'KTH=',18)  
108 FORMAT(////' MATRIX ',A5//)

```
      E14.  
110  FORMAT(1X,7E15.6/1X,7E15.6/1X,3E15.6)  
C  
      RETURN  
C  
      END
```