Innehållsförteckning.

Abstract.

The main purpose of this report has been to examine a method for solving the optimization problem for linear systems known as the method of diagonal decomposition. The purpose of this method is to avoid some drawbacks of the ordinary methods, especially the need for very great computer storage capacity. The method is used on systems described by the state equation $x(t+1)=\Phi x(t)+\Gamma u(t)$ where $\Gamma$ is a diagonal matrix, which means that each control signal affects only one state variable.

The idea is to decompose the matrix $\Phi$ according to $\Phi = \Lambda + D$ where $\Lambda$ contains the diagonal elements of $\Phi$ and D the remainder of $\Phi$. The state equation then becomes $x(t+1)=\Lambda x(t)+\Gamma u(t)+Dx(t)$. If $x^0(t)$ is an approximate guess of the optimal trajectory the equation will be $x(t+1)=\Lambda x(t)+\Gamma u(t)+e(t)$ where $e(t)=Dx^0(t)$. This equation contains n (n is the dimension of the state variable x) independent scalar equations because both $\Lambda$ and $\Gamma$ are diagonal matrices. This means that the computational problems are reduced very much. Of course the computed optimal trajectory $x'(t)$ will in general not be the correct one because an approximate state equation has been used, but one might hope to get a better value than $x^0(t)$. If then $x'(t)$ is substituted for $x^0(t)$ and the process repeated it would seem possible to reach the correct, optimal trajectory after several iterations.

However, tests of the method show that the convergence is towards a trajectory that is not the optimal one. It is shown in the report that this is due to a principal deficiency in the method. This is also the case when the method is applied to continous systems.

This report also contains simulations of the optimal control of systems of the following types

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

$$\dot{x}(t)=Ax(t)+Bu(t)+e(t)$$

with quadratic performance criteria. In the continous case the most important example is one where the equations describe the temperature of a rod where u is the temperature at one end and e the temperature at the other end. Simulations are made to see what happens if the optimal feedback-law computed for a certain e(t) is used when the actual disturbance e(t) is a different one. The answer appears to be that no drastic change of the performance of the system happens.

## Inledning.

Avsikten med examensarbetet var att undersöka en metod för optimering av linjära system, den s.k. diagonala dekomponeringsmetoden. Metoden avsågs kunna användas för godtyckliga kostnadsfunktioner men undersökningen begränsades till fallet med kvadratisk förlustfunktion. Metodens avsikt är att komma undan en del praktiska problem främst i fråga om behov av stort minnesutrymme vid användande av dator som dyker upp vid användande av konventionella metoder. Idén är att för system av typen $x(t+1)=\Phi x(t)+\Gamma u(t)$ där $\Gamma$ är en diagonalmatris sönderlägga $\Phi$ enligt $\Phi=\Lambda+D$ där $\Lambda$ innehåller diagonalelementen och D övriga element i $\Phi$. Ekvationen blir då $x(t+1)=\Lambda x(t)+\Gamma u(t)+Dx(t)$. Om $x^{o}(t)$ är en gissad approximativ lösning fås den approximativa ekvationen $x(t+1)=\Lambda x(t)+\Gamma u(t)+e(t)$ där $e(t)=Dx^{o}(t)$, som sönderfaller i n st (där n är dimensionen på tillståndsvektorn x) skalära ekvationer eftersom både $\Lambda$ och $\Gamma$ är diagonala. Detta betyder att räkningarna förenklas drastiskt. Eftersom en approximativ ekvation använts blir den framräknade optimala banan $x'(t)$ naturligtvis inte i allmänhet den korrekta, men man kan hoppas att den är bättre än $x^{o}(t)$. Genom att använda $x'(t)$ som nytt värde på $x^{o}(t)$ och upprepa metoden skulle man kunna iterera sig fram till den korrekta lösningen.

Examensarbetets uppgift var att skriva ett program för metoden och undersöka dess konvergensegenskaper på några exempel. Härvid konstaterades att metoden visserligen konvergerade (åtminstone för vissa matriser $\Phi$ ) men mot en bana som avvek från den optimala. I kapitel 2 visas genom en teoretisk analys att detta beror på en principiell felaktighet hos metoden. I kapitel 5 undersöks metoden applicerad på kontinuerliga system varvid resultatet blir väsentligen detsamma.

Eftersom diagonala dekomponeringsmetoden ger ekvationen $x(t+1)=Ax(t) + \Gamma u(t) + e(t)$ som avviker från den normala linjära tillståndsekvationen genom att innehålla termen $e(t)$ föll det sig naturligt att ytterligare studera sådana system. Program för lösande av det linjärkvadratiska optimeringsproblemet för fallet med e-term har gjorts både för det diskreta och det kontinuerliga fallet och ett antal simuleringar har genomförts.

## 1. Översikt av teorin för optimering av linjär-kvadratiska diskreta system.

Följande system betraktas:

$$(1.1)\quad \begin{cases} x(i+1)=\Phi x(i)+\Gamma u(i)+e(i) & i=1,\ldots,L \\ x(1)=c \end{cases}$$

där $\Phi$ är en nxn-matris och $\Gamma$ en nxr-matris.

e(i) är en vektor som representerar en känd störning av systemet.

Till systemet hör en kvadratisk förlustfunktion:

$$(1.2)\quad x^T(L)Q_0x(L)+\sum_{j=1}^{L-1}(x^T(j)Q_1x(j)+u(j)^TQ_2u(j))$$

Problemet är att finna en sekvens av styrsignaler så att (1.2) minimeras.

Inför $V(x,j)=\min_u\left(\sum_{k=j}^{L-1}\{x^T(j)Q_1x(j)+u(j)^TQ_2u(j)\}+x^T(L)Q_0x(L)\right)$

Då gäller

$$(1.3)\quad V(x,j)=\min_u(x^T(j)Q_1x(j)+u(j)^TQ_2u(j)+V(x(j+1),j+1))$$

och $V(x(L),L)=x^T(L)Q_0x(L)$

Ansatsen

$$(1.4)\quad V(x(j),j)=x^T(j)R(j)x(j)+x^T(j)S(j)+\varphi(j)$$

där R är en nxn-matris, S är en n-vektor och $\varphi$ en skalär ger efter insättning i (1.3) och identifikation av termer av typen $x^T(\;)x$, $x^T(\,)$ och skalära termer

$$(1.5)\quad u(j)=-(Q_2+\Gamma^T R(j+1)\Gamma)^{-1}\Gamma^T(R(j+1)\Phi x(j)+ +R(j+1)e(j)+S(j+1)/2)$$

och följande rekursionsformler:

$$(1.6)\quad R(j)=\Phi^T R(j+1)\Phi+Q_1- -\Phi^T R(j+1)\Gamma(Q_2+\Gamma^T R(j+1)\Gamma)^{-1}\Gamma^T R(j+1)\Phi$$

$$(1.7)\quad S(j)=(\Phi^T-\Phi^T R(j+1)\Gamma(Q_2+\Gamma^T R(j+1)\Gamma)^{-1}\Gamma^T)(2R(j+1)e(j)+ +S(j+1))$$

$$(1.8)\quad \varphi(j)=e^T R(j+1)e+e^T S(j+1)+\varphi(j+1)- -(R(j+1)e(j)+S(j+1)/2)^T\Gamma(Q_2+\Gamma^T R(j+1)\Gamma)^{-1}\Gamma^T(S(j+1)/2+ +R(j+1)e(j))$$

Randvärden blir:

$$(1.6')\quad R(L)=Q_0$$

$$(1.7')\quad S(L)=0$$

$$(1.8')\quad \varphi(L)=0$$

Observera att $\varphi$ inte behöver vara känd för att u skall kunna beräknas. Däremot kan $\varphi$ användas för att beräkna kostnaden för den optimala banan enligt

$$(1.9)\quad V(x(1),1)=x^T(1)R(1)x(1)+x(1)^T S(1)+\varphi(1)$$

u kan enligt (1.5) beräknas ur återkopplingslagen

$$(1.10)\quad u=-L(j)x(j)-M(j) \quad \text{där}$$

$$(1.11)\quad L(j)=(Q_2+\Gamma^T R(j+1)\Gamma)^{-1}\Gamma^T R(j+1)\Phi$$

$$(1.12)\quad M(j)=(Q_2+\Gamma^T R(j+1)\Gamma)^{-1}\Gamma^T(R(j+1)e(j)+S(j+1)/2)$$

Optimeringsproblemet kan alltså lösas numeriskt genom att (1.6) och (1.7) itereras baklänges. För varje steg beräknas och lagras L(j) och M(j). Ekvation (1.1) kan sedan itereras framlänges genom insättning av u enligt (1.10). Ett problem vid stora system är att lagringen av L(j) och M(j) för varje tidpunkt kräver stort minnesutrymme. Dessutom behövs en matrisinversion vid varje steg.

Ett försök att komma undan dessa svårigheter är metoden med diagonal dekomponering som berörs i nästa avsnitt.

## 2. Diagonala dekomponeringsmetoden.

Metoden används för det fall att systemet är

$$(2.1)\quad x(t+1)=\Phi x(t)+\Gamma u(t) \qquad t=1,2...L$$
$$x(0)=c$$

där $\Gamma$ är en diagonalmatris, dvs varje styrsignal påverkar precis en komponent av x.

Dessutom ges kostnadsfunktionen av

$$(2.2)\quad x^T(L)Q_0x(L)+\sum_{j=1}^{L-1} x^T(j)Q_1x(j)+u^T(j)Q_2u(j)$$

där $Q_0, Q_1, Q_2$ är diagonalmatriser.

Om nu $\Phi$ vore en diagonalmatris skulle de olika komponenterna av x vara helt okopplade och problemet skulle kunna lösas genom att optimeringen gjordes för varje komponent för sig. Diagonala dekomponeringsmetodens idé är därför (se [ 4 ] ) att sönderlägga $\Phi$ enligt $\Phi=\Lambda+D$ där $\Lambda$ innehåller diagonalelementen och D övriga element. Systemet ser då ut som:

$$(2.3)\quad x(t+1)=\Lambda x(t)+\Gamma u(t)+Dx(t)$$

I termen Dx(t) införs i stället för x(t) en gissad optimal bana $x^0(t)$. Man får då:

$$(2.4)\quad x(t+1)=\Lambda x(t)+\Gamma u(t)+Dx^0(t)$$

eller i komponentform

(2.5) $x_i(t+1)=\lambda_i x_i(t)+\gamma_i u_i(t)+e_i(t)$

där $e_i(t)=\sum_k d_{ik}x_k^0(t)$

och $\lambda_i$ är i:te diagonalelementet i $\Lambda$

$\gamma_i$ " " $\Gamma$

$d_{ik}$ elementet i i:te raden, j:te kolonnen i D

Ekvationerna (2.5) löses med hjälp av metoderna i föregående avsnitt. Lösandet blir här speciellt enkelt eftersom ekvationerna är skalära och matrisinversionerna övergår i division. Det behövliga minnesutrymmet reduceras drastiskt. Det värde på den optimala banan som erhålls x) får sedan ersätta det gissade värdet $x^0$ och proceduren upprepas på nytt. Man får alltså en serie succesiva approximationer av den optimala banan. Beräkningarna fortgår tills skillnaden mellan succesivt beräknade banor blivit tillräckligt liten.

En förutsättning för att diagonala dekomponeringsmetoden skall fungera är naturligtvis följande: Om den optimala banan till (2.1),(2.2) insättes som $x^0$ skall den nya bana som beräknas vara just den optimala. För att kontrollera om detta är uppfyllt är det lämpligt att angripa problemet på följande sätt.

x) Ekvationen (2.5) används bara vid beräkning av R och S. Vid rekonstruktion av banan används den ursprungliga ekvationen (2.1).

Enligt [1] sid 2:3 kan optimeringsproblemet (2.1), (2.2) lösas genom att Hamiltonfunktionen

(2.6) $H^j = x^T(j)Q_1x(j) + u^T(j)Q_2u(j) + \lambda^T(j+1)(\Phi x(j) + \Gamma u(j) + e(j))$

införes, varvid följande ekvationer gäller:

(2.7) $\frac{\partial H^j}{\partial u(j)} = 0$

(2.8) $\lambda(j) = \frac{\partial H^j}{\partial x(j)}$

(2.9) $\lambda(L) = \frac{\partial}{\partial x(L)}(x^T(L)Q_0x(L))$

Ekvation (2.7) ger $u(j) = -\frac{1}{2}Q_2^{-1}\Gamma^T\lambda(j+1)$

(2.8) ger $\lambda(j) = \Phi^T\lambda^T(j+1) + 2Q_1x(j)$

(2.9) ger $\lambda(L) = 2x^T(L)Q_0$

Efter insättning av u(j) i (2.1) fås

(2.10) $x(j+1) = \Phi x(j) - \frac{1}{2}\Gamma Q_2^{-1}\Gamma^T\lambda(j+1) + e(j)$

(2.11) $\lambda(j) = 2Q_1x(j) + \Phi^T\lambda(j+1)$

(2.12) $x(0) = c$

(2.13) $\lambda(L) = 2Q_0x(L)$

Med den diagonala dekomponeringen fås alltså följande ekvationer:

(2.14) $x(j+1) = \Lambda x(j) - \frac{1}{2}\Gamma Q_2^{-1}\Gamma^T\lambda(j+1) + Dx^0(j)$

(2.15) $\lambda(j) = 2Q_1x(j) + \Lambda^T\lambda(j+1)$

(2.16) $x(0) = c$

(2.17) $(L) = 2Q_0x(L)$

Å andra sidan ger en direkt användning av (2.10)-(2.13) på systemet att värdena på $x(j)$ och $\lambda(j)$ på den optimala banan, $x_{opt}(j)$ resp. $\lambda_{opt}(j)$ uppfyller:

(2.18) $x_{opt}(j+1)=\Phi x_{opt}(j)-\frac{1}{2}\Gamma Q_2^{-1}\Gamma^T \lambda_{opt}(j+1)$

(2.19) $\lambda_{opt}(j)=2Q_1 x_{opt}(j) + \Phi^T \lambda_{opt}(j+1)$

(2.20) $x_{opt}(0)=c$

(2.21) $\lambda_{opt}(L)=2Q_o x_{opt}(L)$

Antag att diagonaldekomponeringsmetoden ger korrekt resultat dvs antag att (2.14)-(2.17) ger $x(j)=x_{opt}(j)$.

Man får då ur (2.14)

(2.22) $x_{opt}(j+1)=\Lambda x_{opt}(j)-\frac{1}{2}\Gamma Q_2^{-1}\Gamma^T\lambda(j+1)+Dx_{opt}(j)=$

$\Phi x_{opt}(j)-\frac{1}{2}\Gamma Q_2^{-1}\Gamma^T\lambda(j+1)$

Om diagonalmatrisen $\Gamma$ förutsättes ha alla element $\neq 0$ bestämmer denna ekvation $\lambda(j)$ entydigt ur $x(j)$ och $x(j+1)$. En jämförelse med (2.18) ger då att

$\lambda(j)=\lambda_{opt}(j)$ för $j= 1,..L$

(2.15) ger då

$\lambda_{opt}(j)=2Q_1 x_{opt}(j)+\Lambda\lambda_{opt}(j+1)$

vilket tillsammans med (2.19) ger

(2.23) $(\Phi^T-\Lambda)\lambda_{opt}(j)=0$ $j=2,..L$

Ekvationen $u(j)=-\frac{1}{2}Q_2^{-1}\Gamma^T\lambda(j+1)$ visar att för den typ av system som studeras är $\lambda(j)=0$ precis då $u(j)=0$. Utom för de speciella system för vilken den optimala banan erhålles för styrsignalen =0 kan man alltså förutsätta att $\lambda(j)$ i (2.23) är $\neq 0$. Ekvationen (2.23) är naturligtvis sann om $\Phi$ är en diagonalmatris men annars i allmänhet inte, t.ex för n=2 erhålles om $\Phi^T = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

$$\begin{pmatrix} 0 & b \\ c & 0 \end{pmatrix}\begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} b\lambda_2 \\ c\lambda_1 \end{pmatrix} = 0 \qquad \Rightarrow \qquad b=c=0$$

Metoden har också testats på några explicita exempel som följer nedan.

Ex.1 $\quad n=2 \quad L=3$

$$\Phi = \begin{pmatrix} 0.86603 & 0.5 \\ -0.5 & 0.86603 \end{pmatrix}$$

$$\Gamma = Q_0 = Q_1 = Q_2 = I$$

Optimala banan:

| j= | u(j) | |
|---|---|---|
| 1 | -.8196149 | -.2196150 |
| 2 | -.2732049 | .0732052 |

| j= | x(j) | |
|---|---|---|
| 1 | 1.0000000 | 1.0000000 |
| 2 | .5464101 | .1464100 |
| 3 | .2732049 | -.0732052 |

kostnaden för optimala banan blir =3.20000

Om den optimala banan används som startvärde i diagonala dekomponeringsmetoden fås efter första iterationen (då man borde fått den optimala banan)

| j= | u(j) | |
|---|---|---|
| 1 | -.8042033 | -.1620982 |
| 2 | -.2798783 | .0482996 |

| j= | x(j) | |
|---|---|---|
| 1 | 1.0000000 | 1.0000000 |
| 2 | .5618217 | .2039269 |
| 3 | .3086367 | -.0560054 |

kostnaden för denna bana blir 3.20931

Om i stället banan hade beräknats ur (2.5), samma ekvation som användes för att beräkna för att beräkna R och S skulle kostnaden blivit 3.19158.

Optimeringen som görs för $\dot{x}=\Lambda x+\Gamma u+Dx_{opt}$ kan naturligtvis ge minst lika låg kostnad som $\dot{x}=\Phi x+\Gamma u$ eftersom man kan välja $x=x_{opt}$, men tydligen kan man genom att välja en något annorlunda bana få lägre kostnader. Då rekonstruktionen göres enligt den korrekta ekvationen (2.1) fås då en suboptimal bana.

Om iterationerna i diagonaldekomponeringen fortsättes fås till slut följande bana:

| j= | u(j) | |
|---|---|---|
| 1 | -.8095300 | -.1611789 |
| 2 | -.2921798 | .0504226 |

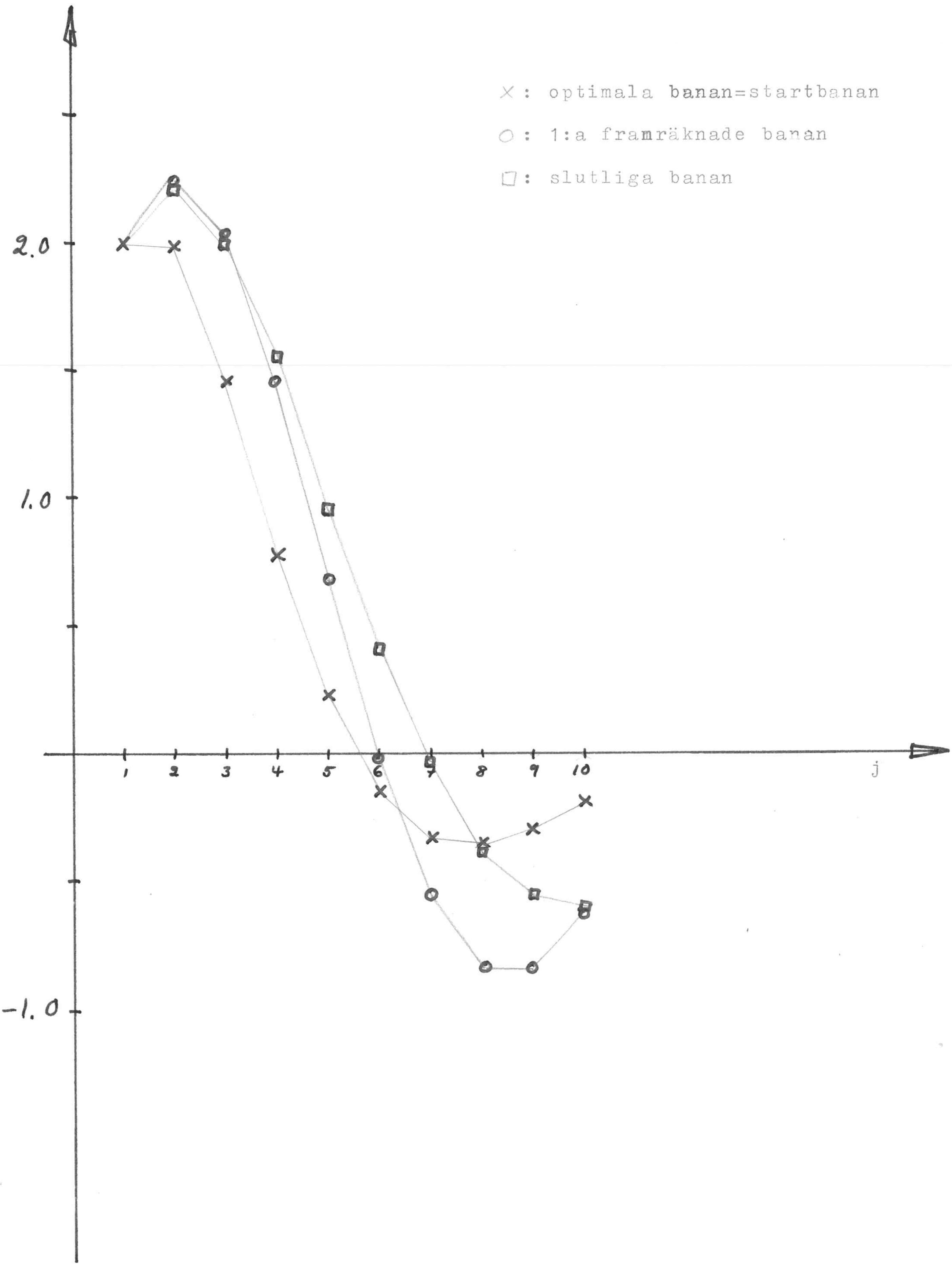
| | x(j) | | |
|---|---|---|---|
| 1 | 1.0000000 | 1.0000000 | |
| 2 | .5564950 | .2048461 | |
| 3 | .2921818 | -.0504231 | kostnad=3.20879 |

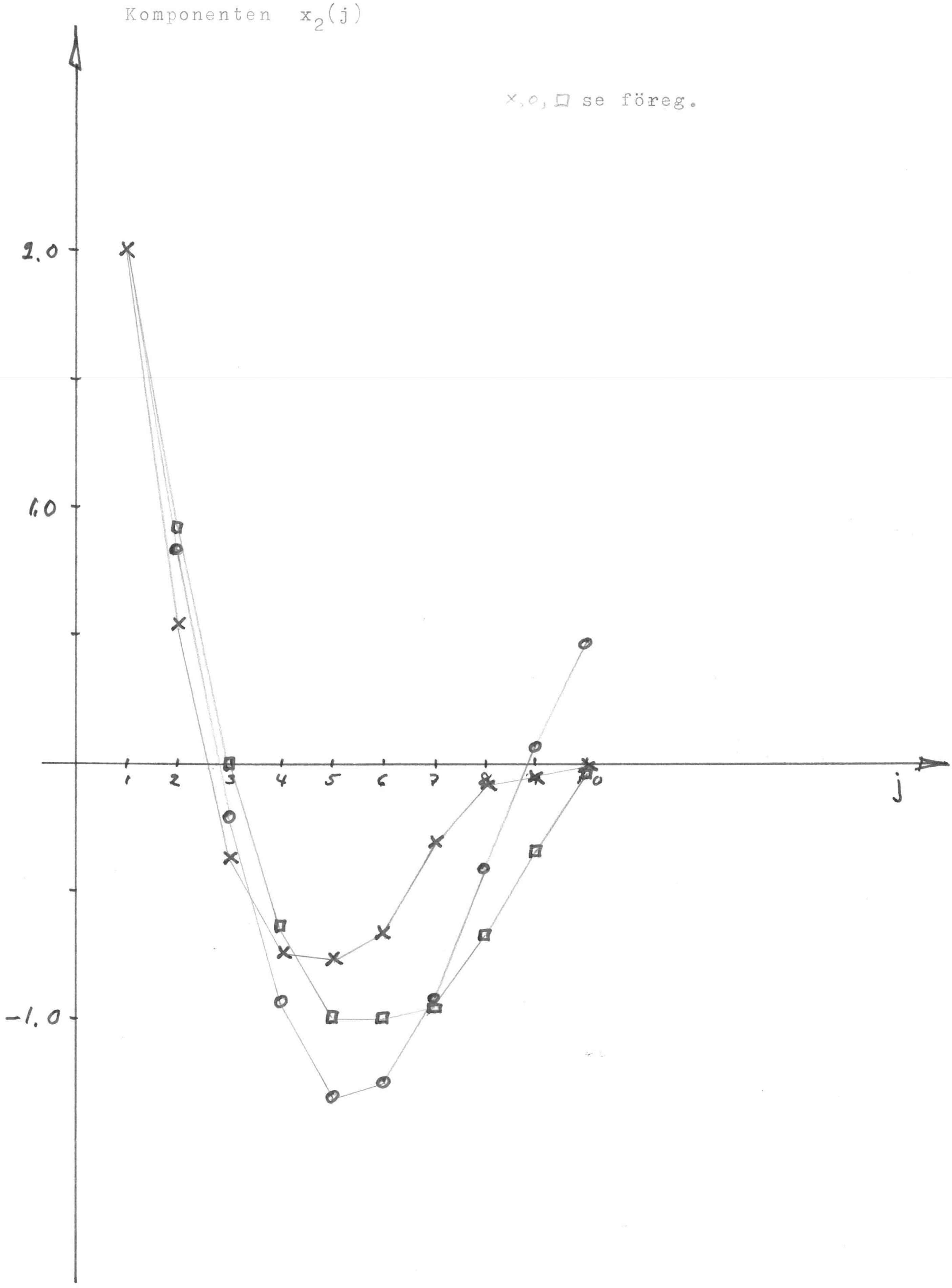
ex.2 $n=2$ $L=10$

$$\Phi=\begin{bmatrix} .86603 & .50000 \\ -.50000 & .86603 \end{bmatrix} \qquad \Gamma=Q_0=Q_1=I \qquad Q_2=10I$$

Optimala banan används som startvärde. På de följande sidorna visas hur de av diagonaldekomponeringsmetoden framräknade banorna ser ut.

Komponenten $x_1(j)$




Komponenten x2(j)
×, o, □ se föreg.
1.0
1.0
-1.0
1
2
3
4
5
6
7
8
10
j

I appendix finns ytterligare exempel på användning av metoden. Sammanfattningsvis kan man alltså säga att metoden i allmänhet inte ger konvergens mot rätt lösning. Dock kan som i exemplet i appendix skillnaden vara ganska liten så att metoden möjligen kan användas för att generera en approximativ bana som sedan tas som utgångspunkt i någon annan iterativ metod.

## 3. Några simuleringar av optimala banor för diskreta system.

Ett program har skrivits för lösande av problemet som beskrivs av ekvationerna (1.1),(1.2), se appendix. Programmet består av huvudprogram samt subrutinerna LBER och EBRKN. Dessutom används standardrutinen SYMIN. Huvuddelen av beräkningsarbetet utförs i LBER. Här itereras ekvationerna (1.6)-(1.8) baklänges varvid inversionen $(Q_2+\Gamma^T R(j+1)\Gamma)^{-1}$ göres av SYMIN och ur de så erhållna R- och S-värdena beräknas L(j) och M(j) (i programmet betecknade LM och RS) enligt (1.11) och (1.12). L(j) och M(j) lagras för j=L-1,L-2,..,1. Dessutom ger LBER som resultat R(1), S(1) och $\varphi(1)$. I huvudprogrammet kan sedan u(1) beräknas ur (10), varefter insättning i systemekvationen (1.1) ger x(2), som genom (1.10) ger u(2) osv. Genom användning av (1.9) kan kostnaden för den optimala banan beräknas. Eftersom denna kostnad dessutom tas fram direkt ur den rekonstruerade banan med hjälp av (1.2) kan man genom att jämföra dessa få en viss uppfattning om noggrannheten i beräkningarna.

Inläsning av data.

Följande data läses in

N,IR,L

där N är antalet tillståndsvektorer

där IR är antalet styrsignaler

där L är antalet tidpunkter (dvs systemet studeras för $j=1,\ldots,L$)

IE som är en parameter som anger hur e(j) skall beräknas

IE=1 betyder att e(j) läses in

IE=2 betyder att e(j) sättes =0 för alla j

IE=3 betyder att e(j) beräknas genom anrop av subrutinen EBRKN.

Matriserna $\Phi$, $\Gamma$, $Q_0$, $Q_1$, $Q_2$ (matriselementen läses in radvis)

XST som är begynnelsevärdet på x.

Om IE=1 läses e(j) sedan in. Är IE=3 behövs eventuellt indata till EBRKN (beroende på vilken beräkning man vill låta EBRKN utföra)

Utskrift göres av alla data som lästs in, och av e(j) oavsett hur den beräknats.

Följande resultat skrivs sedan ut:

$R(j), S(j), \varphi(j) \qquad j=1,..L$

kostnaden för optimala banan enl. $x(1)^T R(1) x(1) + x(1)^T S(1) + \varphi(1)$

$u(j)$ och $x(j)$ $j=1,\ldots L$ på den optimala banan

kostnaden beräknad från denna bana

återkopplingsmatriserna $L(j)$ och $RS(j)$ $j=1,\ldots L$

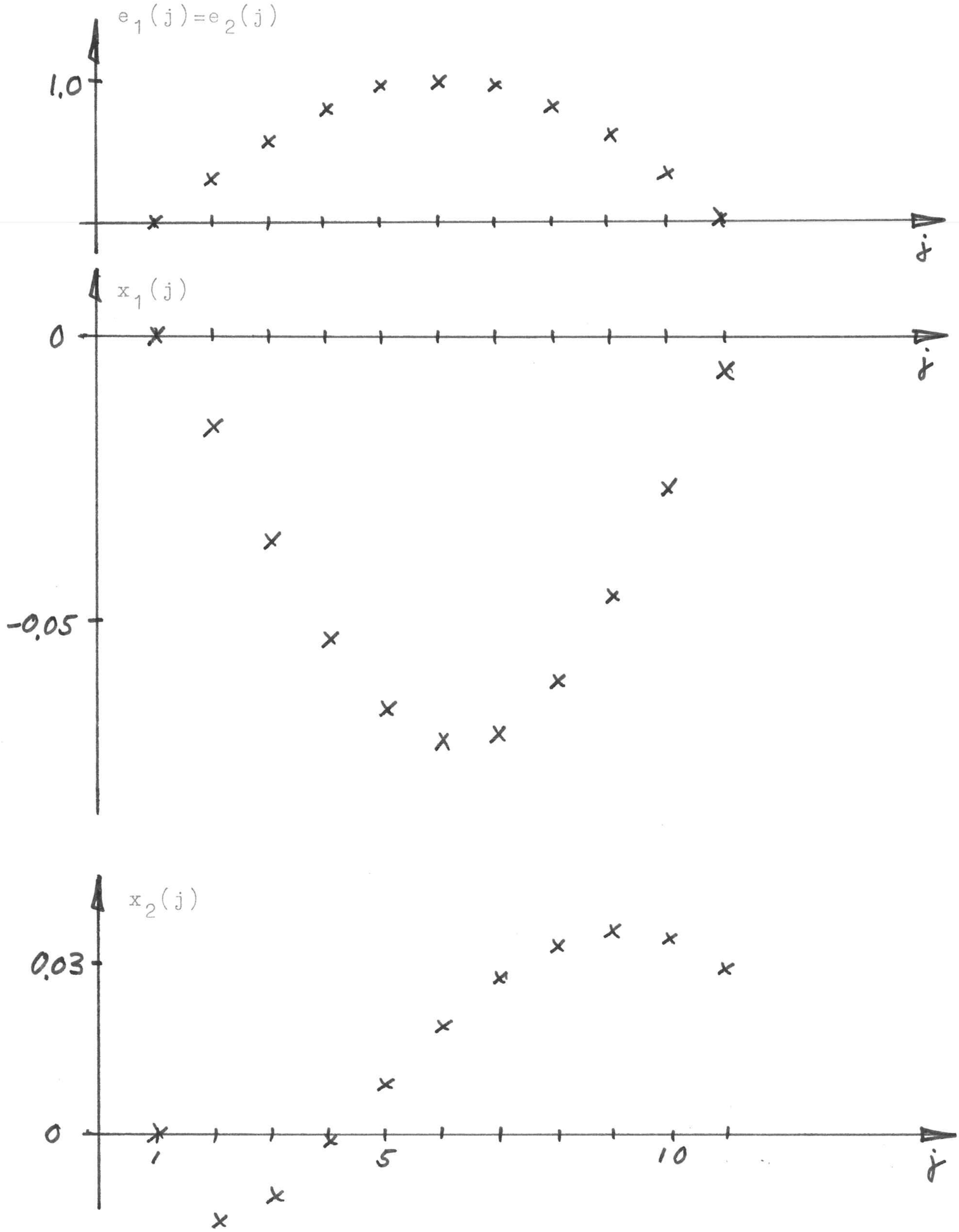
(där $u(j)=-L(j)x(j)-RS(j)$ )

I ex.1 visas ett andra ordningens system där $e_1(t)$ och $e_2(t)$ båda varierar sinusformat i tiden. Båda komponenterna av såväl tillståndsvektorn som styrsignalen bestraffas i kostnadsfunktionen. Den optimala styrsignalen blir som man kan vänta av samma form som e(t) men riktad åt motsatt håll.

Ex.2 visar ett tredje ordningens system med två insignaler. Felsignalen e(t) påverkar alla tre komponenterna av tillståndsvektorn och är konstant i tiden. Kostnadsfunktionen bestraffar alla komponenter av såväl insignal som tillståndsvektor. Som man ser av diagrammen används de inledande styrsignalerna till att pressa ned värdena på komponenterna i tillståndsvektorn till låga värden. Därefter håller sig $u_1$ praktiskt taget konstant på -1 vilket betyder att den precis neutraliserar inverkan av $e_1(t)$ och $e_2(t)$ på $x_1$ resp $x_2$, så att dessa komponenter förblir på låga värden.

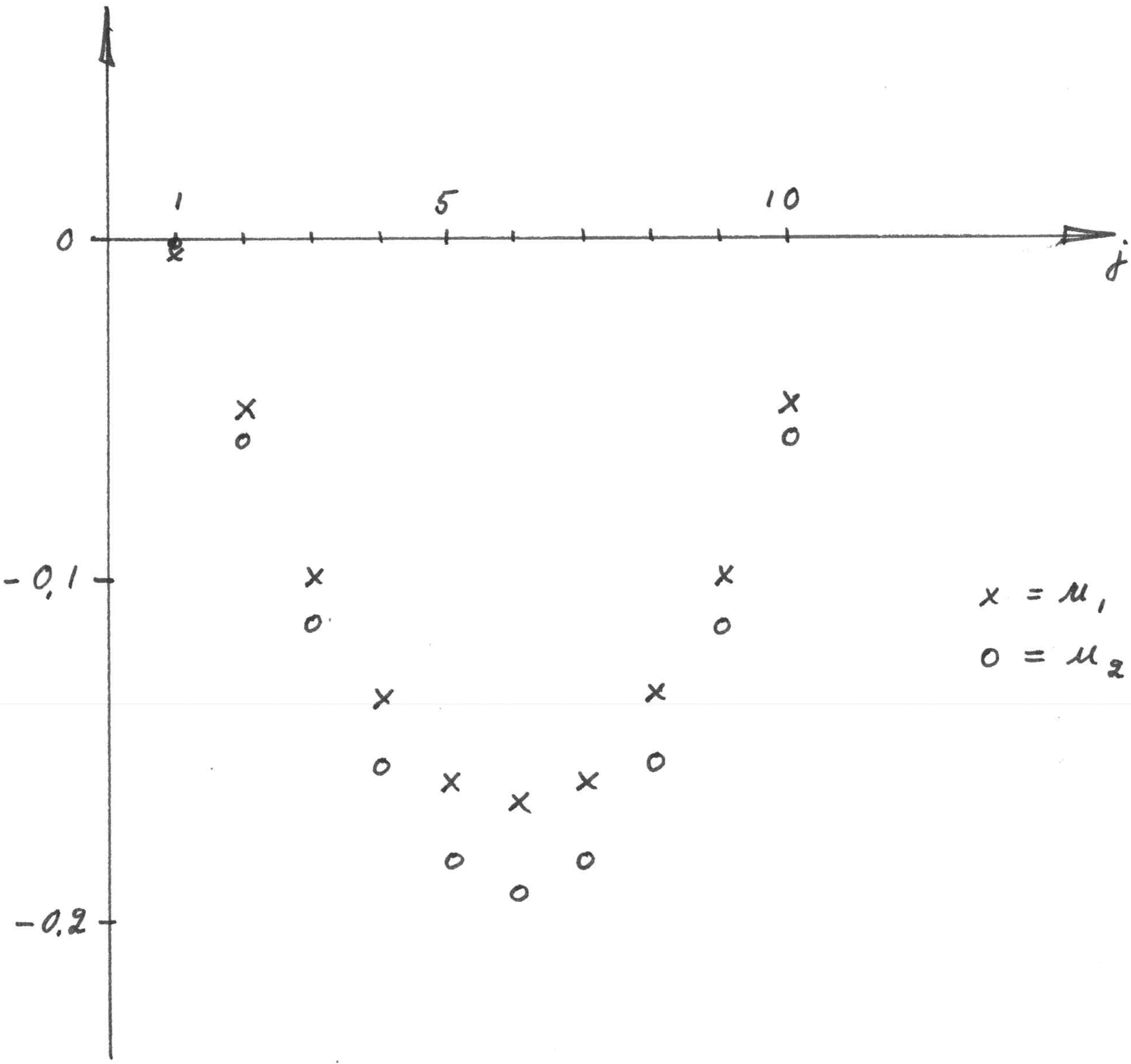
Ex. 1

$$\Phi = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \qquad \Gamma = \begin{pmatrix} 1 & 4 \\ 2 & 3 \end{pmatrix} \qquad Q_0 = Q_1 = Q_2 = I$$



ex 1 forts.

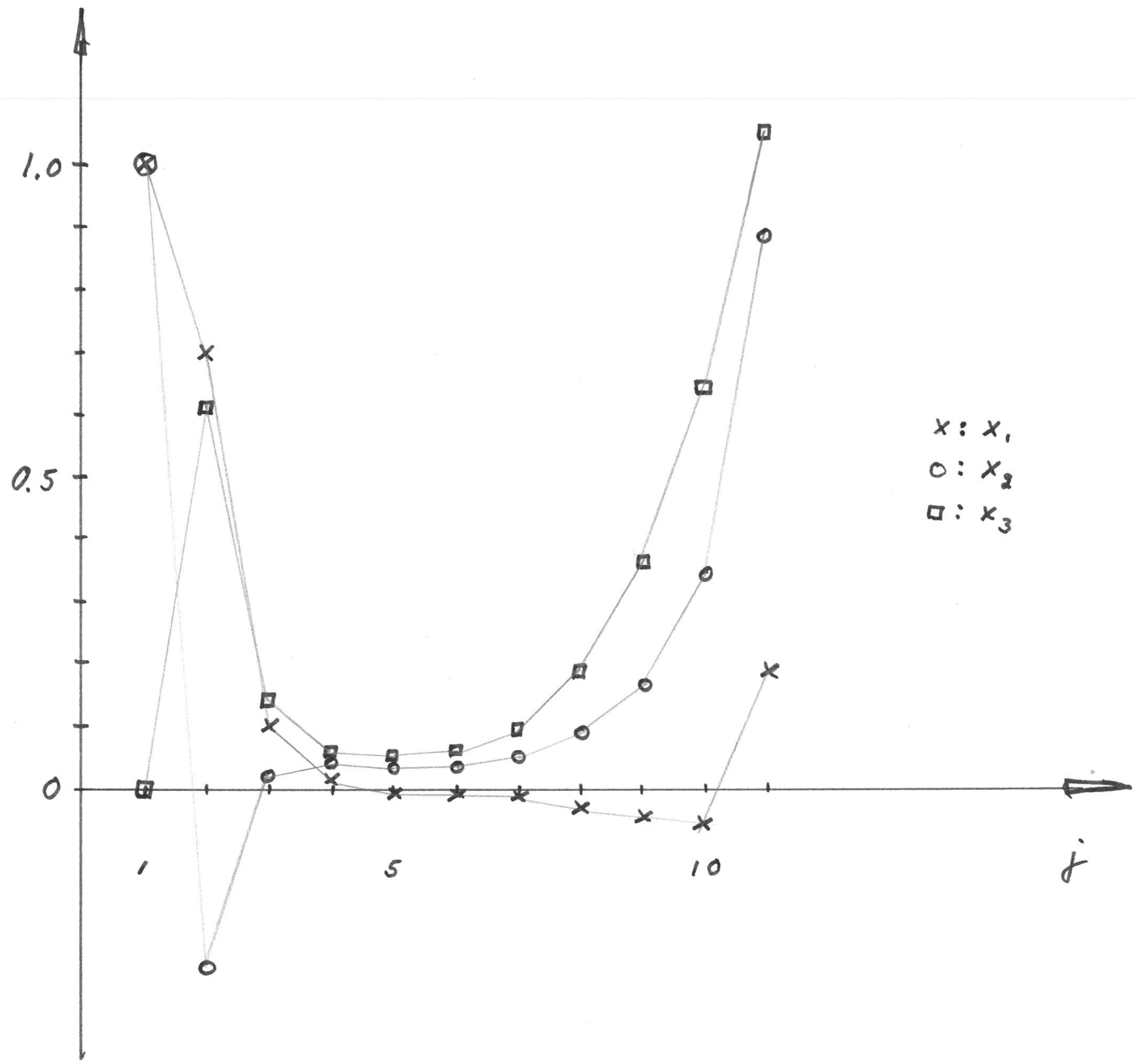




Ex. 2

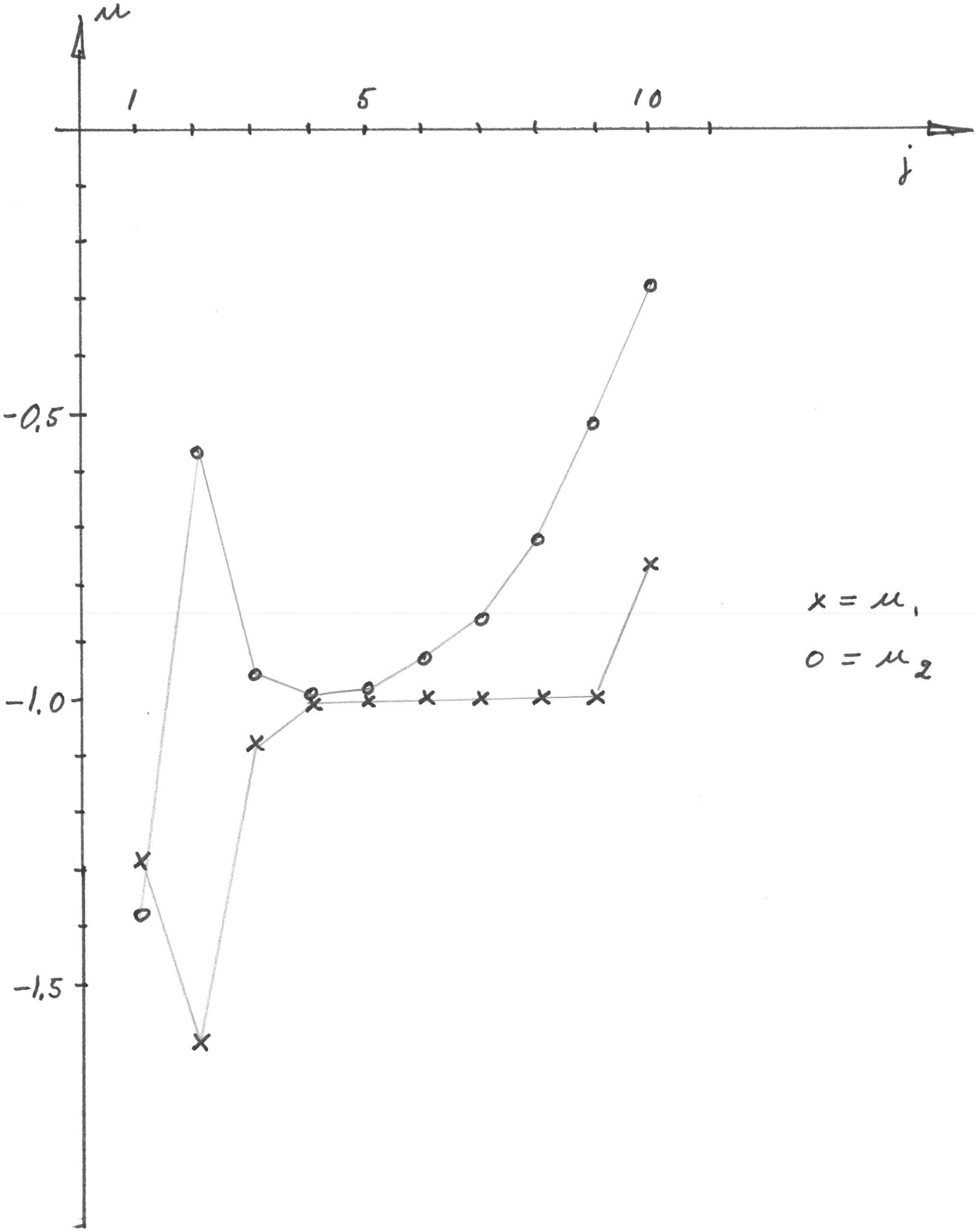
$$\Phi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \Gamma = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad Q_0 = Q_1 = I \quad Q_2 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

$e_1(j) = e_2(j) = e_3(j) = 1$ för alla j



ex 2 forts.





## 4. Översikt av teorin för optimering av lineärkvadratiska kontinuerliga system.

I detta fall beskrivs systemet av följande ekvationer

$$(4.1) \quad \dot{x}(t)=Ax(t)+Bu(t)+e(t) \qquad t_o \le t \le T$$
$$x(0)=c$$

Förlustfunktionen har formen

$$(4.2) \quad x^T(T)Q_o x(T)+\int_{t_o}^{T}(x^T(t)Q_1x(t)+u^T(t)Q_2u(t))dt$$

Enligt [1] kap.4 kan problemet lösas på följande sätt:

Inför $J^o(x,t)=\min_u(x^T(T)Q_o x(T)+\int_{t_o}^{T}(x^T(t)Q_1x(t)+u^T(t)Q_2u(t))dt\ )$

Man får speciellt

$$(4.3) \quad J^o(x,T)=x^T(T)Q_o x(T)$$

Då gäller Hamilton-Jacobi-Bellmans ekvation:

$$(4.4) \quad -\frac{\partial J^o}{\partial t}=\min_u H(x,\frac{\partial J^o}{\partial x},u,t)$$

med randvillkoret (4.3), där H är systemets hamiltonfunktion. För systemet (4.1) blir hamiltonfunktionen

$$(4.4") \quad H(x,\lambda,u,t)=x^T(t)Q_1x(t)+u^T(t)Q_2u(t)+ + \lambda^T(Ax(t)+Bu(t)+e(t))$$

Analogt med det diskreta fallet görs ansatsen

(4.5) $J^o(x,t)=x^T(t)R(t)x(t)+x^T(t)S(t)+\varphi(t)$

där R är en nxn-matris och S en n-kolonnvektor samt $\varphi$ en skalär. Insättning av (4.4') och (4.5) i (4.4) ger att min fås för

(4.6) $u=-Q_2^{-1}B^T(R(t)x(t)+S(t)/2)$

och identifikation av koefficienter ger följande differentialekvationer för R,S och $\varphi$.

(4.7) $\mathring{R}(t)=R(t)BQ_2^{-1}B^TR(t)-R(t)A-A^TR(t)-Q_1 \qquad R(T)=Q_o$

(4.8) $\dot{S}(t)=R(t)BQ_2^{-1}B^TS(t)-A^TS(t)-2R(t)e(t) \qquad S(T)=0$

(4.9) $\dot{\varphi}(t)=\frac{1}{4}S^T(t)BQ_2^{-1}B^TS(t) - S^T(t)e(t) \qquad \varphi(t)=0$

Ekvationerna (4.7)-(4.9) är alltså ett system av ordinära differentialekvationer med randvillkor vid sluttidpunkten T. Efter lösning av detta kan alltså styrsignalen u erhållas ur (4.6) och den optimala banan för systemet beräknas.

## 5. Diagonaldekomponeringsmetoden för kontinuerliga system.

I analogi med det diskreta fallet kan diagonaldekomponeringsmetoden användas på följande typ av kontinuerliga system, se [4] och [5]

(5.1) $\dot{x}=Ax+Bu \qquad t_o \leq t \leq T$

$x(0)=c$

med förlustfunktionen

(5.2) $x^T(T)Q_o x(T)+\int_o^T (x^T(t)Q_1x(t)+u^T(t)Q_2u(t))dt$

$B,Q_o,Q_1,Q_2$ förutsättes vara diagonalmatriser.

A sönderlägges enligt $A=\Lambda+D$ där $\Lambda$ innehåller diagonal- och D övriga element i A.

(5.1) övergår då i

(5.3) $\dot{x}=\Lambda x+Bu+Dx$

Om $x^o$ är ett gissat värde på den optimala banan fås

(5.4) $\dot{x}=\Lambda x+Bu+Dx^o$

som sönderfaller i följande skalära ekvationer:

(5.5) $\dot{x}_i=\lambda_i x_i+\beta_i u_i+e_i$

där $e_i=\sum d_{ik}x_k^o$, $\lambda_i$ element i $\Lambda$, $\beta_i$ element i B och $d_{ik}$ element i D.

Ett villkor för att metoden skall fungera är liksom i det diskreta fallet att om $x^{o}(t)$ är den optimala banan skall metoden som resultat återigen ge den optimala banan.

Dekomponeringen enligt (5.4) leder till att ekvationerna (4.6)-(4.8) får följande utseende:(när diagonaldekomponeringen används betecknas R-matrisen i stället med T)

(5.6) $\dot{T}=TBQ_2^{-1}B^TT-T\Lambda-\Lambda^TT-Q_1$

(5.7) $\dot{S}=TBQ_2^{-1}B^TS-\Lambda^TS-2TDx^o$

(5.8) $u_{min}=-Q_2^{-1}B^T(Tx+S/2)$

Behandling av problemet på konventionellt sätt skulle ge

(5.9) $\dot{R}=RBQ_2^{-1}B^TR-RA-A^TR-Q_1$

(5.10) $u_{min}=-Q_2^{-1}B^TRx$

Om $x=x_{opt}$ användes i (5.7) och det så erhållna värdet på S tillsammans med $x_{opt}$ insättes i (5.8) skall man om metoden är riktig få samma värde på $u_{min}$ som i (5.10)

Om man utgår från antagandet att diagonaldekomponeringsmetoden är riktig skall alltså följande gälla

(5.11) $Q_2^{-1}B^T(Tx_{opt}+S/2)=Q_2^{-1}B^TRx_{opt}$

Om B förutsättes ha alla sina diagonalelement $\neq 0$ fås då

(5.12) $Tx_{opt}+S/2=Rx_{opt}$

Deriveras båda sidor erhålles

$$\dot{T}x_{opt}+T\dot{x}_{opt}+\dot{S}/2=\dot{R}x_{opt}+R\dot{x}_{opt}$$

Insättning av (5.6)-(5.10)och (5.1) ger

$$(5.13)\quad TBQ_2^{-1}B^T(Tx_{opt}+S/2-Rx_{opt})-\Lambda^T(Tx_{opt}+S/2)= =-A^T Rx_{opt}$$

Insättning av sambandet (5.12) ger

$$(5.14)\quad \Lambda^T Rx_{opt}-A^T Rx_{opt}=0$$

Eftersom $Rx_{opt}\neq 0$ precis då u är det enligt (5.10) kan samma resonemang tillämpas som i det diskreta fallet, dvs metoden fungerar bara i vissa specialfall. För n=2 är det bara fallen $u\equiv 0$ och $\Phi$ diagonal, ty

$$\left[\begin{pmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{pmatrix}-\begin{pmatrix}\lambda_1 & a\\ b & \lambda_2\end{pmatrix}\right]Rx_{opt}\neq 0 \text{ om } Rx_{opt}\neq 0 \text{ och a och b } \neq 0.$$

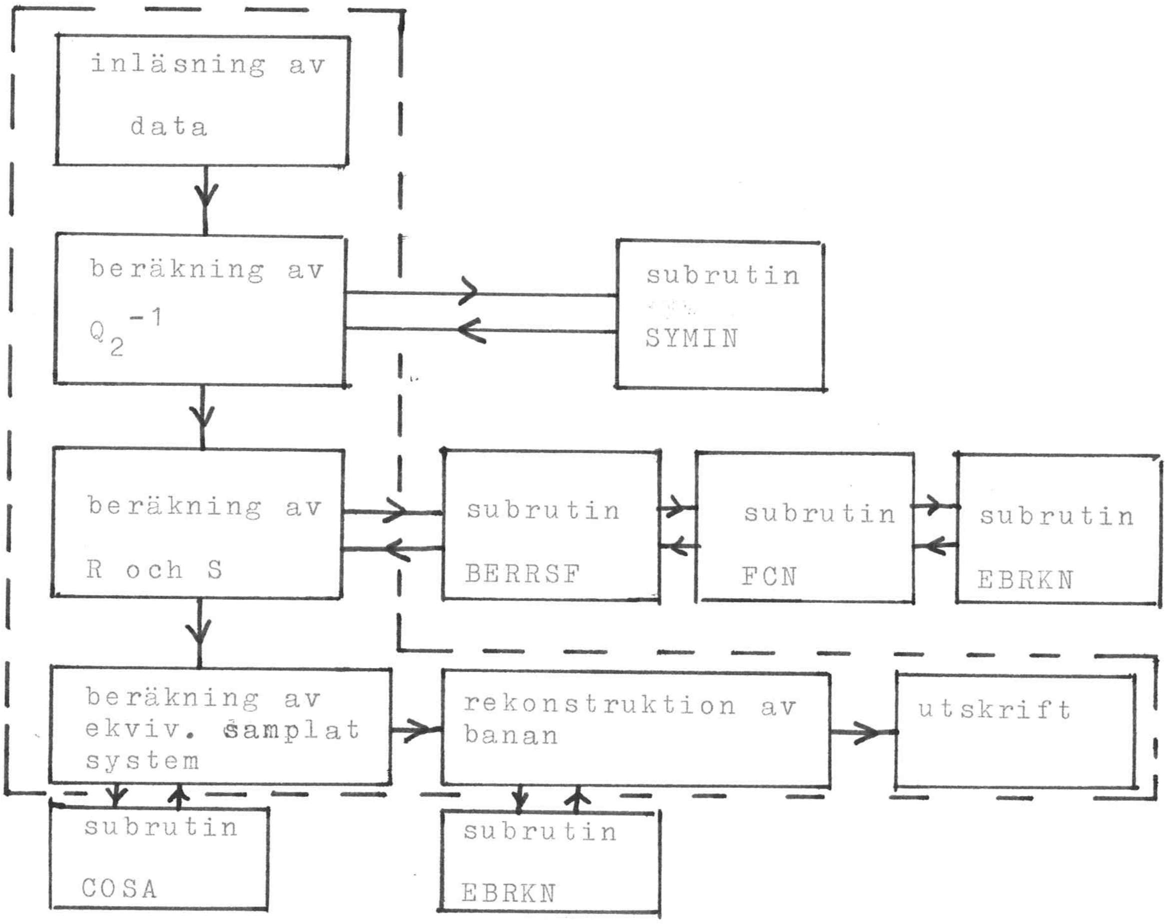
För högre ordningens system kan möjligen ytterligare specialfall tänkas.

# 6. Några simuleringar av optimal styrning av kontinuerliga system.

Numeriskt har det kontinuerliga optimeringsproblemet lösts med utgångspunkt från ekvationerna (4.7)-(4.9). Detta är ett system av 1:a ordningens ordinära differentialekvationer. Eftersom randvärdena alla är givna vid t=T kan de lösas bakåt i tiden från t=T till t=0 med Runge-Kuttas metod, väsentligen på samma sätt som i [2].

Skiss av programmet

huvudprogram



Matrisen $Q_2^{-1}$ beräknas av standardrutinen SYMIN, varefter matriserna $Q_2^{-1}B^T$ (i programmet kallad CON) och $BQ_2^{-1}B^T$ (i programmet kallad RIC) beräknas en gång för alla. Subrutinen BERRSF får som indata R(t), S(t) och $\varphi$(t) och beräknar med hjälp av Runge-Kutta R(t-h), S(t-h) och $\varphi$(t-h). Vid dessa beräkningar måste högerleden i (4.7)-(4.9) evalueras ett flertal gånger. Detta sker i FCN. De värden på e(t) som därvid behövs lämnas av EBRKN. BERRSF anropas upprepade gånger från huvudprogrammet och R och S lagras för tidpunkterna t=T,T-h,...h,0

Rekonstruktionen sker genom att insignalen u(t) och störningen e(t) betraktas som konstanta under tidsintervallen $mh \leq t \leq (m+1)h$ varvid ett samplat system erhålles. Det samplade systemet blir

$$x(t+h)=\Phi x(t)+\Gamma u(t)+\Delta e(t)$$

där $\Phi=\exp(Ah)$, $\Gamma=\int_0^h \exp(As)Bds$ , $\Delta=\int_0^h \exp(As)Ids$

Matriserna $\Phi$, $\Gamma$ och $\Delta$ beräknas genom anrop av standardrutinen COSA. Subrutinen EBRKN innehåller fyra olika alternativ för beräkning av e(t). Heltalet IEC anger genom att anta något av värdena 1-4 vilket alternativ som skall användas i det aktuella fallet. Ett värde på IEC läses in före beräkningen av R, S och $\varphi$ och ett nytt före rekonstruktionen av banan. Man kan alltså beräkna den optimala återkopplingslagen för ett värde på e(t) och simulera systemet för ett annat, vilket görs för värmestaven, se sid 6:8.

Följande data läses in:

N,NB, där N är antalet tillståndsvariabler och NB antalet styrsignaler.

L,TDIST,IAVST där L är antalet och TDIST längden av de intervall som används vid beräkningen med Runge-Kutta. Starttidpunkten blir alltså t=0 och sluttidpunkten L*TDIST. IAVST reglerar utskriften, se nedan.

Matriserna A,B,$Q_o$,$Q_1$,$Q_2$ (matriselementen läses härvid in i ordningen $a_{11}, a_{12} \ldots a_{1n}, a_{21}, a_{22} \ldots a_{2n} \ldots$ ) och startvärdet x(0).

Slutligen skall indata innehålla de två värdena på IEC, se ovan.

Beträffande inläsningsformat se programmet i appendix.

Utskrift görs av alla de data som lästs in och av de beräknade värdena på $\Phi$, $\Gamma$ och $\Delta$. Utskrift görs också av de värden på x(t),u(t) och e(t) som erhålls vid rekonstruktionen av banan och av R(t), S(t) och $\varphi(t)$. Normalt skrivs inte dessa värden ut vid varje tidpunkt som de beräknats eftersom detta skulle betyda väldigt mycket utdata. Om IAVST=1 skrivs de ut vid varje tidpunkt, om IAVST=2 endast vid varannan tidpunkt osv. Dessutom skrivs kostnaden för den optimala banan ut dels beräknad enligt $x(0)^T R(0)x(0)+x(0)^T S(0)+\varphi(0)$, dels beräknad från den rekonstruerade banan.

Som test kördes programmet på följande exempel som kan lösas analytiskt.

ex.1

$$\begin{cases} \dot{x}_1 = x_2 + e(t) \\ \dot{x}_2 = u \end{cases} \qquad 0 \leq t \leq 1 \qquad x_1(0)=1 \quad x_2(0)=0$$

med kostnadsfunktionen

$$\tfrac{1}{2}x_1(1)^2 + \tfrac{1}{2}\int_0^1 \tfrac{1}{2}u^2 dt$$

Man får

$$R(t) = \frac{\frac{1}{2}}{1+\frac{2}{3}(1-t)^3}\begin{pmatrix} 1 & 1-t \\ 1-t & (1-t)^2 \end{pmatrix}$$

och för $e(t)=1$

$$S(t) = \frac{1}{1+\frac{2}{3}(1-t)^3}\begin{pmatrix} 1-t \\ (1-t)^2 \end{pmatrix}$$

$x_1(t)=0.4(t-1)^3-0.2(t-1)+1.2$

$x_2(t)=1.2((t-1)^2-1)$

$u(t)=2.4(t-1)$

jämförelse:

| exakt lösning | lösning beräknad för h=0.001 |
|---|---|
| $R(0)=\begin{pmatrix} 0.3 & 0.3 \\ 0.3 & 0.3 \end{pmatrix}$ | $R(0)=\begin{pmatrix} 0.3000077 & 0.3000013 \\ 0.3000013 & 0.2999981 \end{pmatrix}$ |
| $S(0)=\begin{pmatrix} 0.6 \\ 0.6 \end{pmatrix}$ | $s(0)=\begin{pmatrix} 0.6000025 \\ 0.5999961 \end{pmatrix}$ |

| exakt | beräknad |
|---|---|
| $R(0.5)=\begin{bmatrix}0.4615385 & 0.2307692\\0.2307692 & 0.1153846\end{bmatrix}$ | $R(0.5)=\begin{bmatrix}0.4615422 & 0.2307690\\0.2307690 & 0.1153841\end{bmatrix}$ |
| $S(0.5)=\begin{bmatrix}0.4615385\\0.2307692\end{bmatrix}$ | $S(0.5)=\begin{pmatrix}0.4615380\\0.2307682\end{pmatrix}$ |
| $u(0)=-2.4$ | $u(0)=-2.3999973$ |
| $u(0.5)=-1.2$ | $u(0.5)=-1.1996611$ |
| $x(0.5)=\begin{pmatrix}1.25\\-0.9\end{pmatrix}$ | $x(0.5)=\begin{pmatrix}1.2498742\\-0.9004795\end{pmatrix}$ |
| $u(1)=0.0$ | $u(1)=0.0000000$ |
| $x(1)=\begin{pmatrix}1.2\\-1.2\end{pmatrix}$ | $x(1)=\begin{pmatrix}1.1995192\\-1.2009727\end{pmatrix}$ |

Kostnaden för den optimala banan blir =1.2

Beräknad enligt $x(0)^T R(0)x(0)+x(0)^T S(0)+\varphi(0)$ blir den 1.2000083, beräknad från banan 1.1992825

Med den använda metoden blir alltså noggrannheten vid rekonstruktionen av banan avsevärt sämre än hos R och S.

ex.2 samma system som i föregående exempel men

$e(t)=t$

$R(t)$ blir oförändrad

$$S(t)=\frac{1}{2}\frac{1}{1+\frac{2}{3}(1-t)^3}\begin{pmatrix}1-t^2\\(1+t)(1-t)^2\end{pmatrix}$$

$x_1(t)=0.3(t-1)^3-0.9(t-1)+\frac{1}{2}(t^2-1)+0.9$

$x_2(t)=0.9((t-1)^2-1)$

$u(t)=1.8(t-1)$

I appendix finns en jämförelse mellan den exakta och den framräknade lösningen.

ex.3

$\dot{x}_1=e_1$ $\qquad 0 \leq t \leq 1$

$\dot{x}_2=x_1+u+e_2$ $\qquad x_1(0)=0 \quad x_2(0)=0.5 \qquad e_1=2 \qquad e_2=\frac{1}{2}$

med kostnadsfunktionen

$$\frac{1}{2}(x_1(1)^2+x_2(1)^2)+\frac{1}{2}\int_0^1 (x_1(t)^2+u(t)^2)\,dt$$

Man får

$$R(t)=\begin{pmatrix}1-t+\frac{1}{2(2-t)} & \frac{1-t}{2(2-t)}\\ \frac{1-t}{2(2-t)} & \frac{1}{2(2-t)}\end{pmatrix}$$

$$S(t)=\frac{1}{2-t}\begin{pmatrix}(1-t)(2t^2-7.5t+7.5)\\(1-t)(1.5-t)\end{pmatrix}$$

$x_1(t)=2t$

$x_2(t)=t^2-0.5t+0.5$

$u(t)=-1$

Som jämförelsen i appendix visar är överensstämmelsen i x och u här bättre än i föregående exempel. Detta beror antagligen på att u och e är konstanta, vilket gör att man inte inför något fel genom övergången till samplat system vid beräkningen av banan.

Analys av optimal reglering av temperaturen i en värmestav.

En stav av längden l betraktas. Om $\nu$ är temperaturen och z lägeskoordinaten gäller värmeledningsekvationen

$$(6.1)\quad \frac{\partial^2 \nu}{\partial z^2} = \frac{1}{a^2}\frac{\partial \nu}{\partial t} \qquad \text{där } a^2 = \frac{K}{\rho c}$$

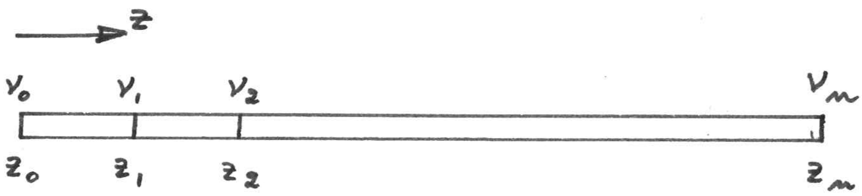
K=värmeledningsförmågan

c=spec. värmet

$\rho$=densiteten

Om man inför $t'=t/T'$, där $T'=l^2/a^2$ fås ekvationen på formen

$$(6.1')\quad \frac{\partial^2 \nu}{\partial z^2} = \frac{1}{l^2}\frac{\partial \nu}{\partial t'}$$




Om staven indelas i n delar med vardera längden h fås följande approximation.

$$\left(\frac{\partial^2 \nu}{\partial z^2}\right)_{z=z_i} = \frac{\nu(z_{i+1}) - 2\nu(z_i) + \nu(z_{i-1})}{h^2}$$

Med beteckningen $\nu_i = \nu(z_i)$ fås då

$$\frac{d\nu_i}{dt'} = n^2(\nu_{i+1} - 2\nu_i + \nu_{i-1}) \qquad i=1,..,n-1$$

Speciellt undersöks fallet n=4 med tidsskalningen $t''=n^2t'$
Tillståndsvariablerna $x_i$ införs enligt $x_i=\nu_i$, i=1,2,3.
Dessutom införs $u=\nu_0$ och $e=\nu_4$, vilket ger

$$(6.2)\quad \dot{x} = \begin{bmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} e$$

Här betraktas u som styrvariabel och e som en störning. Syftet med systemet antas vara att hålla mittpunktens temperatur nära 0 trots störningen e, varför följande kostnadsfunktion införs.

$$\int_0^T (10x_2(t)^2 + u(t)^2)\,dt + 10x_2(T)$$

Avsikten med simuleringen på detta system har varit att se vad som händer om R och S beräknas för ett visst e t.ex. ett steg och systemet sedan styrs enligt den av dessa R och S bestämda styrlagen när det verkliga e har annat utseende.

I fig. 6.1 visas en simulering av systemet för fallet att e(t)=konst=10. Kostnaden för detta fall blir 25.661
I fig 6.2 finns motsvarande simulering för fallet $e(t)=$ $=10-10(1-t)^2$. I detta fall blir kostnaden 9.069. De prickade kurvorna visar det resultat man får om man har $e(t)=$ $=10-10(1-t)^2$ som ovan men använder den återkopplingslag som beräknades för fallet e(t)=10. Avvikelserna blir bara för u och $x_1$ tillräckligt stora för att synas i diagrammet, för exakta resultat se appendix. Kostnaden för denna bana blir 9.342
Fig. 6.3 visar fallet e(t)=10(1-t). De prickade kurvorna visar liksom ovan vad som händer vid styrning enligt de R och S som framräknats i fallet e(t)=10. Denna bana ger kostnaden 9.927 medan den riktiga, optimala styrningen ger 9.038.

Tydligen ger körning av systemet "med fel e" inte i detta fall någon drastisk förändring.

fig 6.1

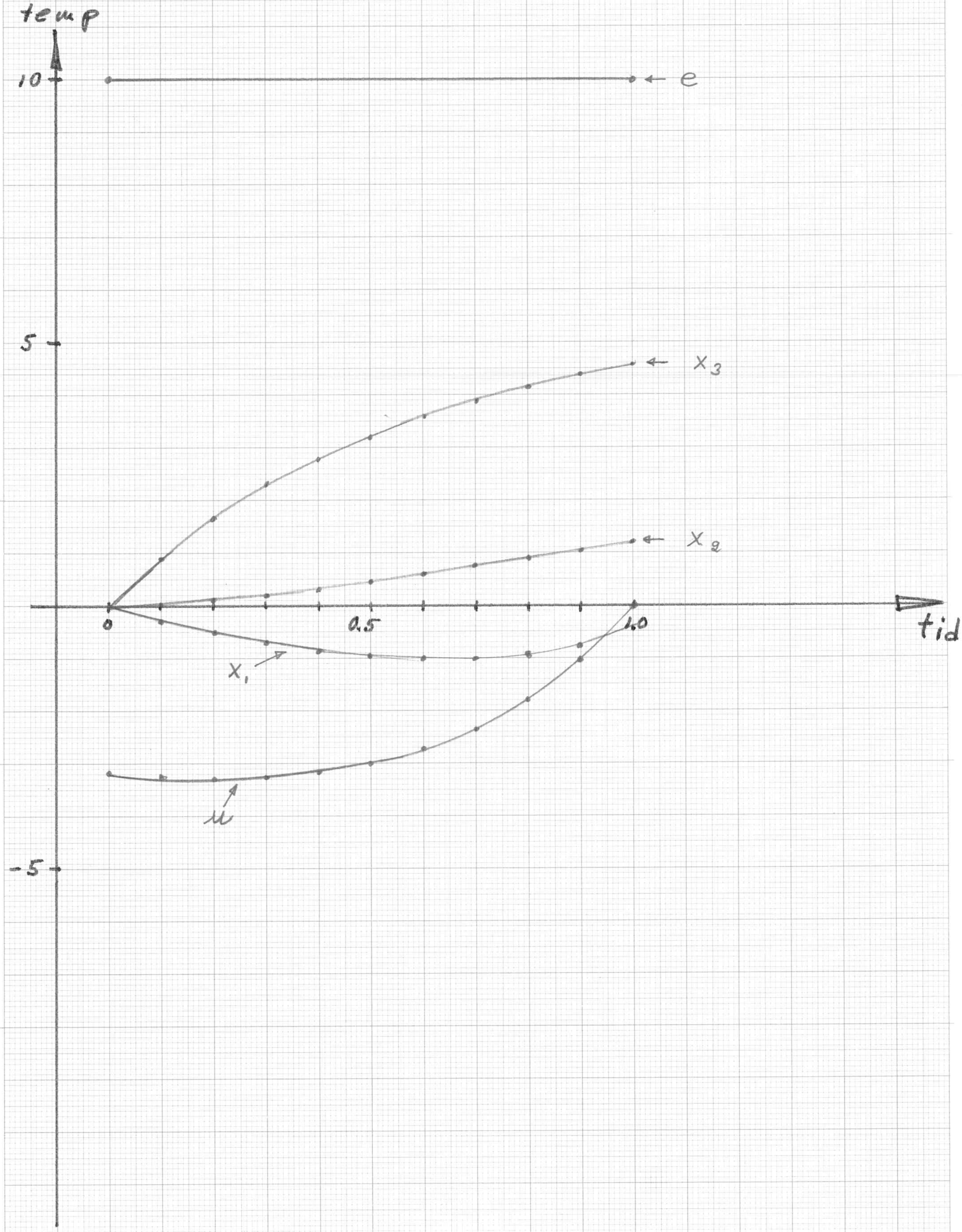


fig 6.2

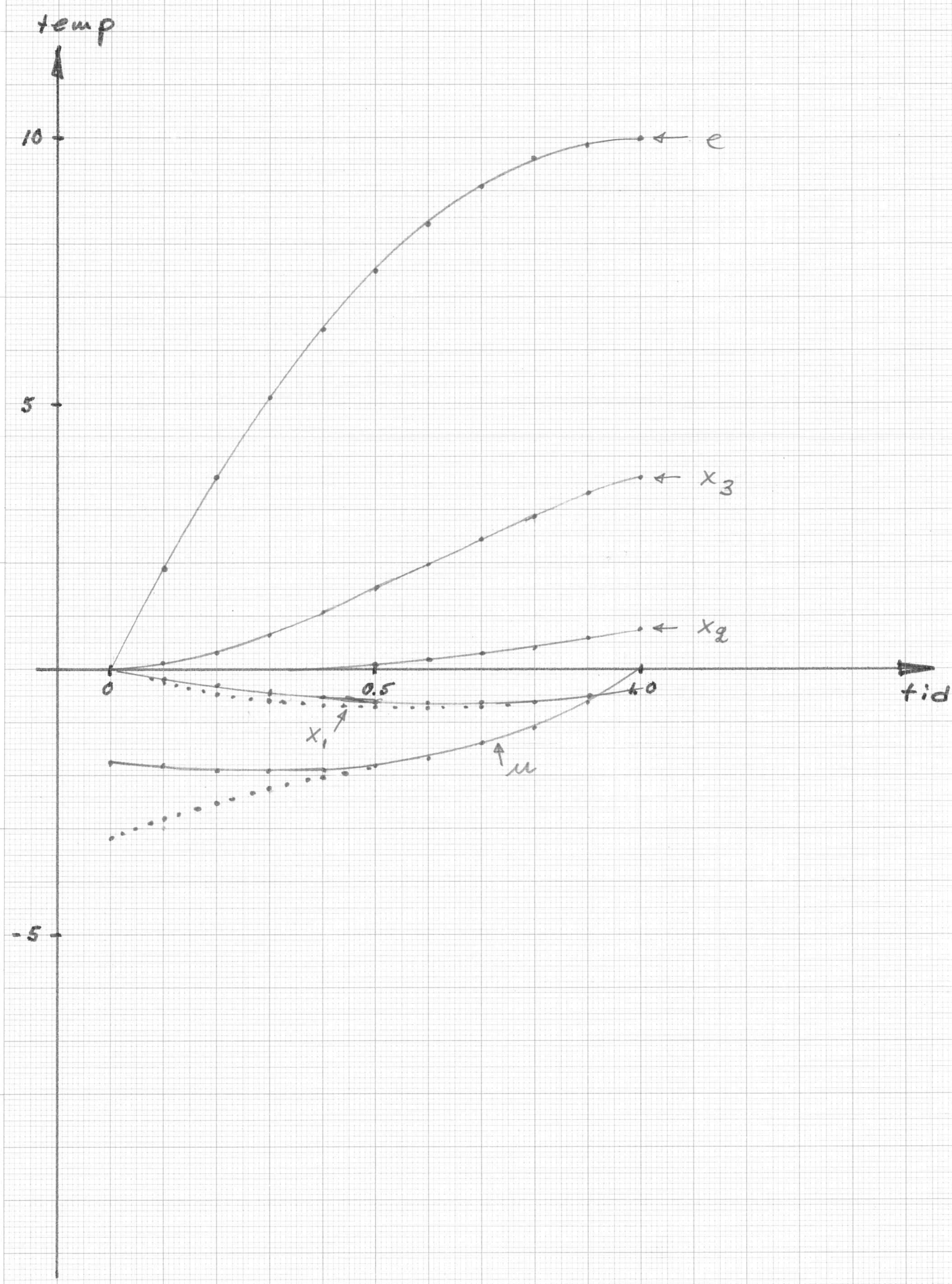
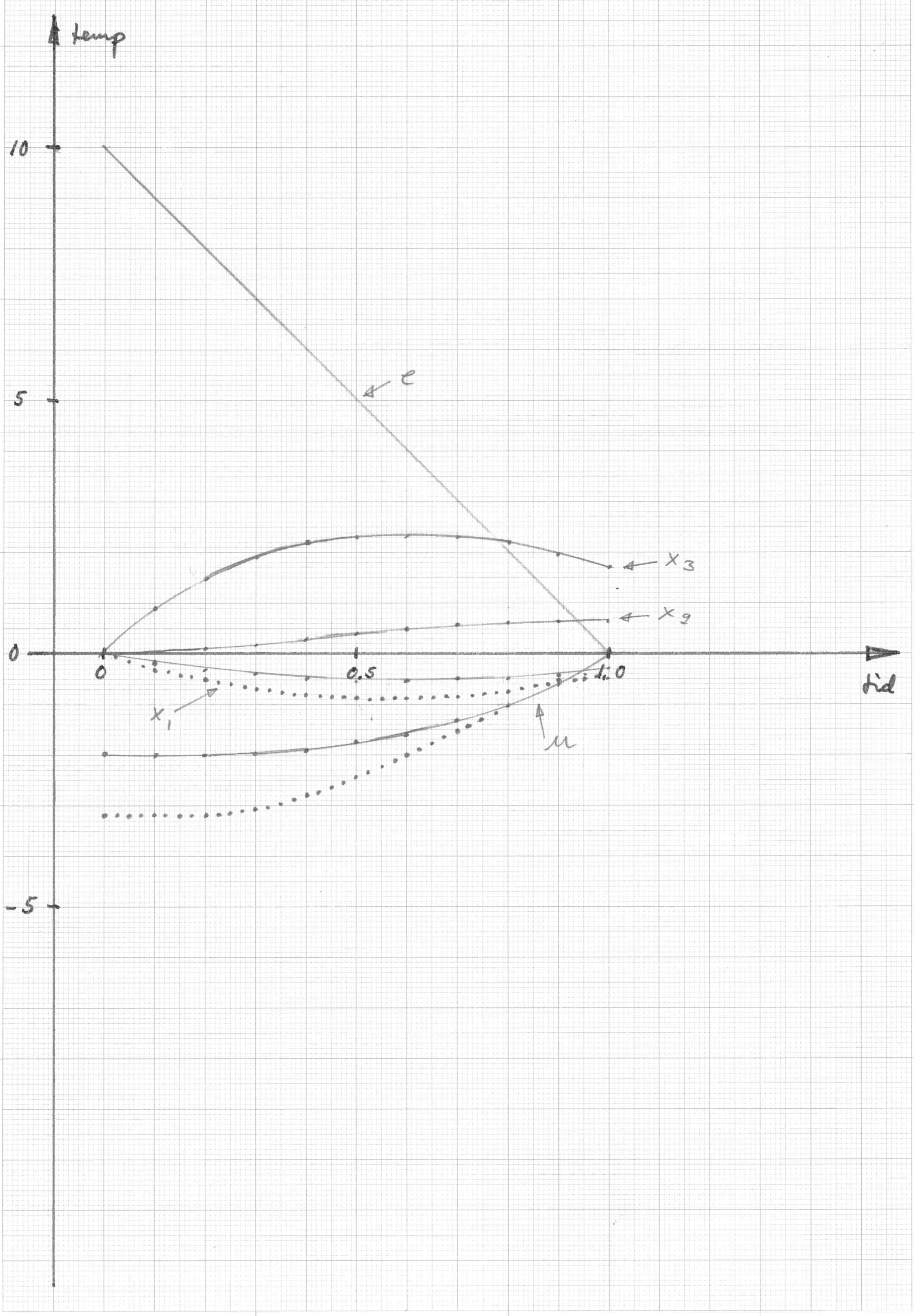


fig 6.3




514 A4
SIS 73 25 01
TULLBERG RST KLIPPAN
Nr 1624

Referenser.

1. Bryson-Ho: Applied Optimal Control.
Gin-Blaisdell 1969.

2. K. Mårtensson: Linear quadratic control package
Part I - The continous problem.
Report 6802 april 1968, Lund Institute of Technology,
Division of Automatic Control.

3. K. Mårtensson: Linear quadratic control package
Part II - The discrete problem.
Report 6904 febr 1969, Lund Institute of Technology,
Division of Automatic Control.

4. David C. Collins: Reduction of dimensionality in dynamic
programming via the method of diagonal decomposition.
University of Southern California, Technical Report,
aug 1969, USCEE-377.

5. David C. Collins, Art Lew: Dimensional approximation in
dynamic programming: Application of structural
decomposition to the optimal control of diffusion
processes.
University of Suothern California, Technical Report
No. 69-5, sept 1969.

# APPENDIX.

A:1

Program för optimering av diskreta system.

```
      DIMENSION R(10,10),S(10)
      DIMENSION FI(10,10), GM(10,10), XST(10), Q0(10,10), Q1(10,10)
      DIMENSION Q2(10,10), U(10,101), X(10,101), E(10,101), RS(10,101)
      REAL LM(10,10,101)
       COMMON R,S,AF
      READ 1000, N,IR,L
      READ 1001,IE
 1001 FORMAT (I5)
 1000 FORMAT (I5,I5,I5)
      READ 1004,((FI(I,J),J=1,N),I=1,N)
      READ 1004,  ((GM(I,J),J=1,IR),I=1,N)
      READ 1004,  ((Q0(I,J),J=1,N ),I=1,N)
      READ 1004,  ((Q1(I,J),J=1,N ),I=1,N)
      READ 1004,  ((Q2(I,J),J=1,IR),I=1,IR)
      READ 1004,(XST(I),I=1,N)
      GO TO (101,102,104),IE
  101 READ 1004,((E(I,J),I=1,N),J=1,L)
      GO TO 105
  102 DO 103 I=1,N
      DO 103 J=1,L
  103 E(I,J)=0
      GO TO 105
  104 CALL EBRKN(N,L,E)
  105 CONTINUE
 1004 FORMAT (8F10.5)
      PRINT 2000, N,IR,L
 2000 FORMAT (/1X,7HDIM. X=,I3,5X,7HDIM. U=,I3,5X,10HINTERVALL=,I4)
 2003 FORMAT(10F11.7)
      PRINT 2004
 2004 FORMAT(/1X,3HFI=)
      DO 1 I=1,N
    1 PRINT 2003, (FI(I,J),J=1,N)
      PRINT 2005
 2005 FORMAT (/1X,6HGAMMA=)
      DO 2 I=1,IR
    2 PRINT 2003, (GM(I,J),J=1,IR)
      PRINT 2006
 2006 FORMAT (/1X,3HQ0=)
      DO 3 I=1,N
    3 PRINT 2003, (Q0(I,J),J=1,N)
      PRINT 2007
 2007 FORMAT(/1X,3HQ1=)
      DO 4 I=1,N
    4 PRINT 2003, (Q1(I,J),J=1,N)
      PRINT 2009
 2009 FORMAT (/1X,3HQ2=)
      DO 6 I=1,IR
    6 PRINT 2003, (Q2(I,J),J=1,IR)
      PRINT 2008, (XST(I),I=1,N)
 2008 FORMAT (/10X,7HXSTART=/10F11.5)
      PRINT 2049
 2049 FORMAT(//1X,1HE)
      DO  5  J=1,L
    5 PRINT 2050, J , (E(I,J),I=1,N)
 2050 FORMAT(//1X,I3,3X,10F11.5)
C
C
```

A:2

```
      CALL LDER(N,IR,L,10,10,101,FI,GM,E,Q0,Q1,Q2,LM,RS)
C
C
      FKST=AF
      DO 11 I=1,N
      DO 10 J=1,N
   10 FKST=FKST+XST(I)*R(I,J)* XST(J)
   11 FKST=FKST+XST(I)*S(I)
      PRINT 2400,FKST
 2400 FORMAT(/1X,28HKOSTNAD ENL. XT*R*X+XT*S+FI=,F15.7)
      LL=L-1
      DO 12  K=1,N
   12 X(K,1)=XST(K)
      SK=0
      DO 50 J=1,LL
      DO 20 I=1,IR
      U(I,J)=0
      DO  18  K=1,N
   18 U(I,J)=U(I,J)+LM(I,K,J)*X(K,J)
   20 U(I,J)=-U(I,J)-RS(I,J)
      DO 21 I=1,N
      DO 21 K=1,N
   21 SK=SK+X(I,J)*Q1(I,K)*X(K,J)
      DO 22 I=1,IR
      DO 22 K=1,IR
   22 SK=SK+U(I,J)*Q2(I,K)*U(K,J)
      DO 30  I=1,N
      X(I,J+1)=0
      DO  25  K=1,N
   25 X(I,J+1)=X(I,J+1)+FI(I,K)*X(K,J)
      DO  27  K=1,IR
   27 X(I,J+1)=X(I,J+1)+GM(I,K)*U(K,J)
   30 X(I,J+1)=X(I,J+1)+E(I,J)
   50 CONTINUE
      DO 40 I=1,N
      DO 40 K=1,N
   40 SK=SK+X(I,L)*Q0(I,K)*X(K,L)
C
C
      PRINT 2100
 2100 FORMAT(//1X,1HJ,10X,1HU)
      DO  60  J=1,LL
   60 PRINT 2101, J,(U(I,J),I=1,IR)
 2101 FORMAT(1X,I3,2X,10F11.7)
      PRINT 2150
 2150 FORMAT(//1X,1HJ,10X,1HX)
      DO  70  J=1,L
   70 PRINT 2151,J,(X(I,J),I=1,N)
 2151 FORMAT (1X,I3,2X,10F11.7)
      PRINT 2152,SK
 2152 FORMAT (/1X,8HKOSTNAD=,F15.7)
      DO 80 J=1,LL
      PRINT 2200,J
 2200 FORMAT(//1X,2HJ=,I3,10X,2HL=)
      DO 80 K=1,IR
   80 PRINT 2201, (LM(K,I,J),I=1,N)
 2201 FORMAT (1X,10F11.7)
      DO 90 J=1,LL
      PRINT 2300,J
 2300 FORMAT (//1X,2HJ=,I3,10X,3HRS=)
      DO 90 K=1,IR
   90 PRINT 2301, RS(K,J)
 2301 FORMAT (1X,10F11.7)
      STOP
      END
```

```
      SUBROUTINE LBER(N,IR,L,ID,IDR,IDT,FI,GM,E0,Q0,Q1,Q2,LM,RS)
      DIMENSION FI(ID,ID),GM(ID,IDR),E0(ID,IDT),RS(IDR,IDT)
      DIMENSION Q0(ID,ID),Q1(ID,ID),Q2(IDR,IDR),V(10,10),R(10,10),S(10)
      DIMENSION Z(10)
       DOUBLE PRECISION W(10,10)
      REAL LM(IDR,ID,IDT)
       COMMON R,S,AF
      DO 5  I=1,N
      DO 5 J=1,N
    5 R(I,J  ) =Q0(I,J)
      DO  7  I=1,N
    7 S(I)=0
      AF=0
      LL=L-1
      DO 100 J=LL,1,-1
      DO  10  I=1,N
      DO  10  K=1,IR
      V(I,K)=0
      DO 10  M=1,N
   10 V(I,K)=V(I,K) + R(I,M) * GM(M,K)
      DO  15  I=1,IR
      DO  15  K=1,IR
      W(I,K)=0
      DO  15  M=1,N
   15 W(I,K)=W(I,K) + GM(M,I) * V(M,K)
      DO  17  I=1,IR
      DO  17  K=1,IR
   17 W(I,K)=Q2(I,K) + W(I,K)
      CALL DSYMIN(IR,10,IFAIL,W)
      IF (IFAIL-1) 109,19,109
   19 PRINT 2000,J
 2000 FORMAT (/1X,31HINVERSIONEN MISSLYCKADES VID J=,I5)
  109 CONTINUE
      DO  20  I=1,IR
      DO  20  K=1,N
      V(I,K)=0
      DO 20  M=1,IR
   20 V(I,K)=V(I,K) + W(I,M) * GM(K,M)
      DO  23  I=1,IR
      RS(I,J)=0
      DO  23  M=1,N
   23 RS(I,J)=RS(I,J) + V(I,M) * S(M) * 0.5
      DO 25  I=1,IR
      DO  25  K=1,N
      W(I,K)=0
      DO  25  M=1,N
   25 W(I,K)=W(I,K) + V(I,M) * R(M,K)
      DO  30  I=1,IR
      DO  30  M=1,N
   30 RS(I,J) =RS(I,J)+W(I,M) * E0(M,J)
      DO 35 I=1,IR
      DO 35  K=1,N
      LM(I,K,J)=0
      DO 35  M=1,N
   35 LM(I,K,J) = LM(I,K,J) + W(I,M) * FI(M,K)
      DO  40  I=1,N
      DO 40  K=1,N
      W(I,K)=0
      DO  40  M=1,IR
   40 W(I,K)=W(I,K) + GM(I,M) * V(M,K)
      DO 42 I=1,N
      Z(I)=0
      DO 41 M=1,N
   41 Z(I)=Z(I)+R(I,M)*E0(M,J)
   42 Z(I)=Z(I)+0.5*S(I)
```

```
      DO 80 I=1,N
      DO 80 M=1,N
   80 AF=AF-Z(I)*W(I,M)*Z(M)+ E0(I,J)*R(I,M)*E0(M,J)
      DO 82 I=1,N
   82 AF=AF+E0(I,J)*S(I)
      DO  45  I=1,N
      DO  45  K=1,N
      V(I,K)=0
      DO  45  M=1,N
   45 V(I,K)=V(I,K) + R(I,M) * W(M,K)
      DO 50  I=1,N
      DO 50 K=1,N
      W(I,K)=0
      DO  50  M=1,N
   50 W(I,K)=W(I,K) + FI(M,I) * V(M,K)
      DO  55  I=1,N
   55 Z(I)=2.0*Z(I)
      DO 60  I=1,N
      S(I)=0
      DO  60  M=1,N
   60 S(I)=S(I) + (FI(M,I)-W(I,M))*Z(M)
      DO  65  I=1,N
      DO  65  K=1,N
      V(I,K)=0
      DO  65  M=1,N
   65 V(I,K)=V(I,K) + (FI(M,I)-W(I,M))*R(M,K)
      DO  70  I=1,N
      DO  70  K=1,N
      R(I,K)=0
      DO  68  M=1,N
   68 R(I,K)=R(I,K) + V(I,M)*FI(M,K)
   70 R(I,K)=R(I,K) + Q1(I,K)
      PRINT 2005,J
 2005 FORMAT(/1X,2HJ=,I3,10X,2HR=)
      DO 75 I=1,N
   75 PRINT 2010, (R(I,K),K=1,N)
 2010 FORMAT(1X,10F11.7)
      PRINT 2015,J
 2015 FORMAT(/1X,2HJ=,I3,10X,2HS=)
      DO 90 I=1,N
   90 PRINT 2010, S(I)
      PRINT 2020, AF
 2020 FORMAT(/1X,4HLFI=,F15.7)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EBRKN(N,L,E)
      DIMENSION E(10,101),AK(10),AL(10)
      READ 1000,AMP,OMEGA,IT
 1000 FORMAT (8F10.5)
      DO 10 I=1,N
      DO 10 J=1,L
   10 E(I,J)=AMP*SIN(OMEGA *(J-IT-1))
      RETURN
      END
```

```
C THIS PROGRAM COMPUTES THE OPTIMAL TRAJECTORY OF THE SYSTEM
C DX/DT=A*X+B*U+E  BY USING THE HAMILTON-JACOBI-BELLMAN METHOD
C THE PERFORMANCE INDEX IS  X(TF)*Q0*X(TF)+INTEGRAL(X(T)*Q1*X(T)+
C +U(T)*Q2*U(T))DT
C
C    N IS DIMENSION OF X
C   NB IS DIMENSION OF U
C
      DIMENSION R(5,5,1002),S(5,1002)
      REAL LFI(1002)
      DIMENSION A(10,10),B(10,10),Q0(10,10),Q1(10,10),Q2(10,10)
      DIMENSION E(10),RIC(10,10),CON(10,10),RIN(10,10),RUT(10,10)
      DIMENSION SI (10),SUT(10), DELTA(10,10),X(10),U(10)
      COMMON /MAIN/A,Q1,RIC /EB/E /EE/IEC
      READ 1000,N,NB
 1000 FORMAT (I5,I5)
      READ 1001,L,TDIST,IAVST
 1001 FORMAT (I5,F20.10,I5)
      READ 1005,((A(I,J),J=1,N),I=1,N)
      READ 1005,((B(I,J),J=1,NB),I=1,N )
      READ 1005,((Q0(I,J),J=1,N ),I=1,N )
      READ 1005,((Q1(I,J),J=1,N),I=1,N)
      READ 1005,((Q2(I,J),J=1,NB),I=1,NB)
      READ 1005,(X(I),I=1,N)
 1005 FORMAT(8F10.5)
      PRINT 2000
 2000 FORMAT (1H1,10HINPUT DATA)
      PRINT 2001,N,NB
 2001 FORMAT(/1X,2HN=,I5,25X,3HNB=,I5)
      PRINT 2002,L,TDIST
 2002 FORMAT(/1X,20HNUMBER OF INTERVALS=,I5,25X,18HLENGTH OF INTERVAL,
     *F15.9)
      PRINT 2003,IAVST
 2003 FORMAT(/1X,37HNUMBER OF INTERVALS BETWEEN PRINTOUTS,I5)
 2222 FORMAT(1X,10F11.7)
      PRINT 2004
 2004 FORMAT(//1X,33HTHE SYSTEM IS DX/DT=AX+BU+E WHERE)
      PRINT 2005
 2005 FORMAT(/1X,2HA=)
      DO 5 I=1,N
    5 PRINT 2222,(A(I,J),J=1,N)
      PRINT 2006
 2006 FORMAT (/1X,2HB=)
      DO 7 I=1,N
    7 PRINT 2222,(B(I,J),J=1,NB)
      PRINT 2007
 2007 FORMAT (//1X,29HTHE COST FUNCTION IS GIVEN BY)
      PRINT 2008
 2008 FORMAT (/1X,3HQ0=)
      DO 9 I=1,N
    9 PRINT 2222,(Q0(I,J),J=1,N)
      PRINT 2009
 2009 FORMAT (/1X,3HQ1=)
      DO 11 I=1,N
   11 PRINT 2222,(Q1(I,J),J=1,N)
      PRINT 2010
 2010 FORMAT (/1X,3HQ2=)
      DO 13 I=1,NB
   13 PRINT 2222,(Q2(I,J),J=1,NB)
      PRINT 2011,(X(I),I=1,N)
```

```
 2011 FORMAT (//1X,5HX(0)=/1X,10F12.7)
      DO 101 I=1,NB
      DO 101 J=1,NB
  101 RIC(I,J)=Q2(I,J)
      CALL SYMIN(NB,10,IFAIL,RIC)
      IF(IFAIL-1)103,102,103
  102 PRINT 2100
 2100 FORMAT (/1X,26HINVERSION OF Q2 HAS FAILED)
  103 CONTINUE
      DO 110 I=1,NB
      DO 110 J=1,N
      CON(I,J)=0
      DO 110 K=1,NB
  110 CON(I,J)=CON(I,J)+RIC(I,K)*B(J,K)
      DO 115 I=1,N
      DO 115 J=1,N
      RIC(I,J)=0
      DO 115 K=1,NB
  115 RIC(I,J)=RIC(I,J)+B(I,K)*CON(K,J)
C
C
C CALCULATION OF R,S AND LFI
C
      READ 1000,IEC
      PRINT 3160,IEC
      DO 130 I=1,N
      DO 130 J=1,N
  130 R(I,J,L+1)=Q0(I,J)
      DO 135 I=1,N
  135 S(I,L+1)=0
      LFI(L+1)=0
      T=L*TDIST
      DO 200 J=L,1 ,-1
      DO 140 I=1,N
      SI (I)=S(I,J+1)
      DO 140 K=1,N
  140 RIN(I,K)=R(I,K,J+1)
      FIN=LFI(J+1)
      H=-TDIST
      CALL BERRSF(T,RIN,RUT,SI ,SUT,FIN,FUT,H,N,10)
      DO 144 I=1,N
      S(I,J)=SUT(I)
      DO 144 K=1,N
  144 R(I,K,J)= RUT(I,K)
      LFI(J)=FUT
      T=T-TDIST
  200 CONTINUE
      SK=LFI(1)
      DO 220 I=1,N
      SK=SK+X(I)*S(I,1)
      DO 220 K=1,N
  220 SK=SK+X(I)*R(I,K,1)*X(K)
      PRINT 2900,SK
 2900 FORMAT (//5X,29H                  XT*R*X+XT*S+LFI=,F12.7)
C
C COMPUTATION OF MATRICES OF THE EQUIVALENT DISCRETE SYSTEM
C
      DO 250 I=1,N
      DO 250 J=1,N
  250 A(I,J)=A(I,J)*TDIST
      DO 253 I=1,N
      DO 253 J=1,N
  253 RIN(I,J)=A(I,J)
      DO 255 I=1,N
```

```
      DO 255 J=1,NB
  255 B(I,J)=B(I,J)*TDIST
      CALL COSA(A,B,N,NB,10)
      DO 260 I=1,N
      DO 260 J=1,N
  260 DELTA(I,J)=0
      DO 261 I=1,N
  261 DELTA(I,I)=TDIST
      CALL COSA(RIN,DELTA,N,N,10)
      PRINT 4555
 4555 FORMAT (/5X,2HFI)
      DO 263 I=1,N
  263 PRINT 2222,(A(I,J),J=1,N)
      PRINT 4556
 4556 FORMAT (/5X,5HGAMMA)
       DO 264 I=1,N
  264 PRINT 2222,(B(I,J),J=1,NB)
      PRINT 4557
 4557 FORMAT (/5X,5HDELTA)
      DO 265 I=1,N
  265 PRINT 2222,(DELTA(I,J),J=1,N)
C
C RECONSTRUCTION OF TRAJECTORY
C
      READ 1000,IEC
      PRINT 3160,IEC
 3160 FORMAT(/5X,4HIEC=,I5)
      SIMP=0
       ISIMP=1
      T=0
      LL=L+1
      KK=IAVST-1
      J=0
  270 J=J+1
      KK=KK+1
      DO 275 I=1,NB
      U(I)=0
      DO 275 K=1,N
      W=0
      DO 272 M=1,N
  272 W=W+R(K,M,J)*X(M)
      W=W+0.5*S(K,J)
  275 U(I)=U(I)-CON(I,K)*W
      T=(J-1)*TDIST
      CALL EBRKN(N,T)
      IF(KK-IAVST) 277,276,277
  276 PRINT 3000,T
 3000 FORMAT (//1X,7HTIME T=,F12.7)
      PRINT 3001,(X(I),I=1,N)
 3001 FORMAT (/10X,2HX=,/1X,10F12.7)
      PRINT 3002, (U(I),I=1,NB)
 3002 FORMAT (/10X,2HU=,/1X,10F12.7)
      PRINT 4000,(E(I),I=1,N)
 4000 FORMAT (/10X,2HE=,/1X,10F12.7)
      KK=0
  277 W=0
      DO 280 I=1,N
      DO 280 K=1,N
  280 W=W+X(I)*Q1(I,K)*X(K)
      DO 282 I=1,NB
      DO 282 K=1,NB
  282 W=W+U(I)*Q2(I,K)*U(K)
```

```
      IF(J.EQ.1.OR.J.EQ.LL) GO TO 295
      GO TO (290,292),ISIMP
  290 W=4*W
      ISIMP= 2
      GO TO 295
  292 W=2*W
      ISIMP=1
  295 SIMP=SIMP+W
      IF(J.EQ.LL) GO TO 350
      DO 305 I=1,N
      W=0
      DO 302 K=1,N
  302 W=W+A(I,K)*X(K)+DELTA(I,K)*E(K)
      DO 303 K=1,NB
  303 W=W+B(I,K)*U(K)
  305 SUT(I)=W
      DO 310 I=1,N
  310 X(I)=SUT(I)
      GO TO 270
  350 SIMP=TDIST*SIMP/3
      W=0
      DO 360 I=1,N
      DO 360 K=1,N
  360 W=W+X(I)*Q0(I,K)*X(K)
      SIMP=SIMP+W
      PRINT 3100, SIMP
 3100 FORMAT (//1X,30HCOST COMPUTED FROM TRAJECTORY=,F15.7)
      KK=IAVST-1
      DO 410 J=1,LL
      KK=KK+1
      IF(IAVST-KK)410,390,410
  390 W=(J-1)*TDIST
      PRINT 3200,W
 3200 FORMAT (//5X,7HTIME T=,F10.7)
      PRINT 3203
 3203 FORMAT (/5X,2HR=)
      DO 391 I=1,N
  391 PRINT 3201,(R(I,K,J),K=1,N)
 3201 FORMAT (/1X,10F12.7)
      PRINT 3202,(S(I,J),I=1,N)
 3202 FORMAT (/5X,2HS=,/1X,10F12.7)
      PRINT 3205, LFI(J)
 3205 FORMAT (/5X,4HLFI=,F12.7)
      KK=0
  410 CONTINUE
      STOP
      END
```

```
      SUBROUTINE BERRSF(T0,RIN,RUT,SI,SUT,FIN,FUT,H,N,IA)
      DIMENSION W(10,10),Z(10,10),WW(10),ZZ(10)
      DIMENSION RIN(IA,IA) , RUT(IA,IA),SI (IA),SUT(IA)
      DIMENSION A(5)
      COMMON /FUN /W,Z,WW,ZZ,WWW,ZZZ,TE
      T=T0
      A(1)=H/2
      A(2)=H/2
      A(5)=H/2
      A(3)=H
      A(4)=H
      TE=T
      DO 25 I=1,N
      DO 25 J=1,N
      RUT(I,J)=RIN(I,J)
   25 W(I,J)=RIN(I,J)
      DO 30 I=1,N
      SUT(I)=SI (I)
   30 WW(I)=SI (I)
      FUT=FIN
      WWW=FIN
      DO 150 K=1,4
      CALL FCN (N)
      TE=T+A(K)
      DO 50 I=1,N
      DO 50 J=1,N
      W(I,J)=RIN(I,J)+A(K)*Z(I,J)
   50 RUT(I,J)=RUT(I,J)+A(K+1)*Z(I,J)/3
      DO 60 I=1,N
      WW(I)=SI (I)+A(K)*ZZ(I)
   60 SUT(I)=SUT(I)+A(K+1)*ZZ(I)/3

      WWW=FIN+A(K)*ZZZ
  150 FUT=FUT+A(K+1)*ZZZ/3
      RETURN
      END
```

A: 11

```
      SUBROUTINE FCN (N)
      DIMENSION R(10,10),DRDT(10,10),S(10),DSDT(10)
      DIMENSION V(10,10)
      DIMENSION A(10,10),RIC(10,10),E(10),Q1(10,10)
      COMMON /EB/E /MAIN/A,Q1,RIC /FUN /R,DRDT,S,DSDT,FI,DFIDT,TE
      CALL EBRKN(N,TE)
      DO 15 I=1,N
      DO 15 K=1,N
      V(I,K)=0
      DO 15 M=1,N
   15 V(I,K)=V(I,K)+R(I,M)*RIC(M,K)
      DO 25 I=1,N
      DO 25 K=1,N
      DRDT(I,K)=0
      DO 20 M=1,N
   20 DRDT(I,K)=DRDT(I,K)+V(I,M)*R(M,K)-R(I,M)*A(M,K)-A(M,I)*R(M,K)
   25   DRDT(I,K)=DRDT(I,K)-Q1(I,K)
      DO 30 I=1,N
      DSDT(I)=0
      DO 30 M=1,N
   30 DSDT(I)=DSDT(I)+V(I,M)*S(M)-A(M,I)*S(M)-2*R(I,M)*E(M)
      DFIDT=0
      DO 35 I=1,N
      DO 35 K=1,N
   35 DFIDT=DFIDT+S(I)*RIC(I,K)*S(K)/4
      DO 40 I=1,N
   40 DFIDT=DFIDT-S(I)*E(I)
      RETURN
      END
```

```
      SUBROUTINE EBRKN(N,T)
      DIMENSION E(10),F(10),G(10)
      COMMON /EB/ E /EE/ IEC
      GO TO (50,100,200,250),IEC
   50 CONTINUE
      E(1)=0
      E(2)=0
      E(3)=10-10*(T-1)*(T-1)
      GO TO 300
  100 CONTINUE
      E(1)=2
      E(2)=0.5
      GO TO 300
  200 CONTINUE
      E(1)=1.0
      E(2)=0
      GO TO 300
  250 CONTINUE
      E(1)=T
      E(2)=0
  300 CONTINUE
      RETURN
      END
```

A: 23

Exempel på användning av diagonaldekomponering

N= 2 L= 10 EPS= .00010

FI

| | |
|---|---|
| .70000 | .20000 |
| .30000 | .75000 |

GAMMA

| | |
|---|---|
| 1.00000 | 2.00000 |

Q0

| | |
|---|---|
| 2.00000 | 2.00000 |

Q1

| | |
|---|---|
| 2.00000 | 2.00000 |

Q2

| | |
|---|---|
| 1.00000 | 1.00000 |

| J | X0 | |
|---|---|---|
| 1 | 10.0000000 | 5.0000000 |
| 2 | .0000000 | .0000000 |
| 3 | .0000000 | .0000000 |
| 4 | .0000000 | .0000000 |
| 5 | .0000000 | .0000000 |
| 6 | .0000000 | .0000000 |
| 7 | .0000000 | .0000000 |
| 8 | .0000000 | .0000000 |
| 9 | .0000000 | .0000000 |
| 10 | .0000000 | .0000000 |

1:a iterationen

| J | X | |
|---|---|---|
| 1 | 10.0000000 | 5.0000000 |
| 2 | 2.3927420 | .7102789 |
| 3 | .6430120 | .7738777 |
| 4 | .2893997 | .2539779 |
| 5 | .1113857 | .1068638 |
| 6 | .0446930 | .0418494 |
| 7 | .0177271 | .0167107 |
| 8 | .0070544 | .0066369 |
| 9 | .0028118 | .0026403 |
| 10 | .0011841 | .0010636 |

KOSTNAD= 307.03249

4:e iterationen

| J | X | |
|---|---|---|
| 1 | 10.0000000 | 5.0000000 |
| 2 | 2.3726122 | .6591033 |
| 3 | .5324150 | .1153498 |
| 4 | .1167412 | .0229829 |
| 5 | .0229606 | .0033473 |
| 6 | -.0006993 | -.0035930 |
| 7 | -.0052725 | -.0084510 |
| 8 | -.0063412 | -.0076219 |
| 9 | -.0051130 | -.0061088 |
| 10 | -.0035784 | -.0041260 |

KOSTNAD= 305.67407

J slutgiltig lösn. X

| J | | |
|---|---|---|
| 1 | 10.0000000 | 5.0000000 |
| 2 | 2.3726571 | .6591088 |
| 3 | .5326398 | .1154327 |
| 4 | .1177025 | .0233868 |
| 5 | .0258879 | .0050033 |
| 6 | .0056859 | .0010897 |
| 7 | .0012487 | .0002391 |
| 8 | .0002752 | .0000539 |
| 9 | .0000631 | .0000153 |
| 10 | .0000194 | .0000099 |

KOSTNAD= 305.67335

optimal bana

| | | |
|---|---|---|
| 1 | 10.0000000 | 5.0000000 |
| 2 | 2.3517090 | .5534918 |
| 3 | .5181592 | .0838500 |
| 4 | .1120313 | .0156394 |
| 5 | .0240827 | .0031961 |
| 6 | .0051676 | .0006747 |
| 7 | .0011082 | .0001440 |
| 8 | .0002377 | .0000309 |
| 9 | .0000513 | .0000068 |
| 10 | .0000124 | .0000023 |

KOSTNAD= 305.64290

kap.3 ex.1

DIM. X= 2 DIM. U= 2 INTERVALL= 11

FI=

| | |
|---|---|
| 2.0000000 | 1.0000000 |
| 1.0000000 | 1.0000000 |

GAMMA=

| | |
|---|---|
| 1.0000000 | 4.0000000 |
| 2.0000000 | 3.0000000 |

Q0=

| | |
|---|---|
| 1.0000000 | .0000000 |
| .0000000 | 1.0000000 |

Q1=

| | |
|---|---|
| 1.0000000 | .0000000 |
| .0000000 | 1.0000000 |

Q2=

| | |
|---|---|
| 1.0000000 | .0000000 |
| .0000000 | 1.0000000 |

XSTART=

| | |
|---|---|
| .00000 | .00000 |

E

| | | |
|---|---|---|
| 1 | .00000 | .00000 |
| 2 | .30902 | .30902 |
| 3 | .58779 | .58779 |
| 4 | .80902 | .80902 |
| 5 | .95106 | .95106 |
| 6 | 1.00000 | 1.00000 |
| 7 | .95106 | .95106 |
| 8 | .80902 | .80902 |
| 9 | .58779 | .58779 |
| 10 | .30902 | .30902 |
| 11 | -.00000 | -.00000 |

kap.3 ex1 forts.

| J | U | |
|---|---|---|
| 1 | -.0051403 | -.0029173 |
| 2 | -.0527917 | -.0600151 |
| 3 | -.0992344 | -.1141861 |
| 4 | -.1364431 | -.1572198 |
| 5 | -.1603753 | -.1848356 |
| 6 | -.1686244 | -.1943501 |
| 7 | -.1603705 | -.1848385 |
| 8 | -.1364192 | -.1572331 |
| 9 | -.0991143 | -.1142365 |
| 10 | -.0521075 | -.0600577 |

| J | X | |
|---|---|---|
| 1 | .0000000 | .0000000 |
| 2 | -.0168096 | -.0190326 |
| 3 | -.0364869 | -.0124538 |
| 4 | -.0536209 | -.0021823 |
| 5 | -.0657292 | .0086683 |
| 6 | -.0714513 | .0187382 |
| 7 | -.0701894 | .0269877 |
| 8 | -.0620591 | .0325983 |
| 9 | -.0478545 | .0350186 |
| 10 | -.0289656 | .0340111 |
| 11 | -.0072414 | .0296744 |

KOSTNAD= .3630241

KOSTNAD ENL. XT*R*X+XT*S+FI= .3630245

A:17

kap.3 ex.2

DIM. X= 3 DIM. U= 2 INTERVALL= 11

FI=

| | | |
|---|---|---|
| 1.0000000 | .0000000 | .0000000 |
| .0000000 | .0000000 | 1.0000000 |
| .0000000 | 1.0000000 | .0000000 |

GAMMA=

| | |
|---|---|
| 1.0000000 | .0000000 |
| 1.0000000 | .0000000 |
| 0.0 | 1.0 |

Q0=

| | | |
|---|---|---|
| 1.0000000 | .0000000 | .0000000 |
| .0000000 | 1.0000000 | .0000000 |
| .0000000 | .0000000 | 1.0000000 |

Q1=

| | | |
|---|---|---|
| 1.0000000 | .0000000 | .0000000 |
| .0000000 | 1.0000000 | .0000000 |
| .0000000 | .0000000 | 1.0000000 |

Q2=

| | |
|---|---|
| 1.0000000 | 1.0000000 |
| 1.0000000 | 1.0000000 |

XSTART=

| | | |
|---|---|---|
| 1.00000 | 1.00000 | .00000 |

E

| | | | |
|---|---|---|---|
| 1 | 1.00000 | 1.00000 | 1.00000 |
| 2 | 1.00000 | 1.00000 | 1.00000 |
| 3 | 1.00000 | 1.00000 | 1.00000 |
| 4 | 1.00000 | 1.00000 | 1.00000 |
| 5 | 1.00000 | 1.00000 | 1.00000 |
| 6 | 1.00000 | 1.00000 | 1.00000 |
| 7 | 1.00000 | 1.00000 | 1.00000 |
| 8 | 1.00000 | 1.00000 | 1.00000 |
| 9 | 1.00000 | 1.00000 | 1.00000 |
| 10 | 1.00000 | 1.00000 | 1.00000 |
| 11 | 1.00000 | 1.00000 | 1.00000 |

kap.3 ex.2 forts.

| J | U | |
|---|---|---|
| 1 | -1.2991851 | -1.3852224 |
| 2 | -1.6024995 | -.5737044 |
| 3 | -1.0903860 | -.9669660 |
| 4 | -1.0136854 | -.9986837 |
| 5 | -1.0047538 | -.9756597 |
| 6 | -1.0074206 | -.9336207 |
| 7 | -1.0140010 | -.8618992 |
| 8 | -1.0215357 | -.7341739 |
| 9 | -1.0038361 | -.5203056 |
| 10 | -.7654700 | -.2910927 |

| J | X | | |
|---|---|---|---|
| 1 | 1.0000000 | 1.0000000 | .0000000 |
| 2 | .7008149 | -.2991851 | .6147776 |
| 3 | .0983154 | .0122781 | .1271105 |
| 4 | .0079294 | .0367245 | .0453121 |
| 5 | -.0057560 | .0316267 | .0380408 |
| 6 | -.0105097 | .0332871 | .0559670 |
| 7 | -.0179304 | .0485464 | .0996664 |
| 8 | -.0319313 | .0856654 | .1866471 |
| 9 | -.0534671 | .1651114 | .3514916 |
| 10 | -.0573032 | .3476554 | .6448059 |
| 11 | .1772268 | .8793359 | 1.0565627 |

KOSTNAD= 43.6199899

KOSTNAD ENL. XT*R*X+XT*S+FI= 43.6199899

A:19

Kont. system ex.2

| | | | |
|---|---|---|---|
| TIME T= .0000000 | | korrekta värden | |
| X= 1.0000000 .0000000 | | | |
| U= -1.8000042 | | -1.8 | |
| TIME T= .5000000 | | | |
| X= .9371671 -.6753012 | | o.9375 | -0.675 |
| U= -.8995531 | | -0.9 | |
| E= .5000000 .0000000 | | | |
| TIME T= 1.0000000 | | | |
| X= .8992021 -.9006041 | | 0.9 | -0.9 |
| U= .0000000 | | 0.0 | |
| E= 1.0000000 .0000000 | | | |
| COST COMPUTED FROM TRAJECTORY= | .6741486 | 0.675 | |
| XT*R*X+XT*S+LFI= | .6750123 | | |

Kont. system ex.2 forts.

TIME T= .0000000

korrekta värden

R=

| | | | |
|---|---|---|---|
| .3000077 | .3000013 | 0.3 | 0.3 |
| .3000013 | .2999981 | 0.3 | 0.3 |

S=

| | | | |
|---|---|---|---|
| .3000039 | .2999996 | 0.3 | 0.3 |

LFI= .0750006

TIME T= .5000000

R=

| | | | |
|---|---|---|---|
| .4615422 | .2307690 | 0.4615385 | 0.2307692 |
| .2307690 | .1153841 | 0.2307692 | 0.1153846 |

S=

| | | | |
|---|---|---|---|
| .3461536 | .1730764 | 0.3461538 | 0.1730769 |

LFI= .0649037

A 21

Kont. system ex. 3

korrekta värden

TIME T= .0000000

X=
.0000000 .5000000

U=
-.9999963 -1.0

TIME T= .5000000

X=
.9999986 .4999992 1.0 0.5

U=
-.9999991 -1.0

E=
2.0000000 .5000000

TIME T= 1.0000000

X=
1.9999932 .9999948 2.0 1.0

U=
-.9999948 -1.0

E=
2.0000000 .5000000

COST COMPUTED FROM TRAJECTORY= 3.6666402

XT*R*X+XT*S+LFI= 3.6666012

3.6666667

A:22

Kont. system ex. 3 forts.

| | | | korrekta värden | |
|---|---|---|---|---|
| TIME T= | .5000000 | | | |
| R= | | | | |
| .8333257 | .1666662 | | 0.8333333 | 0.1666667 |
| .1666662 | .3333359 | | 0.1666667 | 0.3333333 |
| S= | | | | |
| 1.4166521 | .3333319 | | 1.4166667 | 0.3333333 |
| LFI= | .6666596 | | | |

| | | | | |
|---|---|---|---|---|
| TIME T= | .0000000 | | | |
| R= | | | | |
| 1.2499801 | .2499989 | | 1.25 | 0.25 |
| .2499989 | .2500042 | | 0.25 | 0.25 |
| S= | | | | |
| 3.7499331 | .7499921 | | 3.75 | 0.75 |
| LFI= | 3.2291041 | | | |

A:23

Värmestaven. Optimal bana för e(t)=10

GLAD

INPUT DATA

N= 3 NB= 1

NUMBER OF INTERVALS= 1000 LENGTH OF INTERVAL .001

NUMBER OF INTERVALS BETWEEN PRINTOUTS 50

THE SYSTEM IS DX/DT=AX+BU+E WHERE

A=

| | | |
|---|---|---|
| -2.0000000 | 1.0000000 | .0000000 |
| 1.0000000 | -2.0000000 | 1.0000000 |
| .0000000 | 1.0000000 | -2.0000000 |

B=

1.0000000
.0000000
.0000000

THE COST FUNCTION IS GIVEN BY

Q0=

| | | |
|---|---|---|
| .0000000 | .0000000 | .0000000 |
| .0000000 | 10.0000000 | .0000000 |
| .0000000 | .0000000 | .0000000 |

Q1=

| | | |
|---|---|---|
| .0000000 | .0000000 | .0000000 |
| .0000000 | 10.0000000 | .0000000 |
| .0000000 | .0000000 | .0000000 |

Q2=

1.0000000

X(0)=

.0000000 .0000000 .0000000

IEC= 1

XT*R*X+XT*S+LFI= 25.6610897

IEC= 1

TIME T= .0000000

X=

.0000000 .0000000 .0000000

U=

-3.1884666

E=

.0000000 .0000000 10.0000000

TIME T= .0500000

X=

GLAD

| | | |
|---|---|---|
| -.1525684 | .0079545 | .4759444 |

U=
-3.2288296

E=

| | | |
|---|---|---|
| .0000000 | .0000000 | 10.0000000 |

TIME T= .1000000

X=

| | | |
|---|---|---|
| -.2916133 | .0297683 | .9073219 |

U=
-3.2605051

E=

| | | |
|---|---|---|
| .0000000 | .0000000 | 10.0000000 |

TIME T= .1500000

X=

| | | |
|---|---|---|
| -.4173799 | .0627591 | 1.2989663 |

U=
-3.2819857

E=

| | | |
|---|---|---|
| .0000000 | .0000000 | 10.0000000 |

TIME T= .2000000

X=

| | | |
|---|---|---|
| -.5301352 | .1047030 | 1.6551351 |

U=
-3.2917983

E=

| | | |
|---|---|---|
| .0000000 | .0000000 | 10.0000000 |

TIME T= .2500000

X=

| | | |
|---|---|---|
| -.6301469 | .1537655 | 1.9795847 |

U=
-3.2884583

E=

| | | |
|---|---|---|
| .0000000 | .0000000 | 10.0000000 |

TIME T= .3000000

X=

| | | |
|---|---|---|
| -.7176646 | .2084420 | 2.2756339 |

U=
-3.2704237

E=
.0000000 .0000000 10.0000000

TIME T= .3500000

X=
-.7929031 .2675085 2.5462225

U=
-3.2360484

E=
.0000000 .0000000 10.0000000

TIME T= .4000000

X=
-.8560272 .3299802 2.7939575

U=
-3.1835350

E=
.0000000 .0000000 10.0000000

TIME T= .4500000

X=
-.9071378 .3950764 3.0211564

U=
-3.1108844

E=
.0000000 .0000000 10.0000000

TIME T= .5000000

X=
-.9462581 .4621919 3.2298827

U=
-3.0158399

E=
.0000000 .0000000 10.0000000

TIME T= .5500000

X=
-.9733194 .5308727 3.4219790

U=
-2.8958260

E=
.0000000 .0000000 10.0000000

GLAD

TIME T= .6000000

X=
-.9881463 .6007966 3.5990940

U=
-2.7478789

E=
.0000000 .0000000 10.0000000

TIME T= .6500000

X=
-.9904409 .6717579 3.7627071

U=
-2.5685692

E=
.0000000 .0000000 10.0000000

TIME T= .7000000

X=
-.9797651 .7436544 3.9141497

U=
-2.3539087

E=
.0000000 .0000000 10.0000000

TIME T= .7500000

X=
-.9555196 .8164786 4.0546228

U=
-2.0992500

E=
.0000000 .0000000 10.0000000

TIME T= .8000000

X=
-.9169215 .8903107 4.1852153

U=
-1.7991638

E=
.0000000 .0000000 10.0000000

TIME T= .8500000

X=

```
  -.8629770     .9653152   4.3069212

       U=
 -1.4472995

       E=
   .0000000     .0000000  10.0000000

TIME T=     .9000000

       X=
  -.7924495   1.0417389   4.4206471

       U=
 -1.0362175

       E=
   .0000000     .0000000  10.0000000

TIME T=     .9500000

       X=
  -.7038231   1.1199121   4.5272278

       U=
  -.5571960

       E=
   .0000000     .0000000  10.0000000

TIME T=    1.0000000

       X=
  -.5952583   1.2002518   4.6274362

       U=
   .0000000

       E=
   .0000000     .0000000  10.0000000

COST COMPUTED FROM TRAJECTORY=     25.6562586
```

A:29

Värmestaven. Optimal bana för $e(t)=10-10(t-1)^2$

A:30 D

GLAD

INPUT DATA

N= 3 NB= 1

NUMBER OF INTERVALS= 100 LENGTH OF INTERVAL .0010

NUMBER OF INTERVALS BETWEEN PRINTOUTS 50

THE SYSTEM IS DX/DT=AX+BU+E WHERE

```
A=
 -2.0000000  1.0000000   .0000000
  1.0000000 -2.0000000  1.0000000
   .0000000  1.0000000 -2.0000000

B=
  1.0000000
   .0000000
   .0000000
```

THE COST FUNCTION IS GIVEN BY

```
Q0=
   .0000000   .0000000   .0000000
   .0000000 10.0000000   .0000000
   .0000000   .0000000   .0000000

Q1=
   .0000000   .0000000   .0000000
   .0000000 10.0000000   .0000000
   .0000000   .0000000   .0000000

Q2=
  1.0000000

X(0)=
    .0000000     .0000000     .0000000

    IEC=    2

                    XT*R*X+XT*S+LFI=   9.0694363

    IEC=    2

TIME T=     .0000000

          X=
    .0000000     .0000000     .0000000

          U=
  -1.7767875

          E=
    .0000000     .0000000     .0000000

TIME T=     .0500000

          X=
```

GLAD

-.0853191 -.0017113 .0232868

U=
-1.8078572

E=
.0000000 .0000000 .9750000

TIME T= .1000000

X=
-.1640566 -.0049860 .0894342

U=
-1.8358164

E=
.0000000 .0000000 1.9000000

TIME T= .1500000

X=
-.2366852 -.0074946 .1919686

U=
-1.8597352

E=
.0000000 .0000000 2.7750000

TIME T= .2000000

X=
-.3034868 -.0074374 .3251302

U=
-1.8785218

E=
.0000000 .0000000 3.5999999

TIME T= .2500000

X=
-.3645862 -.0034544 .4837818

U=
-1.8909337

E=
.0000000 .0000000 4.3750000

TIME T= .3000000

X=
-.4199788 .0054518 .6633308

U=
-1.8955997

```
        E=
   .0000000      .0000000    5.1000000

TIME T=     .3500000

        X=
  -.4695517      .0199794     .8596617

        U=
 -1.8910225

        E=
   .0000000      .0000000    5.7750000

TIME T=     .4000000

        X=
  -.5131015      .0405825    1.0690778

        U=
 -1.8755798

        E=
   .0000000      .0000000    6.4000000

TIME T=     .4500000

        X=
  -.5503475      .0675168    1.2882506

        U=
 -1.8475173

        E=
   .0000000      .0000000    6.9749999

TIME T=     .5000000

        X=
  -.5809406      .1008784    1.5141771

        U=
 -1.8049371

        E=
   .0000000      .0000000    7.5000000

TIME T=     .5500000

        X=
  -.6044694      .1406365    1.7441410

        U=
 -1.7457786

        E=
   .0000000      .0000000    7.9750000
```

TIME T= .6000000

X=
-.6204632 .1866612 1.9756806

U=
-1.6677934

E=
.0000000 .0000000 8.4000000

TIME T= .6500000

X=
-.6283922 .2387473 2.2065593

U=
-1.5685126

E=
.0000000 .0000000 8.7750000

TIME T= .7000000

X=
-.6276648 .2966344 2.4347442

U=
-1.4452063

E=
.0000000 .0000000 9.1000000

TIME T= .7500000

X=
-.6176217 .3600244 2.6583815

U=
-1.2948326

E=
.0000000 .0000000 9.3750000

TIME T= .8000000

X=
-.5975283 .4285963 2.8757792

U=
-1.1139780

E=
.0000000 .0000000 9.6000000

TIME T= .8500000

X=

GLAD

```
-.5665624     .5020201    3.0853905

      U=
-.8987848

      E=
 .0000000     .0000000    9.7750001

TIME T=     .9000000

      X=
-.5238001     .5799677    3.2858002

      U=
-.6448645

      E=
 .0000000     .0000000    9.9000001

TIME T=     .9500000

      X=
-.4681974     .6621254    3.4757123

      U=
-.3471954

      E=
 .0000000     .0000000    9.9750000

TIME T=    1.0000000

      X=
-.3985677     .7482027    3.6539402

      U=
 .0000000

      E=
 .0000000     .0000000   10.0000000

COST COMPUTED FROM TRAJECTORY=      9.0502737
```

Värmestaven. Bana då $e(t)=10-10(1-t)^2$, men styrningen sker enligt den återkopplingslag som beräknats för $e(t)=10$

GLAD

INPUT DATA

N= 3 NB= 1

NUMBER OF INTERVALS= 1000 LENGTH OF INTERVAL .001

NUMBER OF INTERVALS BETWEEN PRINTOUTS 50

THE SYSTEM IS DX/DT=AX+BU+E WHERE

A=

| | | |
|---|---|---|
| -2.0000000 | 1.0000000 | .0000000 |
| 1.0000000 | -2.0000000 | 1.0000000 |
| .0000000 | 1.0000000 | -2.0000000 |

B=

1.0000000
.0000000
.0000000

THE COST FUNCTION IS GIVEN BY

Q0=

| | | |
|---|---|---|
| .0000000 | .0000000 | .0000000 |
| .0000000 | 10.0000000 | .0000000 |
| .0000000 | .0000000 | .0000000 |

Q1=

| | | |
|---|---|---|
| .0000000 | .0000000 | .0000000 |
| .0000000 | 10.0000000 | .0000000 |
| .0000000 | .0000000 | .0000000 |

Q2=

1.0000000

X(0)=

.0000000 .0000000 .0000000

IEC= 1

XT*R*X+XT*S+LFI= 25.6610897

IEC= 2

TIME T= .0000000

X=

.0000000 .0000000 .0000000

U=

-3.1884666

E=

.0000000 .0000000 .0000000

TIME T= .0500000

X=

GLAD



-.1472325 -.0032769 .0232606

U=
-2.9999115

E=
.0000000 .0000000 .9750000

TIME T= .1000000

X=
-.2721241 -.0105265 .0892474

U=
-2.8270178

E=
.0000000 .0000000 1.9000000

TIME T= .1500000

X=
-.3776274 -.0185101 .1914080

U=
-2.6686431

E=
.0000000 .0000000 2.7750000

TIME T= .2000000

X=
-.4662191 -.0247185 .3239491

U=
-2.5234584

E=
.0000000 .0000000 3.5999999

TIME T= .2500000

X=
-.5399685 -.0272488 .4817335

U=
-2.3899660

E=
.0000000 .0000000 4.3750000

TIME T= .3000000

X=
-.6005948 -.0246991 .6601904

U=
-2.2664999

```
E=
 .0000000    .0000000   5.1000000

TIME T=    .3500000

X=
-.6495138   -.0160807    .8552407

U=
-2.1512148

E=
 .0000000    .0000000   5.7750000

TIME T=    .4000000

X=
-.6878755   -.0007431   1.0632319

U=
-2.0420584

E=
 .0000000    .0000000   6.4000000

TIME T=    .4500000

X=
-.7165934    .0216884   1.2808827

U=
-1.9367336

E=
 .0000000    .0000000   6.9749999

TIME T=    .5000000

X=
-.7363649    .0513670   1.5052374

U=
-1.8326506

E=
 .0000000    .0000000   7.5000000

TIME T=    .5500000

X=
-.7476874    .0882692   1.7336238

U=
-1.7268680

E=
 .0000000    .0000000   7.9750000
```

TIME T= .6000000

X=
-.7508661 .1322338 1.9636196

U=
-1.6160384

E=
.0000000 .0000000 8.4000000

TIME T= .6500000

X=
-.7460176 .1829950 2.1930216

U=
-1.4963588

E=
.0000000 .0000000 8.7750000

TIME T= .7000000

X=
-.7330695 .2402115 2.4198231

U=
-1.3635464

E=
.0000000 .0000000 9.1000000

TIME T= .7500000

X=
-.7117591 .3034914 2.6421909

U=
-1.2128399

E=
.0000000 .0000000 9.3750000

TIME T= .8000000

X=
-.6816307 .3724135 2.8584462

U=
-1.0390488

E=
.0000000 .0000000 9.6000000

TIME T= .8500000

X=

GLAD

```
 -.6420370     .4465468    3.0670497

       U=
 -.8366470

       E=
  .0000000     .0000000    9.7750001

TIME T=     .9000000

       X=
 -.5921446     .5254666    3.2665879

       U=
 -.5999006

       E=
  .0000000     .0000000    9.9000001

TIME T=     .9500000

       X=
 -.5309433     .6087677    3.4557622

       U=
 -.3229846

       E=
  .0000000     .0000000    9.9750000

TIME T=    1.0000000

       X=
 -.4572596     .6960769    3.6333795

       U=
  .0000000

       E=
  .0000000     .0000000   10.0000000

COST COMPUTED FROM TRAJECTORY=       9.3419032
```

A:41

Värmestaven. Optimal bana för e(t)=10(1-t)

GLAD

INPUT DATA

N= 3 NB= 1

NUMBER OF INTERVALS= 1000 LENGTH OF INTERVAL .001

NUMBER OF INTERVALS BETWEEN PRINTOUTS 50

THE SYSTEM IS DX/DT=AX+BU+E WHERE

A=
```
 -2.0000000  1.0000000   .0000000
  1.0000000 -2.0000000  1.0000000
   .0000000  1.0000000 -2.0000000
```

B=
```
  1.0000000
   .0000000
   .0000000
```

THE COST FUNCTION IS GIVEN BY

Q0=
```
   .0000000   .0000000   .0000000
   .0000000 10.0000000   .0000000
   .0000000   .0000000   .0000000
```

Q1=
```
   .0000000   .0000000   .0000000
   .0000000 10.0000000   .0000000
   .0000000   .0000000   .0000000
```

Q2=
```
  1.0000000
```

X(0)=
```
   .0000000   .0000000   .0000000
```

IEC= 2

XT*R*X+XT*S+LFI= 9.0249070

IEC= 2

TIME T= .0000000

X=
```
   .0000000   .0000000   .0000000
```

U=
```
 -1.9663766
```

E=
```
   .0000000   .0000000 10.0000000
```

TIME T= .0500000

X=

-.0938601 .0092009 .4641102

U=
-1.9843179

E=
.0000000 .0000000 9.5000000

TIME T= .1000000

X=
-.1786601 .0337082 .8610892

U=
-1.9955042

E=
.0000000 .0000000 9.0000000

TIME T= .1500000

X=
-.2542914 .0694789 1.1979529

U=
-1.9990405

E=
.0000000 .0000000 8.5000000

TIME T= .2000000

X=
-.3207920 .1131689 1.4808747

U=
-1.9941907

E=
.0000000 .0000000 8.0000001

TIME T= .2500000

X=
-.3783084 .1620235 1.7152960

U=
-1.9803242

E=
.0000000 .0000000 7.5000000

TIME T= .3000000

X=
-.4270635 .2137846 1.9060214

U=
-1.9568664

E=
.0000000 .0000000 7.0000000

TIME T= .3500000

X=
-.4673291 .2666116 2.0573012

U=
-1.9232479

E=
.0000000 .0000000 6.5000001

TIME T= .4000000

X=
-.4994018 .3190160 2.1729023

U=
-1.8788636

E=
.0000000 .0000000 6.0000001

TIME T= .4500000

X=
-.5235832 .3698054 2.2561704

U=
-1.8230228

E=
.0000000 .0000000 5.5000001

TIME T= .5000000

X=
-.5401609 .4180369 2.3100818

U=
-1.7549010

E=
.0000000 .0000000 5.0000000

TIME T= .5500000

X=
-.5493912 .4629777 2.3372902

U=
-1.6734897

E=
.0000000 .0000000 4.5000000

TIME T= .6000000

X=
-.5514831 .5040732 2.3401659

U=
-1.5775407

E=
.0000000 .0000000 4.0000001

TIME T= .6500000

X=
-.5465819 .5409197 2.3208314

U=
-1.4655051

E=
.0000000 .0000000 3.5000001

TIME T= .7000000

X=
-.5347531 .5732424 2.2811908

U=
-1.3354605

E=
.0000000 .0000000 3.0000001

TIME T= .7500000

X=
-.5159645 .6008776 2.2229567

U=
-1.1850364

E=
.0000000 .0000000 2.5000000

TIME T= .8000000

X=
-.4900669 .6237591 2.1476737

U=
-1.0113212

E=
.0000000 .0000000 2.0000000

TIME T= .8500000

X=

GLAD

```
-.4567732     .6419069    2.0567381

       U=
-.8107578

       E=
 .0000000     .0000000    1.5000001

TIME T=     .9000000

       X=
-.4156335     .6554199    1.9514168

       U=
-.5790211

       E=
 .0000000     .0000000    1.0000001

TIME T=     .9500000

       X=
-.3660077     .6644710    1.8328623

       U=
-.3108742

       E=
 .0000000     .0000000     .5000000

TIME T=    1.0000000

       X=
-.3070324     .6693047    1.7021266

       U=
 .0000000

       E=
 .0000000     .0000000     .0000000

COST COMPUTED FROM TRAJECTORY=       9.0381856
```

A:47

Värmestaven. Bana då $e(t)=10(1-t)$, men styrning sker enligt den återkopplingslag som beräknats för $e(t)=10$

```
                      GLAD

INPUT DATA

N=     3                          NB=     1

NUMBER OF INTERVALS= 1000                               LENGTH OF INTERVAL      .001

NUMBER OF INTERVALS BETWEEN PRINTOUTS   50


THE SYSTEM IS DX/DT=AX+BU+E WHERE

A=
 -2.0000000  1.0000000   .0000000
  1.0000000 -2.0000000  1.0000000
   .0000000  1.0000000 -2.0000000

B=
  1.0000000
   .0000000
   .0000000


THE COST FUNCTION IS GIVEN BY

Q0=
   .0000000   .0000000   .0000000
   .0000000 10.0000000   .0000000
   .0000000   .0000000   .0000000

Q1=
   .0000000   .0000000   .0000000
   .0000000 10.0000000   .0000000
   .0000000   .0000000   .0000000

Q2=
  1.0000000


X(0)=
     .0000000     .0000000     .0000000

     IEC=     1

                    XT*R*X+XT*S+LFI=  25.6610897

     IEC=     2


TIME T=     .0000000

          X=
     .0000000     .0000000     .0000000

          U=
   -3.1884666

          E=
     .0000000     .0000000   10.0000000


TIME T=     .0500000

          X=
```

-.1524784 .0077632 .4640865

U=
-3.2229168

E=
.0000000 .0000000 9.5000000

TIME T= .1000000

X=
-.2909054 .0282970 .8609123

U=
-3.2365849

E=
.0000000 .0000000 9.0000000

TIME T= .1500000

X=
-.4150756 .0580309 1.1973967

U=
-3.2280194

E=
.0000000 .0000000 8.5000000

TIME T= .2000000

X=
-.5248924 .0940498 1.4796469

U=
-3.1959645

E=
.0000000 .0000000 8.0000001

TIME T= .2500000

X=
-.6203383 .1339921 1.7130639

U=
-3.1393678

E=
.0000000 .0000000 7.5000000

TIME T= .3000000

X=
-.7014536 .1759618 1.9024341

U=
-3.0574119

E=
.0000000 .0000000 7.0000000

TIME T= .3500000

X=
-.7683229 .2184551 2.0520073

U=
-2.9495694

E=
.0000000 .0000000 6.5000001

TIME T= .4000000

X=
-.8210696 .2602967 2.1655655

U=
-2.8156908

E=
.0000000 .0000000 6.0000001

TIME T= .4500000

X=
-.8598585 .3005863 2.2464833

U=
-2.6561179

E=
.0000000 .0000000 5.5000001

TIME T= .5000000

X=
-.8849066 .3386518 2.2977752

U=
-2.4718216

E=
.0000000 .0000000 5.0000000

TIME T= .5500000

X=
-.8965013 .3740090 2.3221430

U=
-2.2645559

E=
.0000000 .0000000 4.5000000

```
TIME T=     .6000000

        X=
  -.8950272     .4063254    2.3220110

        U=
 -2.0370049

        E=
   .0000000     .0000000    4.0000001

TIME T=     .6500000

        X=
  -.8809969     .4353884    2.2995605

        U=
 -1.7928901

        E=
   .0000000     .0000000    3.5000001

TIME T=     .7000000

        X=
  -.8550842     .4610759    2.2567574

        U=
 -1.5369690

        E=
   .0000000     .0000000    3.0000001

TIME T=     .7500000

        X=
  -.8181504     .4833295    2.1953762

        U=
 -1.2748369

        E=
   .0000000     .0000000    2.5000000

TIME T=     .8000000

        X=
  -.7712529     .5021312    2.1170209

        U=
 -1.0123825

        E=
   .0000000     .0000000    2.0000000

TIME T=     .8500000

        X=
```

GLAD

```
 -.7156157     .5174824    2.0231425

      U=
 -.7546996

      E=
  .0000000     .0000000    1.5000001

TIME T=     .9000000

      X=
 -.6525341     .5293896    1.9150552

      U=
 -.5041734

      E=
  .0000000     .0000000    1.0000001

TIME T=     .9500000

      X=
 -.5831704     .5378588    1.7939465

      U=
 -.2573483

      E=
  .0000000     .0000000     .5000000

TIME T=   1.0000000

      X=
 -.5081797     .5429039    1.6608923

      U=
  .0000000

      E=
  .0000000     .0000000     .0000000

COST COMPUTED FROM TRAJECTORY=      9.9270443
```