

DEKOMPOSITION AV SYSTEM AV HÖG ORDNING

BÖRJE AULIN

EXAMENSARBETE

Utfört höstterminen 1968 vid inst för
reglerteknik vid Lunds Tekniska Högskola
av Börje Aulin.

DEKOMPOSITION AV SYSTEM AV HÖG ORDNING.

SAMMANFATTNING.

En algoritm har utvecklats för dekomponering av ett lineärt dynamiskt system i subsystem, vilka definierar ett flernivå-system (multilevel system). Teorin för dekomponeringen grundar sig på grafteori. Grafen konstrueras från koefficient-matrisen.

För att kunna göra dekomponeringen på ett systematiskt sätt, införes en matris, definierad av grafen, som kallas reachability matris.

Två olika sätt att genomföra dekomponeringen demonstreras. Den ena metoden skall användas vid handräkning; den andra användes i ett datamaskin program. Flödesdiagram och FORTRAN program presenteras.

Därefter studeras några numeriska exempel.

En kort diskussion om lokal optimering av subsystemen i flernivå-system utföres. Två system simuleras, ett med normal koppling mellan subsystemen, det andra med stark koppling.

En anmärkning angående stabilitet för flernivå-system avslutar rapporten.

DECOMPOSITION OF LARGE SCALE SYSTEMS.

ABSTRACT.

An algorithm has been developed for decomposing a linear dynamic system into subsystems, building up a multilevel control system.

The theory for the decomposition is based upon graph theory.

The graph is constructed from the coefficient matrix.

In order to make the decomposition in a systematic way, a matrix, called the reachability matrix, which is based upon the graph, is introduced.

Two different ways of performing the decomposition are demonstrated.

The first method should be used when calculating for hand; the other is used in a computer program. Flow diagrams and the FORTRAN program are presented.

Then some numerical examples are studied.

A short discussion of local optimization of the subsystems in multilevel systems is made. Two systems are simulated, one with "normal" interaction between the subsystems, the other with "strong" interaction.

A notation about stability for multilevel systems finishes the report.

Innehållsförteckning.

Sid

Kap 1.	Dekomponering, teori.	1 - 22
Kap 2.	Flödesscheman och beskrivningar av subrutiner.	22 - 26
Kap 3.	Numeriska exempel. (Värmestaven, observerbara och kontrollerbara kanoniska formerna).	27 - 30
Kap 4.	Marvikenreaktorn.	31 - 40
Kap 5.	Optimering.	41 - 44
Kap 6.	En iakttagelse av stabilitet för flernivå-system.	45 - 47
	Referenser	48
Appendix 1.	Program och subrutiner för dekomponering.	
Appendix 2.	Subrutinen MARVIK.	
Appendix 3.	Olof Norinder, Atomenergi, Rapport R4-441.	
Appendix 4.	Sato, "On a decomposition technique for multilevel systems".	

KAP 1. DEKOMPONERING. TEORI.

Ett datamaskinprogram har konstruerats för dekomponering av ett lineärt dynamiskt system

$$\dot{x} = Ax + u \quad (1)$$

Om A är reducerbar kan (1) reduceras till k subsystem

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ & A_{22} & \cdots & A_{2k} \\ & & \ddots & \vdots \\ & & & A_{kk} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix} \quad (2)$$

där varje kvadratisk submatris A_{ii} ($1 \leq i \leq k$) är antingen irreducibel eller en 1 x 1 nollmatris.

Vi ser att (2) är ett flernivåsystem (multilevel system) med k st. nivåer. Ett flernivåsystem karakteriseras av existensen av en familj av regleringsenheter grupperade i en hierarki.

Av två enheter på olika nivåer kallas enheten på den högre nivån den suprema enheten (supremal unit), medan den på den lägre nivån kallas den infimala enheten (infimal unit). Effekten av den suprema på den infimala kallas intervention.

Vi kan schematiskt upprita strukturen för (2). Låt $k = 6$.

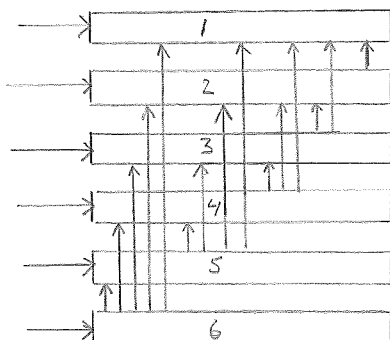


fig 1.

Matriserna A_{ij} ($i \neq j$) representerar en koppling från subsystem j till subsystem i . Är någon lika med en nollmatrix utgår motsvarande pil i fig 1.

Dekomponeringen göres med en grafisk metod.

Från matrisen A bildas en ny matris R enligt följande

$$\begin{cases} \text{Om } a_{ij} = 0 & \Rightarrow & r_{ij} = 0 \\ \text{Om } a_{ij} \neq 0 & \Rightarrow & r_{ij} = 1 \end{cases}$$

Vi sätter således alla positioner i R , som motsvarar fyllda positioner i A , lika med ett och övriga positioner lika med noll. I programmet sättes $r_{ij} = 0$, om $a_{ij} < \xi$. Här krävs en viss försiktighet vid valet av ξ . Det är lämpligast, att för hand med kännedom av det fysikaliska systemet avgöra vilka a_{ij} som kan sättas lika med noll.

Från matrisen R kan en riktad graf, som vi kallar tillståndsgraf, uppritas enligt följande recept:

$$\left\{ \begin{array}{l} \text{Om } r_{ij} = 1 \text{ drages en branch riktad från vertex } i \text{ till vertex } j. \\ \text{Om } r_{ij} = 0 \text{ drages ingen branch mellan vertex } i \text{ och vertex } j. \\ \text{Om } r_{ii} = 1 \text{ drages en branch från vertex } i \text{ till sig självt.} \\ \text{Om } r_{ii} = 0 \text{ drages ingen branch från vertex } i \text{ till sig självt.} \end{array} \right.$$

Vi observerar, att brancherna i tillståndsgrafan får motsatt riktning jämfört med brancherna i ett signalflödesdiagram. Anledningen till den valda metoden för konstruktionen av tillståndsgrafan är att dekomponeringen blir enklare att utföra .

Låt oss ta ett enkelt exempel och därvid jämföra tillståndsgrafan med en analogimaskinkoppling, där vi underlåter att rita ut potentiometrar och teckenvändare.

$$A = \begin{bmatrix} -2 & 0 & 0 & 0 & 2 \\ 6 & -4 & 2 & 1 & -3 \\ 3 & 0 & -1 & 0 & 0 \\ 0 & -3 & 2 & 0 & 5 \\ -2 & 0 & 4 & 0 & 0 \end{bmatrix}$$

Vi får $R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}$

Systemet kan dekomponeras i två subsystem

$$\frac{d}{dt} \begin{bmatrix} x_2 \\ x_4 \\ x_1 \\ x_3 \\ x_5 \end{bmatrix} = \begin{bmatrix} -4 & 1 & 6 & 2 & -3 \\ -3 & 0 & 0 & 2 & 5 \\ 0 & -2 & 0 & 2 & 0 \\ 0 & 3 & -1 & 0 & 0 \\ 0 & -2 & 4 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \\ x_1 \\ x_3 \\ x_5 \end{bmatrix} + \begin{bmatrix} u_2 \\ u_4 \\ u_1 \\ u_3 \\ u_5 \end{bmatrix}$$

Vi markerar dessa subsystem i fig 2 och fig 3.

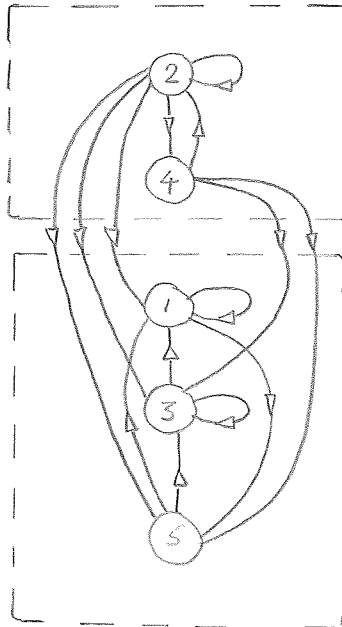


fig 2.

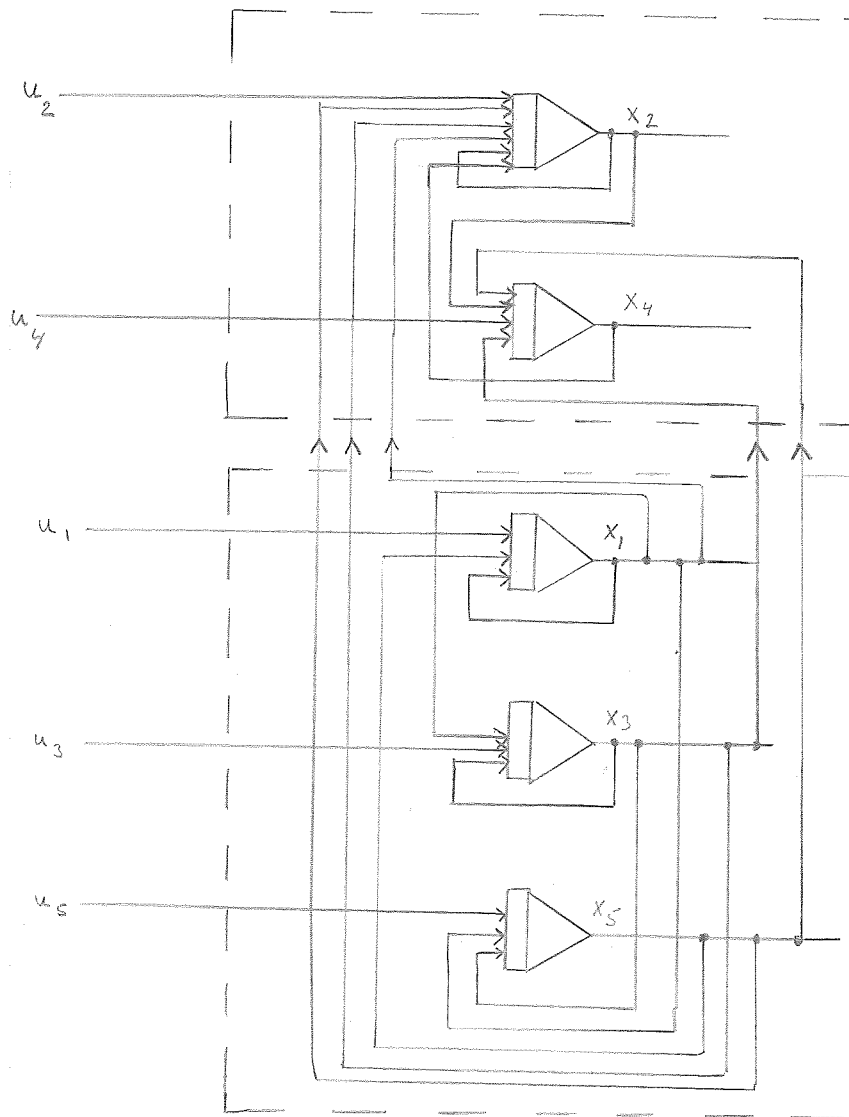


fig 3.

Vi ser att fig 3 har den i fig 1 skisserade strukturen.

Subsystemet med tillståndsvariablerna x_1, x_3, x_5 är det suprema systemet, medan subsystemet med tillståndsvariablerna x_2, x_4 är det infimala systemet.

Jämför vi nu med tillståndsgrafan i fig 2 finner vi, att brancherna som förbinder subsystemen har motsatt riktning .


Innan vi anger metoden för dekomponering, inför vi en del definitioner och bevisar några sätser, som gäller för riktade grafer.

Definition. En tillståndsgraf är starkt förbunden (strongly connected) då och endast då till varje vertexpar (I, J) finns en branchväg som börjar i vertex I och slutar i vertex J , (av godtycklig längd).


Definition. En delgraf till en tillståndsgraf är en subgraf sådan att man från ett godtyckligt vertex i denna ej kan nå ett vertex, som ej tillhör subgrafen.

Sats 1. Varje tillståndsgraf har en starkt förbunden delgraf.

Bevis: Är tillståndsgrafen starkt förbunden, har den sig själv som starkt förbunden delgraf. Är den ej starkt förbunden, finns ett vertexpar (I, J) sådant att vertex I ej når J . Låt T^x beteckna mängden av I och alla vertex som kan nås från I . Då gäller, att $I \in T^x$, $J \notin T^x$, dvs att T^x är en icke-tom, äkta delmängd till den givna vertexmängden T . T^x definierar en äkta delgraf G^x till tillståndsgrafens G . Är nu G^x starkt förbunden är sats 1 sann. Är åter G^x ej starkt förbunden innehåller den enligt ovanstående resonemang en äkta delgraf G^{xx} . Processen stannar alltså, om den leder till en starkt förbunden delgraf, och fortsätter annars. Eftersom i varje steg T^x är icke-tom kan processen leda ända ner till ett T^{xx} som omfattar endast ett vertex. Då kan två fall inträffa:

1) Motsvarande delgraf har följande utseende: 

som ju är starkt förbunden.

2) Den kan även se ut så här: , dvs ingen branch vilket motsvarar en 1×1 nollmatrix i den blocktriangulära matrisen A , vilket kommer att framgå senare.

För att sats 1 skall gälla även för fall 2), inför vi definitionsmässigt att en dylik graf är starkt förbunden. Härmed är sats 1 bevisad.

Sats 2. En tillståndsgraf är starkt förbunden då och endast då den ej har någon delgraf förutom sig själv.

Bevis: Om tillståndsgraf G ej har någon delgraf förutom sig själv måste den vara starkt förbunden. Ty om den ej är starkt förbunden, följer av beviset för sats 1 att det finns en äkta delgraf G^* till G .

Om å andra sidan G är starkt förbunden följer omedelbart att G ej har någon delgraf förutom sig själv. Härmed är sats 2 bevisad.

Redan nu kan man inse hur man från grafen skall kunna utföra dekomponeringen. Ur grafen bestämmer vi en starkt förbunden delgraf.

Subsystemet med motsvarande tillståndsvariabler kommer då att utgöra subsystemet på högsta nivån, nr k , eftersom detta ej skall påverkas av övriga subsystem. (Observera riktningen på brancherna! Om det gäller att inga brancher lämnar en vertexmängd gäller att motsvarande tillståndsvariabler ej påverkas av övriga tillståndsvariabler).

Därefter upprepas proceduren för den resterande reducerade grafen, ända tills denna blir starkt förbunden, då någon ytterligare delgraf ej kan finnas enligt sats 2.

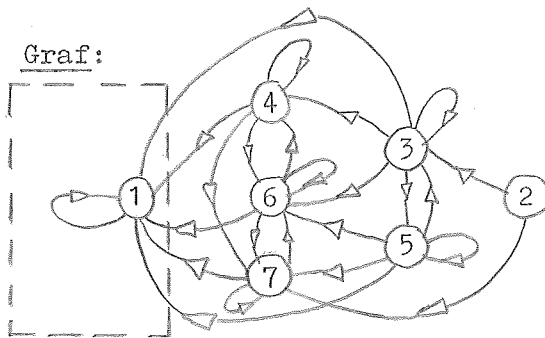
De i delgraferna ingående vertexmängderna placeras från slutet i en permuteringsvektor LD , med vilken sedan A 's block-triangulära form, AN , erhålles: $AN(I, J) = A(LD(I), LD(J))$.

Hur metoden kan göras systematiskt visas längre fram.

Först demonstrerar vi den med ett exempel.

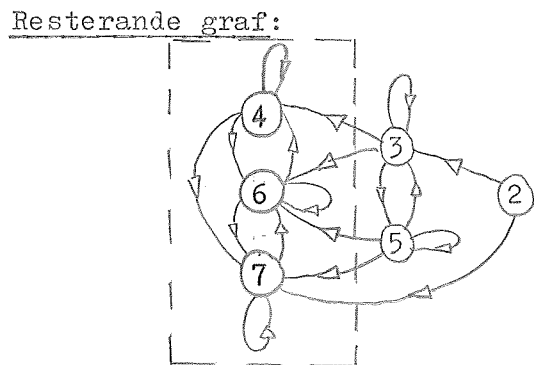
Låt $A = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 6 \\ 4 & 0 & -3 & 2 & 1 & 3 & 0 \\ 1 & 0 & 0 & 1 & 0 & 2 & 1 \\ 1 & 0 & 2 & 0 & 3 & 1 & 1 \\ 2 & 0 & 0 & 4 & 0 & 2 & 2 \\ 5 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$

$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$



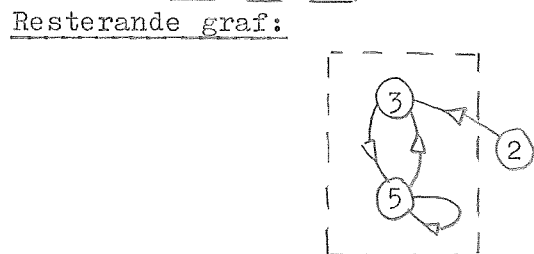
Vi ser, att $\{1\}$ är en starkt förbunden delgraf.

$LD = (0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1)$



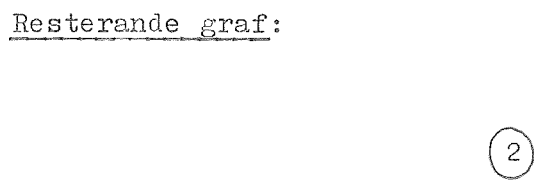
Vi ser, att $\{4,6,7\}$ är en starkt förbunden delgraf.

$LD = (0 \ 0 \ 0 \ 4 \ 6 \ 7 \ 1)$



Vi ser, att $\{3,5\}$ är en starkt förbunden delgraf.

$LD = (0 \ 3 \ 5 \ 4 \ 6 \ 7 \ 1)$



Vi ser, att $\{2\}$ är en starkt förbunden delgraf.

$LD = (2 \ 3 \ 5 \ 4 \ 6 \ 7 \ 1)$

②

Vi får då: $AN = \begin{matrix} & 2 & 3 & 5 & 4 & 6 & 7 & 1 \\ \begin{matrix} 2 \\ 3 \\ 5 \\ 4 \\ 6 \\ 7 \\ 1 \end{matrix} & \begin{bmatrix} 0 & 2 & 0 & 0 & 0 & 6 & 0 \\ 0 & -3 & 1 & 2 & 3 & 0 & 4 \\ 0 & 2 & 3 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 2 & 1 & 1 \\ 0 & 0 & 0 & 4 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \end{matrix}$

Vi fick alltså 4 st subsystem.

Metod 1.

Denna metod är lämplig för datamaskinberäkning.

Vi visar följande uttryck för REACH:

$$\text{REACH} = R \cup R^2 \cup R^3 \cup \dots \cup R^n$$

där n är ordningen på matrisen A.

Med \cup avses logisk addition positionsvis.

$$R^2(I,J) = \bigcup_{K=1}^n R(I,K) \cap R(K,J)$$

Med \cap avses logisk multiplikation.

Man inser lätt följande:

$R(I,J)$ indikerar förekomsten av en branchväg av längden 1 från I till J.

$R^2(I,J)$ " " " " " " " 2 " " " "

$R^3(I,J)$ " " " " " " " 3 " " " "

.....

$R^n(I,J)$ " " " " " " " n " " " "

Att summationen avbrytes efter R^n inser man så här.

Låt M^1 beteckna de vertex som I når med en branchväg av längden 1.

M^1 anges av R. (rad I)

Låt M^2 beteckna de vertex som I når med en branchväg av längden 2.

M^2 anges av R^2 . (rad I)

För att det ska vara lönt att fortsätta processen måste det ha kommit till element i M^2 som inte finns i M^1 . I värsta fall har det bara kommit till ett element.

Därefter beräknas M^3 .

För att det nu ska vara lönt att fortsätta måste det ha kommit till element i M^3 som inte finns i $M^1 \cup M^2$.

På samma sätt fortsätter processen, och eftersom det finns n vertex, kan processen endast hålla på i n steg.

I programmet avbrytes summationen så snart som $R^i = R^{i-1}$.

Vi visar metod 1 med ett enkelt exempel.

$$\text{Låt } R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\text{Vi får } R^2 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$\text{och } R^3 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} = R^2.$$

Således avbrytes summationen här, ty $R^4 = R \cdot R^3 = R \cdot R^2 = R^3 = R^2$.

$$\text{REACH} = R \cup R^2 \cup R^3 \cup R^4 \cup R^5 = R \cup R^2 =$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \cup \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Metod 2.

Denna metod är lämplig för räkning för hand.

Skriv upp R med tomrum i de positioner som är lika med noll.

Betrakta rad 1!

Denna analyseras från vänster till höger, och om $R(1,I) = 1$, så skall rad 1 ha ettor i de positioner som rad I har ettor, varför vi skriver ner dessa i rad 1, såvida det inte redan finns en etta. Detta följer omedelbart, ty vertex 1 når vertex I. Då når vertex 1 även alla de vertex, som vertex I når. (relationen "når" är transitiv).

Därefter går vi till nästa rad och upprepar proceduren ända tills vi har gått igenom rad n.

Om inga nya ettor tillkom, är den beräknade matrisen = REACH.

Om det å andra sidan tillkom nya ettor under analysen av de n raderna, måste vi upprepa analysen igen ända tills inga nya ettor tillkommer.

Metoden är mycket enkel att utföra för hand. Man behöver t.ex inte följa ovanstående algoritm strängt, utan kan hoppa hur som helst mellan raderna. Huvudsaken är emellertid att hela beräkningen avslutas med analys av samtliga rader uppifrån och ner för att vara säker på att inga ettor glömts.

Vi demonstrerar metoden med ett enkelt exempel.

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \text{Steg 1} \begin{bmatrix} 1 & & & & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & & 1 & & \\ & & 1 & 1 & 1 \\ 1 & & 1 & & \end{bmatrix} \cdot$$
$$\text{Steg 2} \begin{bmatrix} 1 & & 1 & & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & & 1 & & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & & 1 & & 1 \end{bmatrix} \cdot \text{Steg 3} \begin{bmatrix} 1 & & 1 & & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & & 1 & & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & & 1 & & 1 \end{bmatrix} \cdot$$

I steg 3 tillkom inga nya ettor, varför vi får

$$\text{REACH} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

I verkligheten arbetar vi givetvis hela tiden bara med en matris, och skriver alltså inte upp den efter varje steg.

När vi nu har lärt oss att beräkna reachability-matriser, visar vi följande viktiga egenskap hos dessa.

Sats 3. Låt rad I vara den rad i REACH som har det minsta antalet ettor. Antag, att raden har ettor i pos x_1, \dots, x_k ($k \geq 1$). Då är den subgraf, som omfattar vertexmängden $\{x_1, \dots, x_k\}$ en starkt förbunden delgraf.

Bevis: A) Låt $\text{REACH}(I, I) = 1$, dvs $I \in \{x_1, \dots, x_k\}$. Först visas att $\{x_1, \dots, x_k\}$ verkligen definierar en delgraf, dvs att man från ett godtyckligt vertex i delgrafan ej kan nå ett vertex, som ej tillhör denna. För matrisen REACH skall detta betyda, att i raderna x_1, \dots, x_k är samtliga element i kolonner som ej är lika med x_1, \dots, x_k lika med noll. Detta följer direkt, ty antag, att exempelvis rad $J \in \{x_1, \dots, x_k\}$ har en etta i kolonn $L \notin \{x_1, \dots, x_k\}$. Men då relationen "når" är transitiv, följer att vertex I når vertex J når vertex L dvs att vertex I når vertex L . Men detta innebär ju, att $\text{REACH}(I, L)$ skulle vara lika med ett. Motsägelse! Vi har därmed visat, att $\{x_1, \dots, x_k\}$ definierar en delgraf.

Återstår att visa att delgrafen är starkt förbunden. Även detta följer omedelbart, ty vore delgrafen ej starkt förbunden, är minst ett element i submatrisen av REACH svarande mot subgrafen som definieras av vertexmängden $\{x_1, \dots, x_k\}$ lika med noll.

(En reachabilitymatris är starkt förbunden då och endast då alla element är lika med ett, och en godtycklig submatris av en reachabilitymatris är också en reachabilitymatris).

Men detta innebär, att i någon rad $\in \{x_1, \dots, x_k\}$ finns det högst $k-1$ st ettor. Motsägelse!

Vi har därmed visat att delgrafen är starkt förbunden.

B) Låt $\text{REACH}(I, I) = 0$, dvs $I \notin \{x_1, \dots, x_k\}$.

Beviset utföres på precis samma sätt som i A.

Med sats 3 har vi alltså fått en mycket enkel metod att medelst reachability-matrisen upptäcka starkt förbundna delgrafer.

Vi är nu klara att ange dekomponerings-metoden. Det är emellertid även nu lämpligt att införa olika metoder för handräkning respektive datamaskinberäkning.

Metod 1. (Handräkning).

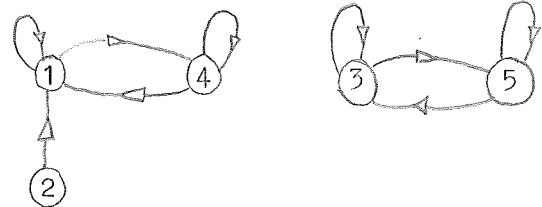
Summera elementen i varje rad I i REACH enligt följande recept:

$$\left\{ \begin{array}{ll} \text{REACH}(I,I) = 1 & \Rightarrow \text{SUM} = \text{SUM} + 1 \\ \text{REACH}(I,I) = 0 & \Rightarrow \text{SUM} = \text{SUM} + 0.5 \\ \text{REACH}(I,J) = 1 \quad (I \neq J) & \Rightarrow \text{SUM} = \text{SUM} + 1 \\ \text{REACH}(I,J) = 0 \quad (I \neq J) & \Rightarrow \text{SUM} = \text{SUM} + 0 \end{array} \right.$$

Därefter erhålles permutationsvektorn LD genom att man från slutet radnummer så att deras motsvarande SUM bildar en växande talföljd (ej strängt växande). Det sistnämnda är inte alltid sant, vilket vi skall visa med ett enkelt exempel. Det är nämligen ingen entydig algoritm om flera subgrafer har samma SUM. Vid handräkning kan man emellertid knappast göra fel. I metoden för datamaskinberäkning har vi måst tillgripa en något krångligare algoritm som dock alltid är entydig.

Exempel.

Betrakta följande tillståndsgraf



$$\text{Vi får REACH} = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ \left[\begin{array}{ccccc} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{array} \right] \end{array} \begin{array}{c} \text{SUM} \\ 2 \\ 2.5 \\ 2 \\ 2 \\ 2 \end{array}$$

Vi får LD = (2 1 4 3 5) (även (2 3 5 1 4) är korrekt).

$$\text{Permuteras REACH fås NREACH} = \begin{array}{c} 2 \\ 1 \\ 4 \\ 3 \\ 5 \end{array} \begin{array}{ccccc} 2 & 1 & 4 & 3 & 5 \\ \left[\begin{array}{ccccc} 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{array} \right] \end{array}$$

Däremot skulle en datamaskin kunna få fram LD = (2 1 3 4 5),

vilken ger felaktigt resultat.

Detta ser vi genast om vi skriver upp NREACH.

$$NREACH = \begin{matrix} & & & & & & 2 & 1 & 3 & 4 & 5 \\ & & & & & & 2 & & & & \\ & & & & & & 1 & & & & \\ NREACH = & & & & & & 3 & & & & \\ & & & & & & 4 & & & & \\ & & & & & & 5 & & & & \end{matrix} \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Anledningen till att den senare permutationsvektorn är felaktig, är givetvis att $\{1,3\}$ och $\{4,5\}$ ej definierar starkt förbundna subgrafer.

Ser man efter i reachability-matrisen samtidigt som man går igenom de olika SUM, kan man inte göra fel.

Vi tar ett litet större exempel, varvid vi utgår från matrisen R.

$$R = \begin{matrix} & & & & & & & & & & & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ & & & & & & & & & & & & 1 & & & & & & & & & & & \\ & & & & & & & & & & & & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & & \\ & & & & & & & & & & & & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & & & \\ & & & & & & & & & & & & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & & & \\ & & & & & & & & & & & & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & & & \\ & & & & & & & & & & & & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & & & \\ & & & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & & & \\ & & & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & & & \\ & & & & & & & & & & & & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & & & \\ & & & & & & & & & & & & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & & & \end{matrix} \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Vi beräknar REACH med metod 2.

$$REACH = \begin{matrix} & & & & & & & & & & & & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & SUM \\ & & & & & & & & & & & & 1 & & & & & & & & & & & \\ & & & & & & & & & & & & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & & 6 \\ & & & & & & & & & & & & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & & 6 \\ & & & & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & 10 \\ & & & & & & & & & & & & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & & 6 \\ & & & & & & & & & & & & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & & 6 \\ & & & & & & & & & & & & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & & 7 \\ & & & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & & 2 \\ & & & & & & & & & & & & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & & 2 \\ & & & & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & 10 \\ & & & & & & & & & & & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & & & 10 \end{matrix} \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Vi får alltså perm.vektorn $LD = (3 \ 9 \ 10 \ 6 \ 1 \ 2 \ 4 \ 5 \ 7 \ 8)$.

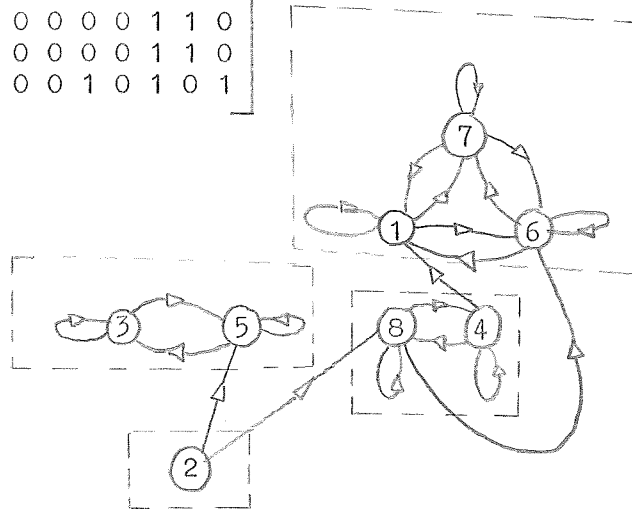
Som en extra kontroll beräknar vi den av LD permuterade REACH, NREACH.

$$\begin{array}{r}
 \begin{array}{c} 3 \\ 9 \\ 10 \\ 6 \\ 1 \\ 2 \\ 4 \\ 5 \\ 7 \\ 8 \end{array} \\
 \text{NREACH} =
 \end{array}
 \begin{array}{c}
 3 \ 9 \ 10 \ 6 \ 1 \ 2 \ 4 \ 5 \ 7 \ 8 \\
 \left[\begin{array}{cccccccccc}
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
 \end{array} \right]
 \end{array}$$

I detta exempel uppstod inga problem vid analysen av SUM.

Låt oss ta ännu ett exempel.

$$\begin{array}{r}
 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \\
 \text{Låt } R =
 \end{array}
 \begin{array}{c}
 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \\
 \left[\begin{array}{cccccccc}
 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1
 \end{array} \right]
 \end{array}$$



Tillståndsgraf:

$$\begin{array}{r}
 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} \\
 \text{REACH} =
 \end{array}
 \begin{array}{c}
 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \\
 \left[\begin{array}{cccccccc}
 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1
 \end{array} \right]
 \end{array}
 \begin{array}{c}
 \text{SUM} \\
 3 \\
 7.5 \\
 2 \\
 5 \\
 2 \\
 3 \\
 3 \\
 5
 \end{array}$$

Vi får $LD = (2\ 4\ 8\ 1\ 6\ 7\ 3\ 5)$

Permuterar vi REACH med LD får vi

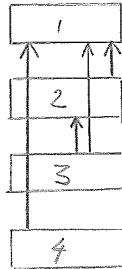
$$\text{NREACH} = \begin{matrix} & & & & 2 & 4 & 8 & 1 & 6 & 7 & 3 & 5 \\ \begin{matrix} 2 \\ 4 \\ 8 \\ 1 \\ 6 \\ 7 \\ 3 \\ 5 \end{matrix} & \left[\begin{array}{cccccccc} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ & 1 & 1 & 1 & 1 & 0 & 0 \\ & & 1 & 1 & 1 & 0 & 0 \\ & & & 1 & 1 & 1 & 0 & 0 \\ & & & & 1 & 1 & 1 & 0 & 0 \\ & & & & & 0 & & 1 & 1 \\ & & & & & & & & 1 & 1 \end{array} \right] \end{matrix}$$

Vi ser att subsystemen som omfattar tillståndsvariablerna

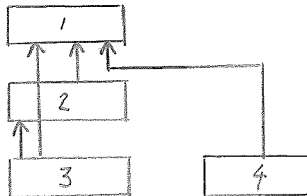
$\{x_1, x_6, x_7\}$ resp $\{x_3, x_5\}$ står på likvärdig nivå.

Orsaken härtill är att submatrisen $A_{34} = \emptyset$, vilket betyder att subsystem 4 inte intervenerar i subsystem 3.

Strukturen för systemet kan ritas så här:



eller bättre:



Metod 2. (Användes i datamaskinprogrammet). (Jfr även sid 6).

Här sker dekomponeringen enligt följande algoritm.

I reachability-matrisen letar vi upp den rad, som har minst antal ettor, säg ettor i positionerna (kolonnerna) x_1, \dots, x_k , $k \geq 1$.

Då är enligt sats 3 den subgraf som definieras av vertexmängden

$\{x_1, \dots, x_k\}$ en starkt förbunden delgraf. Därefter placeras siffrorna x_1, \dots, x_k längst bak i permuteringsvektorn LD.

Plocka därefter in övriga $x_i \notin \{x_1, \dots, x_k\}$, och permutera REACH.

(Är $k = 0$ skall motsvarande radnummer placeras sist i LD).

Därefter avskiljes de k nedersta raderna och de k högra kolonnerna,

(den nedersta raden och den högra kolonnen för $k = 0$), varvid

ordningen på REACH reduceras till $n - k$. ($n-1$ för $k = 0$).

Är den reducerade reachability-matrisen, vilken således är

reachability-matrisen för de vertex som $\in \{x_1, \dots, x_k\}$, starkt

förbunden är vi klara, annars reduceras den reducerade reachability-

matrisen på samma sätt. Processen stannar så snart vi får en

starkt förbunden reducerad reachability-matris eller en reducerad

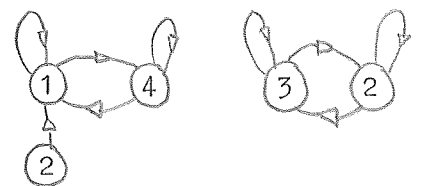
reachability-matris av ordningen 1, vilken enligt definition alltid är

starkt förbunden.

Vi demonstrerar metod 2 med ett exempel.

Tag den förut analyserade tillståndsgraf

$$\text{REACH} = \begin{matrix} & 1 & 2 & 3 & 4 & 5 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{matrix}$$



Rad 1 har minst antal ettor, vilka finns i pos 1 och 4.

LD = (2 3 5 1 4)

Permuteras REACH fås NREACH =

$$\begin{array}{l} 2 \\ 3 \\ 5 \\ 1 \\ 4 \end{array} \left[\begin{array}{cccc|cc} 2 & 3 & 5 & 1 & 4 & & \\ \hline 0 & 0 & 0 & 1 & 1 & & \\ 0 & 1 & 1 & 0 & 0 & & \\ 0 & 1 & 1 & 0 & 0 & & \\ 0 & 0 & 0 & 1 & 1 & & \\ 0 & 0 & 0 & 1 & 1 & & \end{array} \right]$$

I matrisen svarande mot vertex $\{2,3,5\}$ har rad 1 minst antal ettor, nämligen noll st.

$$LD = (3 \ 5 \ 2 \ 1 \ 4).$$

Permuteras matrisen svarande mot vertex $\{2,3,5\}$ fås

$$NREACH = \begin{array}{l} 3 \\ 5 \\ 2 \end{array} \left[\begin{array}{cc|ccc} 3 & 5 & 2 & & & \\ \hline 1 & 1 & 0 & 0 & 0 & \\ 1 & 1 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & \end{array} \right]$$

Den återstående submatrisen är som synes starkt förbunden, varför processen stannar, och vi har fått fram LD, med vilken vi kan permutera den ursprungliga koefficientmatrisen A.

Vi löser exemplet på sid 7 med metod 2.

$$R = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

$$\text{REACH} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

Rad 1 har minst antal ettor, en st. Således def $\{1\}$ en starkt förbunden delgraf. LD = (2 3 4 5 6 7 1)

$$\text{NREACH} = \begin{matrix} & \begin{matrix} 2 & 3 & 4 & 5 & 6 & 7 & 1 \end{matrix} \\ \begin{matrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 1 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

Raden med index 4 har minst antal ettor, tre st. således def $\{4,6,7\}$ en starkt förbunden delgraf. LD = (2 3 5 4 6 7 1)

$$\text{NREACH} = \begin{matrix} & \begin{matrix} 2 & 3 & 5 & 4 & 6 & 7 \end{matrix} \\ \begin{matrix} 2 \\ 3 \\ 5 \\ 4 \\ 6 \\ 7 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \end{matrix}$$

Raden med index 2 har minst antal ettor, två st, i pos 3 och 5. Således def $\{3,5\}$ en starkt förbunden delgraf. LD ändras ej.

$$\text{NREACH} = \begin{matrix} & \begin{matrix} 2 & 3 & 5 \end{matrix} \\ \begin{matrix} 2 \\ 3 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{matrix}$$

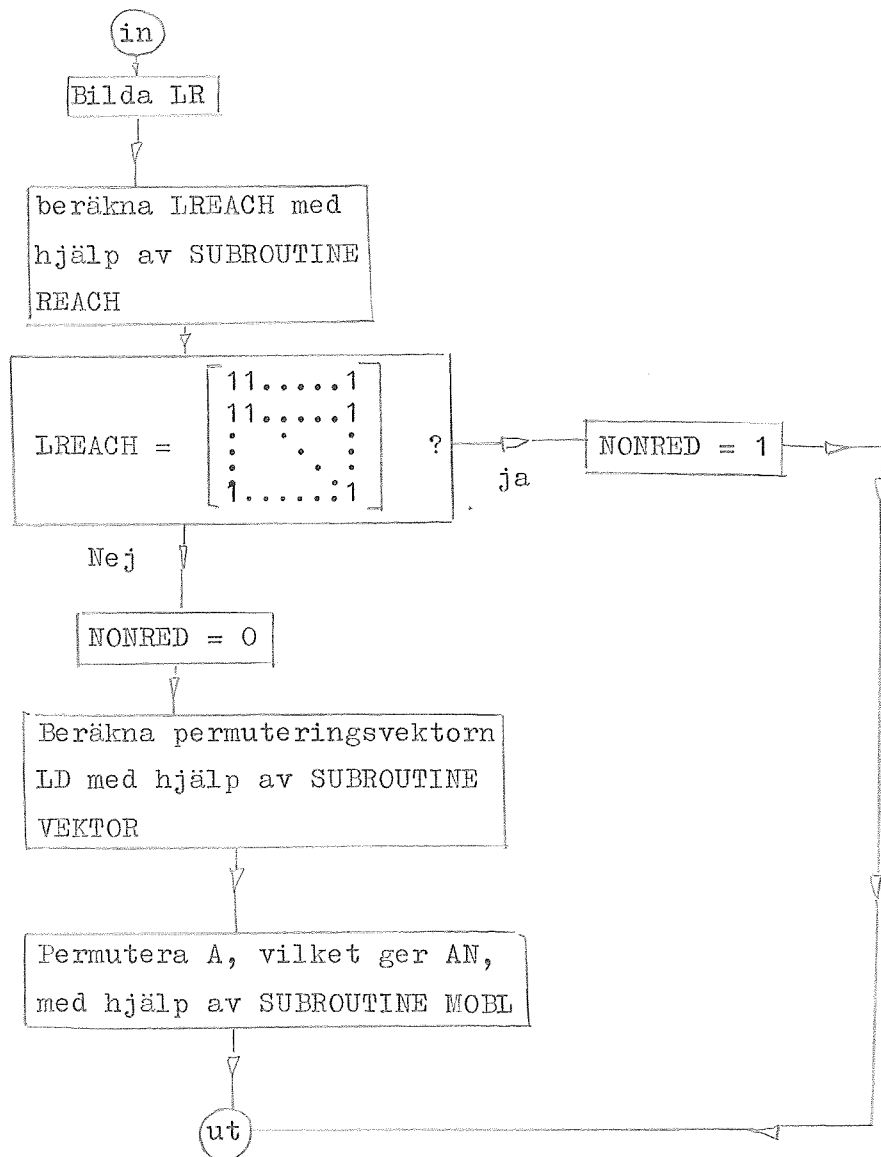
Nu är emellertid den återstående reachability-matrisen av ordning ett, varför processen stannar.

Vi har alltså fått fram LD = (2 3 5 4 6 7 1) dvs samma resultat som på sid 7.

Vi anger nu flödesscheman för de större subrutinerna.

KAP 2. FLÖDESSCHEMAN OCH BESKRIVNINGAR AV SUBROUTINER.

Flödesschema för huvudsubrutinen DEKOS.



Beteckningar: LR: matrisen R.

$$\begin{cases} LR(I,J) = 0, & \text{om } |A(I,J)| \leq EPS \\ LR(I,J) = 1, & \text{om } |A(I,J)| > EPS \end{cases}$$

LREACH: reachability-matrisen.

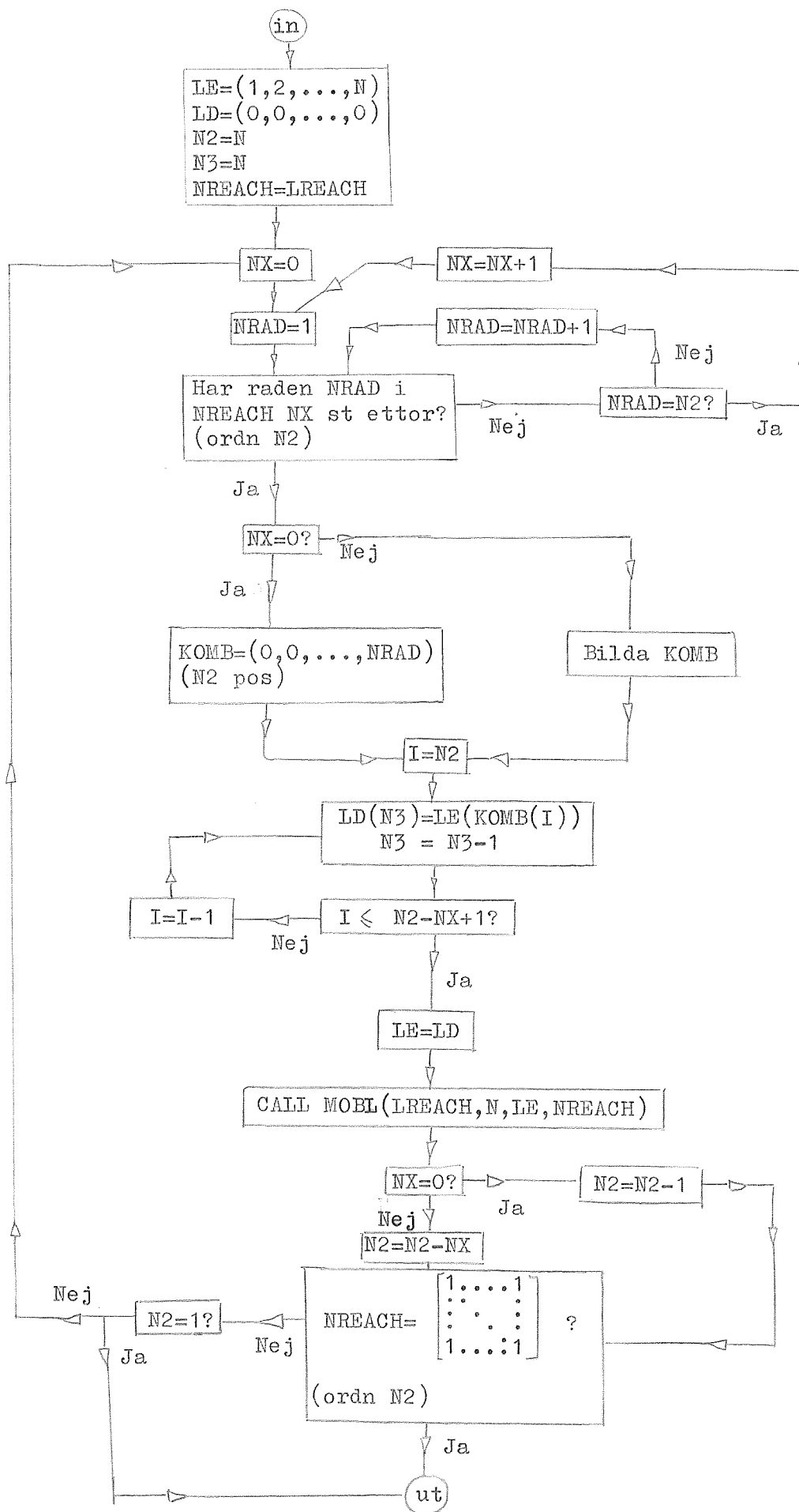
AN: A på block-triangulär form.

Anrop: CALL DEKOS(A,N,EPS,AN,LD, NONRED, LR, LREACH)

Input data: A,N,EPS.

Output data: AN,LD, NONRED, LR, LREACH.

Flödesschema för subrutinen VEKTOR.



Beteckningar: N: ordningen på matrisen A.

LE: en "hjälpvektor" till permuteringsvektorn LD.
Användas för att inte LD skall förstöras.

N2: ordningen på matrisen NREACH, vilken är reachability-matrisen för den resterande subgrafan enligt sid 18-19. N2 minskas med NX= antalet vertex i den senaste funna starkt förbundna delgrafan (jfr sid 18).

N3: anger den första position i vektorn LD, i vilken vertexen för delgrafan skall placeras.

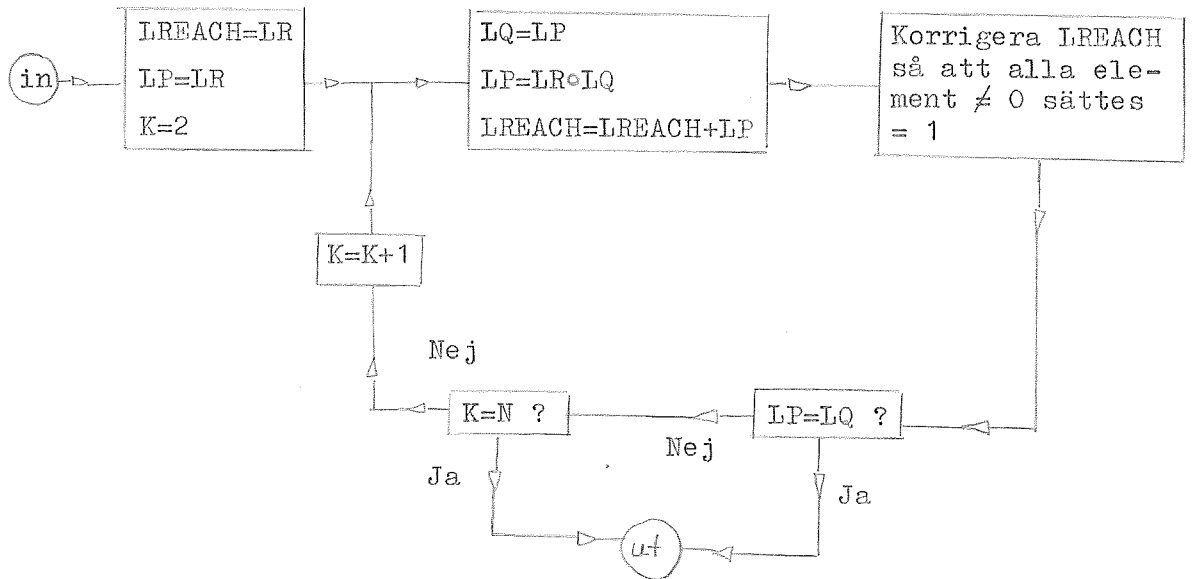
KOMB: om den aktuella raden i NREACH har ettor i positioner x_1, \dots, x_{NX} blir $KOMB = (0, 0, \dots, x_1, \dots, x_{NX})$. (N2 pos).

Subrutinen beräknar permuteringsvektorn LD utgående från reachability-matrisen LREACH.

Anrop: CALL VEKTOR(LREACH, N, LD).

Verkningsätt framgår av flödesschemat jämte sid 18-19.

Flödesschema för subrutinen REACH



$$\text{LREACH} = \sum_{i=1}^N U (\text{LR})^i$$

Som synes avbrytes den logiska summationen så snart $(\text{LR})^i = (\text{LR})^{i-1}$.

Vidare utföres logisk addition som vanlig addition av matriser, varefter alla element $\neq 0$ sättes = 1.

SUBROUTINE MATMPY(L1,N,L2,L3)

Beräknar den logiska matrismultiplikationen $L3 = L1 \circ L2$

$$\text{där } L3(I,J) = \bigcup_{K=1}^N L1(I,K) \wedge L2(K,J) .$$

Beräkningen utföres så att först beräknas den vanliga matrismultiplikationen $L1 \cdot L2$, varefter alla element $\neq 0$ sättes = 1.

SUBROUTINE MOBL(A,N,LD,AN)

Utgående från en eventuellt ofullständig permutationsvektor $LD = (0,0,\dots,0,ld_1,\dots,ld_{NX})$ göres denna fullständig genom att resterande vertex-index plockas in, varefter matrisen A permuteras med den erhållna LD, varvid AN erhålles $(AN(I,J)=A(LD(I),LD(J)))$.

Jfr även sid 18.

Detta inser man även, då man beaktar det fysikaliska systemet. Om det nämligen gått att genomföra en dekomponering, så skulle det finnas gitterpunkter vars temperaturer ej skulle påverkas av temperaturerna i övriga gitterpunkter, vilket är orimligt, eftersom värmeenergi kan passera i båda riktningarna.

Vi visar även, att koefficientmatrisen i den observerbara resp kontrollerbara kanoniska formen definierar starkt förbundna tillståndsgrafer.

1. Den observerbara kanoniska formen.

$$\frac{dz}{dt} = \begin{bmatrix} -a_1 & 1 & 0 & 0 & 0 \\ -a_2 & 0 & 1 & 0 & 0 \\ -a_3 & 0 & 0 & 1 & 0 \\ -a_4 & 0 & 0 & 0 & 1 \\ -a_5 & 0 & 0 & 0 & 0 \end{bmatrix} z + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} u \quad (m = 5)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \end{bmatrix} z + D u$$

Vi får: $R = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$

REACH beräknas med metod 1.

$$R^2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$R^3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$R^4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$R^5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\text{REACH} = \prod_{i=1}^5 U R^i = R^5 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Således är tillståndsgrafan starkt förbunden.

2. Den kontrollerbara kanoniska formen.

$$\frac{dz}{dt} = \begin{bmatrix} -a_1 & -a_2 & -a_3 & -a_4 & -a_5 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} z + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u$$

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Med metod 2 fås direkt $\text{REACH} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$

KAP 4. MARVIKENREAKTORN.

Se arbetsrapport. (Appendix 3).

Ur ekvationerna (24) - (51) beräknas matrisen A i $\dot{z} = A z$.

Ekv (24).

$$\begin{aligned} \frac{d}{dt} \delta\phi &= -\frac{1}{l} \cdot \beta \delta\phi + \frac{1}{l}(1-\beta) \cdot \phi \cdot \delta k + \sum_{i=1}^6 \lambda_i \delta C_i = \\ &= -\frac{\beta}{l} \delta\phi + \frac{1-\beta}{l} \phi (\alpha_f \cdot \delta P_f + \alpha_s \cdot \delta T_s + \alpha_m \cdot \delta T_m + \alpha_X \cdot \delta X + \\ &\quad + \frac{\partial F}{\partial P} \cdot \delta P + \frac{\partial F}{\partial \theta} \cdot \delta \theta + \frac{\partial F}{\partial p} \cdot \delta p + \delta R) + \sum_{i=1}^6 \lambda_i \delta C_i. \end{aligned}$$

Sätt $M = \frac{1-\beta}{l} \phi$.

$$\begin{aligned} \frac{d}{dt} \delta\phi &= -\frac{\beta}{l} \delta\phi + M \alpha_f \cdot \delta P_f + M \alpha_s \cdot \delta T_s + M \alpha_m \cdot \delta T_m + M \alpha_X \cdot \delta X + \\ &\quad + M \frac{\partial F}{\partial P} \cdot \delta P + M \frac{\partial F}{\partial \theta} (\delta T_s - \delta T'_u) + M \frac{\partial F}{\partial p} \delta p + M \delta R + \sum_{i=1}^6 \lambda_i \delta C_i = \\ &= -\frac{\beta}{l} \delta\phi + M \alpha_f \delta P_f + M \alpha_s \cdot \frac{\partial T_s}{\partial p} \delta p + M \alpha_m (0,5 (\delta T_{mi} + \delta T_{mu})) + \\ &\quad + M \alpha_X \delta X + M \frac{\partial F}{\partial P} \delta P + M \frac{\partial F}{\partial \theta} \frac{\partial T_s}{\partial p} \delta p - M \frac{\partial F}{\partial \theta} \delta T'_u + M \frac{\partial F}{\partial p} \delta p + \\ &\quad + M \delta R + \sum_{i=1}^6 \lambda_i \delta C_i = -\frac{\beta}{l} \delta\phi + \sum_{i=1}^6 \lambda_i \delta C_i + M \alpha_f \delta P_f + \\ &\quad + M \frac{\partial F}{\partial P} \delta P + M \alpha_X \delta X + 0,5 M \alpha_m q_i \delta T'_{mi} + 0,5 M \alpha_m q_s \delta T'_s + \\ &\quad + 0,5 M \alpha_m \delta T'_x - M \frac{\partial F}{\partial \theta} \delta T'_u + 0,5 M \alpha_m \frac{W}{W_m} \delta T_{cu} + \\ &\quad + M (\alpha_s \frac{\partial T_s}{\partial p} + \frac{\partial F}{\partial \theta} \frac{\partial T_s}{\partial p} + \frac{\partial F}{\partial p}) \delta p + M \cdot \delta R. \end{aligned}$$

Ekv (36).

$$\frac{d}{dt} \delta T'_s = \gamma_m^2 (-\delta T'_s + \delta T_s) = -\frac{2}{\gamma_m} \delta T'_s + \frac{2}{\gamma_m} \frac{\partial T_s}{\partial p} \delta p.$$

Ekv (38).

$$\frac{d}{dt} \delta T'_u = \gamma_{dc} \frac{1}{\gamma_{dc}} (-\delta T'_u + \delta T_u).$$

Ekv (42) ger

$$(\gamma_u - \gamma_s) \frac{\partial W_K}{\partial P} \delta P + (\gamma_u - \gamma_s) \frac{\partial W_K}{\partial \theta} (\frac{\partial T_s}{\partial p} \delta p - \delta T'_u) +$$

$$\begin{aligned}
 & + (T_u - T_s) \frac{\partial W_K}{\partial p} \delta p + (W_k + W_m) \delta T_u = \\
 & = W_K \frac{\partial T_s}{\partial p} \delta p + W_m (q_i \cdot \delta T'_{mi} + q_s \cdot \delta T'_s + \delta T'_\gamma) \\
 \text{Alltså: } \delta T_u & = - \frac{(T_u - T_s) \frac{\partial W_K}{\partial p}}{W_K + W_m} \delta p + \frac{W_m \cdot q_i}{W_K + W_m} \delta T'_{mi} + \frac{W_m \cdot q_s}{W_K + W_m} \delta T'_s + \\
 & + \frac{W_m}{W_K + W_m} \delta T'_\gamma + \frac{(T_u - T_s) \frac{\partial W_K}{\partial \theta}}{W_K + W_m} \delta T'_u + \\
 & + \frac{W_K \frac{\partial T_s}{\partial p} - (T_u - T_s) \frac{\partial W_K}{\partial \theta} \frac{\partial T_s}{\partial p} - (T_u - T_s) \frac{\partial W_K}{\partial p}}{W_K + W_m} \cdot \delta p.
 \end{aligned}$$

Ekv (38).

$$\begin{aligned}
 \frac{d}{dt} \delta T'_u & = \left(-\frac{1}{\tau_{dc}} + \mathcal{N} \cdot \frac{\partial W_K}{\partial \theta} \right) \delta T'_u - \mathcal{N} \cdot \frac{\partial W_K}{\partial p} \cdot \delta p + \mathcal{M} \cdot q_i \cdot \delta T'_{mi} + \\
 & + \mathcal{M} \cdot q_s \cdot \delta T'_s + \mathcal{M} \delta T'_\gamma + \left(W_K \frac{\partial T_s}{\partial p} \cdot \mathcal{L} - \mathcal{N} \left(\frac{\partial W_K}{\partial \theta} \frac{\partial T_s}{\partial p} + \frac{\partial W_K}{\partial p} \right) \right) \delta p.
 \end{aligned}$$

$$\begin{cases}
 \mathcal{L} = \frac{1}{\tau_{dc} (W_K + W_m)} \\
 \mathcal{M} = W_m \cdot \mathcal{L} \\
 \mathcal{N} = (T_u - T_s) \mathcal{L}
 \end{cases}$$

Ekv. (40). Sätt $\rho = \frac{A}{\tau_{DC} \frac{\partial M_D}{\partial p}}$

$$\begin{aligned}
 \frac{d}{dt} \delta S_{DC} & = - \left(\frac{1}{\tau_{DC}} + \rho \right) \delta S_{DC} + \rho (\delta S_K - \delta S_{UT}) = \\
 & = - \left(\frac{1}{\tau_{DC}} + \rho \right) \delta S_{DC} + \rho \frac{\partial S_K}{\partial p} \delta p - \rho \frac{\partial S_K}{\partial \theta} \delta T'_u + \\
 & + \rho \left(\frac{\partial S_K}{\partial \theta} \frac{\partial T_s}{\partial p} + \frac{\partial S_K}{\partial p} - \frac{\partial S_{UT}}{\partial p} \right) \cdot \delta p.
 \end{aligned}$$

Ekv (41).

$$\begin{aligned} \frac{d}{dt} \delta p = & \frac{\frac{\partial S_K}{\partial P}}{\frac{\partial M_D}{\partial p}} \delta P - \frac{\frac{\partial S_K}{\partial \theta}}{\frac{\partial M_D}{\partial p}} \delta T'_u - \frac{1}{\frac{\partial M_D}{\partial p}} \delta S_{DC} + \\ & + \frac{\frac{\partial S_K}{\partial \theta} \frac{\partial T_s}{\partial p} + \frac{\partial S_K}{\partial p} - \frac{\partial S_{UT}}{\partial p}}{\frac{\partial M_D}{\partial p}} \cdot \delta p. \end{aligned}$$

Ekv (35).

$$\frac{d}{dt} \delta T'_{mi} = \frac{1}{\gamma_m} (-\delta T'_{mi} + \delta T_{mi}) = -\frac{1}{\gamma_m} \delta T'_{mi} + \frac{W_c}{\gamma_m \cdot W_m} \delta T_{cu}$$

Övriga ekvationer skall ej omformas.

Därefter beräknas matrisen A numeriskt, varefter dekomponering genomföres. Det har även skrivits en subrutin MARVIK, som beräknar A. Se appendix, 2.

Analysen har genomförts för full effekt.

DATA.

DAT(NR)	PARAMETER	VÄRDE	DAT(NR)	PARAMETER	VÄRDE
1	\emptyset	1.000 E 00	35	α_f	-6.000 E-03
2	W_K	1.764 E 03	36	α_s	-2.000 E-05
3	W_m	2.360 E 02	37	α_m	-1.000 E-04
4	W_p	2.360 E 02	38	α_x	-3.400 E-02
5	γ_f	8.000 E 00	39	$\partial F/\partial P$	2.645 E-03
6	γ_{ch}	2.000 E 00	40	$\partial F/\partial \theta$	-1.380 E-04
7	γ_m	3.800 E 02	41	$\partial F/\partial p$	-3.000 E-05
8	γ_{dc}	5.500 E 00	<u>52</u>	$1-C_{cc}$	5.300 E-01
9	γ_c	2.000 E 00			
10	γ_{DC}	1.000 E 02			
11	γ_s	2.630 E 02			
12	γ_u	2.250 E 02			
13	ΔT_γ	2.200 E 01	42	δ_I	9.500 E-01
14	q_i	1.000 E-03	43	γ_x	5.000 E-02
15	q_s	6.000 E-01	44	λ_I	2.890 E-05
16	A	2.000 E 01	45	λ_x	2.090 E-05
17	$\partial M_D/\partial p$	1.590 E 02	46	λ_1	1.240 E-02
18	$\partial T_s/\partial p$	1.250 E 00	47	λ_2	3.050 E-02
19	$\partial S_{UT}/\partial p$	4.800 E 00	48	λ_3	1.110 E-01
20	σ	1.400 E-04	49	λ_4	3.010 E-01
21	X	9.200 E-01	50	λ_5	1.140 E 00
22	$\partial S_K/\partial P$	2.970 E 02	51	λ_6	3.010 E 00
23	$\partial S_K/\partial \theta$	-6.510 E 00			
24	$\partial S_K/\partial p$	5.300 E-01			
25	$\partial W_K/\partial P$	-4.560 E 02			
26	$\partial W_K/\partial \theta$	4.260 E 00			
27	$\partial W_K/\partial p$	-5.700 E 00			
28	1	7.000 E-04			
29	β_1	2.130 E-04			
30	β_2	1.415 E-03			
31	β_3	1.266 E-03			
32	β_4	2.552 E-03			
33	β_5	7.440 E-04			
34	β_6	2.710 E-04			

Fall 1.

Vi tar ingen hänsyn till numeriska värden.

Vi beaktar ej, att $W_m = W_p$, dvs $W_c = W_m - W_p = 0$.

LR =

1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	1
2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
11	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
14	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
15	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
17	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1
18	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1

LREACH =

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Matrisen A är alltså ej reducerbar.

Fall 2.

Här beaktar vi, att W_c (moderatorsmassflödet från moderatorkylaren genom temperingspumpen) = 0.

LR =

1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	1	
2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	
7	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
8	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	
10	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	
11	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	
14	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	
15	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	0	0	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
17	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1
18	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1

LREACH =

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1

NREACH =

16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	1															1
3	0	:																	:
4	0	:																	:
5	0	:																	:
6	0																		
7	0																		
8	0																		
9	0																		
10	0																		
11	0																		
13	0																		
14	0																		
15	0																		
17	0																		
18	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

δT_{cu}

δT_{mi}

K = 3.
 $A(16,16) = -0.500000$
 $A(12,12) = -0.002632$

Fall 3.

Vi sätter alla element i LR svarande mot element i A, som är absolut mindre än 1.000E-03, med undantag för diagonalelement.

LR =

1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	0	1
2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
14	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
15	0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	0	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
17	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1
18	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1

LREACH=

1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
2	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
3	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
4	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
5	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
6	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
7	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
8	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
9	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
13	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
14	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
15	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
16	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1
18	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	0	1	1

NREACH=

16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	0															
1	0	1	1											1	1	0	1	0														
2	0	1														1	0	1	0														
3	0	.															0	1	0														
4	0	:																0	1	0													
5	0	:																	0	1	0												
6	0	:																		0	1	0											
7	0																				0	1	0										
8	0																					0	1	0									
9	0																						0	1	0								
13	0																							0	1	0							
14	0																								0	1	0						
15	0																									0	1	0					
17	0	1																									0	1	0				
18	0	1	1																								0	1	0			
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0			
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0		
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

S
X

K = 5. A(10,10) = - 0.000029
A(11,11) = - 0.000161

kommentarer till dekomponeringen.

I fall 2 fick vi således 3 st subsystem, varav de två omfattade en tillståndsvariabel. Motsvarande diagonalelement har angivits. Dessa är då egenvärden till matrisen A, vilket har verifierats i arbetsrapporten.

I fall 3 erhöles ytterligare 2 st subsystem, vilka omfattade en tillståndsvariabel, (jod- och xenonkoncentrationerna).

Härur skulle man då kunna utläsa att matrisen A även skulle ha egenvärdena - 0.000029 och - 0.000161.

Dessa återfinnes emellertid ej i utskriften i arbetsrapporten; de motsvaras troligen av + 0.000035 och + 0.000162.

Således är den gjorda approximationen i fall 3 ej tillfredsställande. För att bättre kunna utföra dylika approximationer, är det lämpligt att utföra en skalning av systemet, så att t.ex. alla variabler får variera mellan noll och ett.

Kap 5. Optimering.

Vi refererar här till Satos artikel "On a decomposition technique for multilevel systems". Vid den lokala optimeringen kan Krister Mårtenssons program LIOPCON ej användas i dess nuvarande skick på grund av förekomsten av termen $x_i^{\#}$ i ekv (8) :

$$\dot{x}_i = A_{ii} x_i + u_i + x_i^{\#}$$

Vi har därför nöjt oss med att utföra optimering för de olika subsystemen $\dot{x}_i = A_{ii} x_i + u_i$. Med andra ord försummar vi kopplingarna mellan subsystemen. Vi demonstrerar med två exempel.

$$1) \quad \frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 2 \\ 0 & -2 & 3 \\ 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$\text{Förlustfunktionen } J = \frac{1}{2} \int_{t_0}^{t_1} (x_1^2 + x_2^2 + x_3^2 + u_1^2 + u_2^2 + u_3^2) dt$$

Lokal optimering utföres alltså för följande tre system:

$$\begin{cases} \frac{dx_1}{dt} = -x_1 + u_1 \\ J_1 = \frac{1}{2} \int_{t_0}^{t_1} (x_1^2 + u_1^2) dt \end{cases}$$

$$\begin{cases} \frac{dx_2}{dt} = -2x_2 + u_2 \\ J_2 = \frac{1}{2} \int_{t_0}^{t_1} (x_2^2 + u_2^2) dt \end{cases}$$

$$\begin{cases} \frac{dx_3}{dt} = -3x_3 + u_3 \\ J_3 = \frac{1}{2} \int_{t_0}^{t_1} (x_3^2 + u_3^2) dt \end{cases}$$

Man kan visa de sistnämnda systemen har följande optimala styrlagar:

$$\begin{cases} L_1(t) = \frac{\tanh(\sqrt{2}(t_1-t))}{\sqrt{2} + \tanh(\sqrt{2}(t_1-t))} \\ L_2(t) = \frac{\tanh(\sqrt{5}(t_1-t))}{\sqrt{5} + \tanh(\sqrt{5}(t_1-t))} \\ L_3(t) = \frac{\tanh(\sqrt{10}(t_1-t))}{\sqrt{10} + \tanh(\sqrt{10}(t_1-t))} \end{cases}$$

(Även verifierat genom program LIOFCON).

$$2) \quad \frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 10 \\ 0 & -2 & 3 \\ 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

med samma förlustfunktion som i 1).

Vi får även samma subsystem som i 1) och därmed även samma lokala optimala styrlagar.

Enda skillnaden mellan de båda systemen är att vi i fall 2 har en stark koppling mellan subsystem 3 och subsystem 1, ($a_{13} = 10$).

Vi anger nu erhållna resultat vid körning med program LIOFCON, där L beräknades i 50 punkter med tidsdifferensen mellan dessa, TD = 0.2.

System 1.	System 2.	Lokal optimering.
$A = \begin{bmatrix} -1 & 1 & 2 \\ 0 & -2 & 3 \\ 0 & 0 & -3 \end{bmatrix}$	$A = \begin{bmatrix} -1 & 1 & 10 \\ 0 & -2 & 3 \\ 0 & 0 & -3 \end{bmatrix}$	$A \Leftrightarrow \begin{bmatrix} -1 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -3 \end{bmatrix}$

TD=0.0

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

TD=0.2

0.163	0.014	0.029	0.162	0.014	0.135	0.163			
0.014	0.138	0.037	0.014	0.138	0.049		0.136		
0.029	0.037	0.134	0.135	0.049	0.277				0.116

TD=0.4

0.265	0.039	0.083	0.255	0.035	0.344	0.266			
0.039	0.202	0.093	0.035	0.201	0.141		0.195		
0.083	0.093	0.229	0.344	0.141	0.831				0.149

TD=0.6

0.325	0.061	0.132	0.293	0.046	0.473	0.328			
0.061	0.234	0.138	0.046	0.228	0.208		0.219		
0.132	0.138	0.310	0.473	0.208	1.317				0.159

TD=0.8

0.359	0.075	0.167	0.304	0.049	0.521	0.365			
0.075	0.250	0.167	0.049	0.238	0.239		0.229		
0.167	0.167	0.371	0.521	0.239	1.561				0.161

TD=1.0

0.376	0.085	0.189	0.307	0.050	0.534	0.386			
0.085	0.258	0.185	0.050	0.242	0.248		0.233		
0.189	0.185	0.412	0.534	0.248	1.648				0.162

.....

TD=5.0

0.394	0.095	0.217	0.308	0.049	0.536	0.414			
0.095	0.267	0.207	0.049	0.243	0.250		0.236		
0.217	0.207	0.467	0.536	0.250	1.677				0.163

För TD=10.0 erhålles samma L-matriser.

Vi ser, att den största skillnaden i L-matriserna för system 1 resp system 2 uppstod i L_{33} , vilket kan tolkas så att pga den starka kopplingen mellan subsystem 3 och subsystem 1 får nedslagning av x_3 hög prioritet.

Antag, att vi låter t_1 gå mot oändligheten. Då är vi enbart intresserade av de konvergerade L-matriserna, L_k .

Givet: $\frac{dx}{dt} = Ax + Iu$

$$u = -L_k x$$

vilket ger $\frac{dx}{dt} = Ax - L_k x = (A-L_k)x$

med lösningen
$$x(t) = e^{(A-L_k)t} \cdot x(0), \quad \text{där vi satt } t_0=0.$$

Samt
$$u(t) = -L_k x(t).$$

Vi gör en jämförelse mellan system 1 och 2.

System 1.

$$A = \begin{bmatrix} -1 & 1 & 2 \\ 0 & -2 & 3 \\ 0 & 0 & -3 \end{bmatrix}$$

Lokal optimering.

$$L_k = \begin{bmatrix} 0.414 & 0.0 & 0.0 \\ 0.0 & 0.236 & 0.0 \\ 0.0 & 0.0 & 0.163 \end{bmatrix}$$

$$A-L_k = \begin{bmatrix} -1.414 & 1.0 & 2.0 \\ 0.0 & -2.236 & 3.0 \\ 0.0 & 0.0 & -3.163 \end{bmatrix}$$

System 2.

$$L_k = \begin{bmatrix} 0.414 & 0.0 & 0.0 \\ 0.0 & 0.236 & 0.0 \\ 0.0 & 0.0 & 0.163 \end{bmatrix}$$

$$A-L_k = \begin{bmatrix} -1.414 & 1.0 & 10.0 \\ 0.0 & -2.236 & 3.0 \\ 0.0 & 0.0 & -3.163 \end{bmatrix}$$

Vanlig optimering.

$$L_k = \begin{bmatrix} 0.394 & 0.095 & 0.217 \\ 0.095 & 0.267 & 0.207 \\ 0.217 & 0.207 & 0.467 \end{bmatrix}$$

$$A-L_k = \begin{bmatrix} -1.394 & 0.905 & 1.783 \\ -0.095 & -2.267 & 2.793 \\ -0.217 & -0.207 & -3.467 \end{bmatrix}$$

$$A = \begin{bmatrix} -1 & 1 & 10 \\ 0 & -2 & 3 \\ 0 & 0 & -3 \end{bmatrix}$$

$$L_k = \begin{bmatrix} 0.308 & 0.049 & 0.536 \\ 0.049 & 0.243 & 0.250 \\ 0.536 & 0.250 & 1.677 \end{bmatrix}$$

$$A-L_k = \begin{bmatrix} -1.308 & 0.951 & 9.464 \\ -0.049 & -2.243 & 2.750 \\ -0.536 & -0.250 & -4.677 \end{bmatrix}$$

Vi finner att den lokala optimeringen sammanfaller bättre med den korrekta optimeringen i systemet med måttlig koppling än systemet med stark koppling, vilket var väntat.

I den av Sato föreslagna algoritmen försummas ej x_i^{st} vid den lokala optimeringen (appendix 4). Härvid kan ej tillgängligt program LIOPCON användas, varför denna metod ej behandlas här.

En iakttagelse av stabilitet för flernivåsystem.

Givet:

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ & A_{22} & \cdots & A_{2k} \\ & & \ddots & \vdots \\ 0 & & & A_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix} \quad (1)$$

Vi visar nu, att följande påstående är sant.

Det totala systemet är stabilt om och endast om samtliga subsystem $\dot{x}_i = A_{ii} x_i + u_i \quad (1 \leq i \leq k)$ är stabila, oberoende av kopplingsmatriserna.

Påståendet följer omedelbart, ty egenvärdena till matrisen A utgöres av egenvärdena till submatriserna $A_{ii} \quad (1 \leq i \leq k)$, vilket är lätt att visa.

Påstående:

$$|xI - A| = \prod_{i=1}^k |xI - A_{ii}| \quad (2)$$

Bevis:

$$|xI - A| = \begin{vmatrix} xI - A_{11} & -A_{12} & -A_{13} & \cdots & -A_{1k} \\ & xI - A_{22} & -A_{23} & \cdots & -A_{2k} \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot \\ & & & & xI - A_{kk} \end{vmatrix}$$

Beviset göres som ett induktionsbevis.

(2) gäller uppenbarligen för $k = 1$.

Antag, att (2) gäller för $k = r$.

Påstående: Då gäller (2) också för $k = r + 1$, där $A_{r+1, r+1}$ är en $s \times s$ kvadratisk submatris.

Uppenbarligen gäller detta påstående för $s = 1$.

Vi vill då visa, att det gäller för ett godtyckligt s .

Antag, att det gäller för $s = p$.

Påstående: Då gäller det också för $s = p + 1$.

Bevis:

$$|xI - A| = \begin{vmatrix} xI-A_{11} & -A_{12} & \cdots & -A_{1r} & -A_{1,r+1} \\ & xI-A_{22} & \cdots & -A_{2r} & -A_{2,r+1} \\ & & \ddots & & \vdots \\ & & & xI-A_{rr} & -A_{r,r+1} \\ 0 & & & & \begin{bmatrix} x-a_{11} & \cdots & -a_{1p} & -a_{1,p+1} \\ \vdots & \ddots & \vdots & \vdots \\ \vdots & & \vdots & \vdots \\ -a_{p1} & \cdots & x-a_{pp} & -a_{p,p+1} \\ -a_{p+1,1} & \cdots & -a_{p+1,p} & x-a_{p+1,p+1} \end{bmatrix} \end{vmatrix} =$$

$$= (x-a_{p+1,p+1}) \cdot (-1)^{p+1+p+1} \cdot D_{p+1,p+1} \cdot \prod_{i=1}^r |xI-A_{ii}| +$$

$$+ (-a_{p+1,p}) \cdot (-1)^{p+1+p} \cdot D_{p+1,p} \prod_{i=1}^r |xI-A_{ii}| + \dots +$$

$$+ (-a_{p+1,1}) \cdot (-1)^{p+1+1} \cdot D_{p+1,1} \prod_{i=1}^r |xI-A_{ii}| .$$

Observera att matrisen $|xI - A|$ fortfarande är på block-triangulär form, efter det att rader och kolonner har strukits vid bildandet av underdeterminanterna, varför sista ledet följer av antagandet, att (2) gäller för $k = r + 1$, där $A_{r+1,r+1}$ är en $p \times p$ matris.

$D_{i,j}$ betecknar den underdeterminant till $xI - A_{r+1,r+1}$, som fås då rad i och kolonn j strykes.

$$\text{Alltså: } |xI - A| = \sum_{i=1}^{p+1} (xI-A_{r+1,r+1})_{p+1,i} \cdot (-1)^{p+1+i} \cdot D_{p+1,i}$$

$$\cdot \prod_{i=1}^r |xI-A_{ii}| = |xI-A_{r+1,r+1}| \cdot \prod_{i=1}^r |xI-A_{ii}| = \prod_{i=1}^{r+1} |xI-A_{ii}| .$$

Vi har härmed visat, att om (2) gäller för $k=r$ och (2) gäller för $k=r+1$, där $A_{r+1,r+1}$ är en $p \times p$ matris, så gäller (2) även för $k=r+1$, där $A_{r+1,r+1}$ är en $p+1 \times p+1$ matris. Eftersom (2) gäller för $p=1$, gäller alltså (2) för $k=r+1$, där $A_{r+1,r+1}$ är av godtycklig ordning.

Eftersom (2) gäller för $k=1$, gäller (2) för alla k .

Härmed är det ursprungliga påståendet bevisat.

Detta inser man även om man betraktar systemets struktur (sid 1).

Vi ser, att subsystemen är kopplade så att inga återkopplingar mellan olika subsystem förekommer, varför det totala systemet är stabilt, om subsystemen är det.

Referenser:

- 1) T. Sato, "On a decomposition technique for multilevel control systems", IEEE Trans. Automatic control. AC-12, p. 457 August 1967.
- 2) I. Lövgren, "Automatik del 2", föreläsningar vid LTH höstterminen 1965.
Sid 1:13 och 2:36.
- 3) K. Mårtensson, "Linear quadratic control package, part 1 - the continuous problem", inst. för Reglerteknik, LTH, Lund 1968.
- 4) O. Norinder, Atomenergi, Rapport R4-441, 19.4.1968.

Appendix 1. Program och subrutiner för dekomponering.

PROGRAM DEKOMPON

```

C
C   THIS PROGRAM MAKES A DECOMPOSITION OF THE MATRIX A INTO
C   A NUMBER OF BLOCK MATRIXES.
C   REFERS X-JOB B. AULIN KAP 1, 2
C   AUTHOR B. AULIN 24/1 - 69
C
C   A IS THE N * N KOEFFICIENT MATRIX
C   N - MAX = 40
C   BY PERMUTATION WITH VECTOR LD BY MATRIX A, MATRIX AN,
C   WHICH IS A BLOCK-TRIANGULAR MATRIX, IS RECEIVED.
C   IF ABS(A(I,J)) IS LESS THAN EPS LR(I,J)=0, ELSE LR(I,J)=1
C
C   FOLLOWING DATA SHALL BE GIVEN N, A, EPS
C   THE PROGRAM PRINTS OUT A, EPS, LR, LD, AN, LREACH.
C   THE MATRIX A IS NOT DESTROYED
C
C   REQUIRED SUBROUTINES
C
C                                     DEKOS
C                                     VEKTOR
C                                     REACH
C                                     MATMPY
C                                     NOBL
C
C   DIMENSION A(40,40),AN(40,40),LREACH(40,40),LR(40,40),LD(40)
5   READ 1000,N
1000  FORMAT(I4)
      READ 1001, EPS
1001  FORMAT(F7.4)
      DO 10 I=1,N
10    READ 1002,(A(I,J),J=1,N)
1002  FORMAT(10F5.1)
      PRINT 1007
1007  FORMAT(1H1,9H MATRIX A,/)
      IF(N.GT.20)GO TO 14
      DO 13 I=1,N
13    PRINT 1008,(A(I,J),J=1,N)
      GO TO 18
14    DO 15 I=1,N
15    PRINT 1008,(A(I,J),J=1,20)
1008  FORMAT(20F5.1)
      PRINT 1020
1020  FORMAT(1H1)
      DO 17 I=1,N
17    PRINT 1008,(A(I,J),J=21,N)
18    PRINT 1009, EPS
1009  FORMAT(///6H EPS =,F7.4)
      CALL DEKOS(A,N, EPS, AN, LD, NONRED, LR, LREACH)
      PRINT 1004
1004  FORMAT(1H1,10H MATRIX LR,/)
      DO 20 I=1,N
20    PRINT 1005,(LR(I,J),J=1,N)
1005  FORMAT(50I2)
      IF(NONRED.EQ.0)GO TO 40
      PRINT 1003

```



```
1003  FORMAT(16H A EJ REDUCERBAR)
      PRINT 1006
1006  FORMAT(1H1,14H MATRIX LREACH,/)
      DO 30 I=1,N
30    PRINT 1005,(LREACH(I,J),J=1,N)
      GO TO 60
40    PRINT 1010
1010  FORMAT(1H1,10H VEKTOR LD)
      PRINT 1011,(LD(I),I=1,N)
1011  FORMAT(40I3)
      PRINT 1012
1012  FORMAT(23H MATRIX A IN NORMALFORM,/)
      IF(N.GT.20)GO TO 45
      DO 42 I=1,N
42    PRINT 1008,(AN(I,J),J=1,N)
      GO TO 60
45    DO 50 I=1,N
50    PRINT 1008,(AN(I,J),J=1,20)
      PRINT 1020
      DO 55 I=1,N
55    PRINT 1008,(AN(I,J),J=21,N)
60    CONTINUE
      CALL EXIT
      END
```

	IDENT	DEKOMPON
PROGRAM LENGTH	15301	
ENTRY POINTS DEKOMPON	14562	
EXTERNAL SYMBOLS		
Q80ENTRY		
THEND.		
Q80DICT.		
DEKUS		
EXIT		
TSH.		
STH.		
QNSINGL.		
00172 SYMBOLS		

```

SUBROUTINE DEKUS(A,N,EPS,AN,LD,NONRED,LR,LREACH)
C
C   AUTHOR B. AULIN 24/1 - 69
C
C   A IS THE N x N KOEFFICIENT MATRIX
C   N = MAX = 40
C   BY PERMUTATION WITH VECTOR LD BY MATRIX A, MATRIX AN,
C   WHICH IS A BLOCK-TRIANGULAR MATRIX, IS RECEIVED.
C   IF ABS(A(I,J)) IS LESS THAN EPS LR(I,J)=0, ELSE LR(I,J)=1
C   IF A IS NOT REDUCIBLE, NONRED IS SET TO 1
C   INPUT DATA   A,N,EPS
C   OUTPUT DATA  AN,LD,NONRED,LR,LREACH
C   THE MATRIX A IS NOT DESTROYED
C
C   REQUIRED SUBROUTINES
C
C                                     VEKTOR
C                                     REACH
C                                     MATMPY
C                                     MOBL
C
C   DIMENSION A(40,40),AN(40,40),LD(40),LREACH(40,40),LR(40,40)
C   DO 20 I=1,N
C   DO 20 J=1,N
C   IF(ABS(A(I,J)).LE.EPS)GO TO 10
C   LR(I,J)=1
C   GO TO 20
10  LR(I,J)=0
20  CONTINUE
C   CALL REACH(LR,LREACH,N)
C   I=0
C   NONRED=0
C   DO 30 J=1,N
C   DO 30 K=1,N
C   IF(LREACH(J,K).EQ.0)I=1
30  CONTINUE
C   IF(I.EQ.1)GO TO 40
C   NONRED=1
C   GO TO 50
40  CALL VEKTOR(LREACH,N,LD)
C   CALL MOBL(A,N,LD,AN)
50  CONTINUE
C   RETURN
C   END

```

5.3DS

DEKUS

29/01-09

		IDENT	DEKUS
PROGRAM LENGTH		00243	
ENTRY POINTS	DEKUS	00003	
EXTERNAL SYMBOLS			
	080DICT.		
	REACH		
	VEKTOR		
	MOBL		
00114	SYMBOLS		

```

SUBROUTINE VEKTOR(LREACH,N,LD)
C
C   AUTHOR B. AULIN 24/1 - 69
C
C   COMPUTES THE PERMUTATION VECTOR LD
C   N IS THE ORDER OF THE REACHABILITY MATRIX, LREACH
C   N - MAX = 40
C
C   REQUIRED SUBROUTINES
C
C
C
C
C   DIMENSION LREACH(40,40),NREACH(40,40),LD(40),LE(40),MSTART(40)
C   DIMENSION KOMB(40)
C   DO 5 I=1,N
C   DO 5 J=1,N
5   NREACH(I,J)=LREACH(I,J)
C   DO 10 I=1,N
10  LD(I)=0
C   DO 20 I=1,N
20  LE(I)=I
C   N2=N
C   N3=N
30  NX=0
40  NRAD=1
45  NANT=0
C   DO 50 J=1,N2
C   IF(NREACH(NRAD,J).EQ.1)NANT=NANT+1
50  CONTINUE
C   IF(NANT.EQ.NX)GO TO 70
55  IF(NRAD.EQ.N2)GO TO 60
C   NRAD=NRAD+1
C   GO TO 45
60  NX=NX+1
C   GO TO 40
70  IF(NX.EQ.0)GO TO 165
C   DO 80 I=1,N2
80  MSTART(I)=NREACH(NRAD,I)
C   DO 90 I=1,N2
90  KOMB(I)=0
C   N1=N2-NX
C   I=1
C   DO 100 J=1,N2
C   IF(MSTART(J).EQ.0)GO TO 100
C   N7=N1+I
C   KOMB(N7)=J
C   I=I+1
100 CONTINUE
C   GO TO 170
165 DO 167 I=2,N2
167 KOMB(I-1)=0
C   KOMB(N2)=NRAD
170 I=N2
180 LKOMB=KOMB(I)
C   LD(N3)=LE(LKOMB)
C   N3=N3-1

```

```
IF(I.LE.N2-NX+1)GO TO 190
I=I-1
GO TO 180
190 DO 200 I=1,N2
200 LE(I)=LD(I)
CALL MOBL(LREACH,N,LE,NREACH)
IF(NX.EQ.0)N2=N2-1
N2=N2-NX
LUS=0
DO 210 I=1,N2
DO 210 J=1,N2
IF(NREACH(I,J).EQ.0)LUS=1
210 CONTINUE
IF(LUS.EQ.0)GO TO 220
IF(N2.EQ.1)GO TO 220
GO TO 30
220 RETURN
END
```

5.3DS

VEKTOR

29/01-69

	IDENT	VEKTOR
PROGRAM LENGTH	03672	
ENTRY POINTS	VEKTOR	03273
EXTERNAL SYMBOLS		
	080DICT.	
	MOBL	
00164 SYMBOLS		

```

SUBROUTINE REACH(LR,LREACH,N)
C
C   AUTHOR B. AULIN 24/1 - 69
C
C   COMPUTES THE REACHABILITY MATRIX FROM MATRIX LR
C   N IS THE ORDER OF THE MATRIXES
C   N - MAX = 40
C
C   REQUIRED SUBROUTINES
C
C                                     MATMPY
C
C
C   DIMENSION LR(40,40),LREACH(40,40),LP(40,40),LQ(40,40)
C   DO 2 I=1,N
C   DO 2 J=1,N
2   LREACH(I,J)=LR(I,J)
C   DO 5 I=1,N
C   DO 5 J=1,N
5   LP(I,J)=LR(I,J)
C   K=2
10  DO 15 I=1,N
10  DO 15 J=1,N
15  LQ(I,J)=LP(I,J)
C   CALL MATMPY(LR,N,LQ,LP)
C   DO 20 I=1,N
C   DO 20 J=1,N
20  LREACH(I,J)=LREACH(I,J)+LP(I,J)
C   DO 25 I=1,N
C   DO 25 J=1,N
C   IF(LREACH(I,J).EQ.0)GO TO 25
C   LREACH(I,J)=1
25  CONTINUE
C   M=0
C   DO 30 I=1,N
C   DO 30 J=1,N
C   IF (LP(I,J).EQ.LQ(I,J))M=M+1
30  CONTINUE
C   IF(M.EQ.N*N)GO TO 40
C   IF(K.EQ.N)GO TO 40
C   K=K+1
C   GO TO 10
40  RETURN
END

```


5.3DS

REACH

29/01-69

	IDENT	REACH
PROGRAM LENGTH	06473	
ENTRY POINTS REACH	06203	
EXTERNAL SYMBOLS		
OBODICT.		
MATMPY		
00132 SYMBOLS		

```
SUBROUTINE MATMPY(L1,N,L2,L3)
C
C   AUTHOR B. AULIN 24/1 - 69
C
C   COMPUTES LOGICAL MATRIX MULTIPLICATION  $L_1 = L_2 * L_3$ 
C   N IS THE ORDER OF THE MATRIXES
C   N = MAX = 40
C
C   REQUIRED SUBROUTINES
C
C
C
C
C   DIMENSION L1(40,40),L2(40,40),L3(40,40)
C   DO 30 I=1,N
C   DO 30 J=1,N
C   L3(I,J)=0
C   DO 20 K=1,N
20  L3(I,J)=L3(I,J)+L1(I,K)*L2(K,J)
C   IF(L3(I,J).EQ.0) GO TO 30
C   L3(I,J)=1
30  CONTINUE
C   RETURN
C   END
```

5.3DS

MATMPY

29/01-09

	IDENT	MATMPY
PROGRAM LENGTH	00173	
ENTRY POINTS	MATMPY	00003
EXTERNAL SYMBOLS	0800ICT.	
00050 SYMBOLS		

5.3DS

MOBL

29/01-69

	IDENT	MOBL
PROGRAM LENGTH	00211	
ENTRY POINTS	MOBL	00003
EXTERNAL SYMBOLS		
	Q8QDICT.	
00075 SYMBOLS		

LOAD

RUN,3,500, , ,

PROGRAM NAMES	15301	0 62233	DEKUS	00243	0 56341	VEKTUR	03572	0 47046	REACH	00470
0 62476 DEKOMPON	00173	0 47242	MOBL	00211	0 47000	ALLUC.	00234	0 44537	IUR.	02247
0 47453 MATMPY	00204	0 42762	IOP.	01351	0 42724	SIH.	00030	0 42057	ISH.	00045
0 44333 080ERROR	00423	0 42154	080ENTRY	00000						

PROGRAM EXTENS.
NONE

LABELED COMMON
NONE

NUMBERED COMMON
NONE

ENTRY POINTS	0 77777	0 72260	DEKUMPON	0 42100	080ENTRY	0 42553	IEND.
0 42154 080DICI.	0 62236	DEKUS	0 42155	EXIT	0 42603	ISH.	
0 42730 SIH.	0 42545	ONSINGL.	0 50051	REACH	0 01034	VEKTUR	
0 47245 MOBL	0 47456	MATMPY	0 42765	IUP.	0 42307	080HIST.	
0 44334 080ERROR	0 44563	IUH.	0 42254	IUS.	0 42557	IUR.	
0 46661 .TSERR.	0 44537	BCDBUF.	0 47144	ALLOC.	0 47014	RETURN.	
0 47120 BUSY.	0 47127	IRETURN.	0 42550	UNDOUBL.	0 47210	ALLUCIN.	
0 46354 ELD.	0 46640	.REPCNT.	0 44333	080IRAC	0 44444	080ERSET	
0 42234 IOE.	0 42426	080CHAIN	0 42233	080TIME.			

EXECUTION STARTED AT 2107 -14

Appendix 2. Subrutinen MARVIK.

BESKRIVNING AV INDATA OCH TILLSTÅNDSVEKTOR FÖR SUBROUTINEN MARVIK.

Parameter	DAT(NR)	Parameter	DAT(NR)	Variabel	X(NR)
\emptyset	1	α_f	35	$\delta\emptyset$	1
W_K	2	α_s	36	δC_1	2
W_m	3	α_m	37	δC_2	3
W_p	4	α_X	38	δC_3	4
τ_f	5	$\partial F/\partial P$	39	δC_4	5
τ_{ch}	6	$\partial F/\partial \theta$	40	δC_5	6
τ_m	7	$\partial F/\partial p$	41	δC_6	7
τ_{dc}	8	$(1-C_{cc})$	<u>52</u>	δP_f	8
τ_c	9			δP	9
τ_{DC}	10	δ_I	42	δI	10
τ_s	11	δ_X	43	δX	11
τ_u	12	λ_I	44	$\delta T'_{mi}$	12
ΔT_v	13	λ_X	45	$\delta T'_s$	13
a_i	14	λ_1	46	$\delta T'_\gamma$	14
a_s	15	λ_2	47	$\delta T'_u$	15
A	16	λ_3	48	$\delta T'_{cu}$	16
$\partial M_D/\partial P$	17	λ_4	49	δS_{DC}	17
$\partial T_s/\partial p$	18	λ_5	50	δp	18
$\partial S_{UT}/\partial p$	19	λ_6	51		
σ	20				
X	21				
$\partial S_K/\partial P$	22				
$\partial S_K/\partial \theta$	23				
$\partial S_K/\partial p$	24				
$\partial W_K/\partial P$	25				
$\partial W_K/\partial \theta$	26				
$\partial W_K/\partial p$	27				
l	28				
β_1	29				
β_2	30				
β_3	31				
β_4	32				
β_5	33				
β_6	34				

Numeriska värden, se sid 34.


```
PROGRAM TEST
DIMENSION A(18,18),B(18,1),C(1,18),DAT(52)
READ 100,DAT
100  FORMAT(8E10.3)
     CALL MARVIK(A,B,C,DAT)
     PRINT 110
110  FORMAT(9H MATRIX A,/)
     DO 10 I=1,18
10   PRINT 120,(A(I,J),J=1,9)
120  FORMAT(1X,9E12.3)
     PRINT 130
130  FORMAT(1H0)
     DO 20 I=1,18
20   PRINT 120,(A(I,J),J=10,18)
     PRINT 140
140  FORMAT(9H1MATRIX B,/)
     DO 30 I=1,18
30   PRINT 150,B(I,1)
150  FORMAT(E12.3)
     CALL EXIT
     END
```

	IDENT	TEST
PROGRAM LENGTH	01057	
ENTRY POINTS	TEST	00671
EXTERNAL SYMBOLS		
	080ENTRY	
	THEND.	
	080DICT.	
	MARVIK	
	EXIT	
	TSH.	
	STH.	
	SLI.	
	0NSINGL.	
00060 SYMBOLS		

```

SUBROUTINE MARVIK(A,B,C,DAT)
C
C REFERENS OLOF NORINDER, ATOMENERGI, RAPPORT R4-441, 19.4.68
C
C BERAKNAR MATRISERNA A,B OCH C I SYST-EKV    $DX/DI = A*X + B*U$ 
C                                              $Y = U*X$ 
C
C  $X = 18 * 1$  TILLST-VEKTOR = DE I EKV (24) - (41) TIDS DERIVERADE
C   VARIABLERNA.
C
C  $U =$  STYRSIGNALEN = DELTA-R, JFR EKV (51).
C
C SOM Y = UTSIGNAL HAR VALTS AVVIKELSEN I DEN NUKLEARA EFFEKTEN=X(1)
C
C   AUTHOR B. AULIN 24/1 - 69
C
C SOM INDATA TILL RUTINEN KRAVS DAT(1) - DAT(52)
C VILKA ANGES I EXAMENSARBETE B. AULIN APPENDIX 2
C SÄMT I REF NORINDER.
C A,B,C, FAS SOM INDATA
C
C REQUIRED SUBROUTINES
C
C                                     NONE
C
C
C DIMENSION A(18,18),B(18,1),C(1,18),DAT(52)
C BETA=0.
C DO 10 I=29,34
10  BETA=BETA+DAT(I)
C DM=(1.-BETA)*DAT(1)/DAT(28)
C
C MATRISEN B GENERERAS
C
C B(1,1)=DM
C DO 20 I=2,18
20  B(1,I)=0.
C
C MATRISEN C GENERERAS
C
C C(1,1)=1.
C DO 30 I=2,18
30  C(1,I)=0.
C
C MATRISEN A GENERERAS
C
C DO 40 I=1,18
C DO 40 J=1,18
40  A(I,J)=0.
C A(1,1)=-BETA/DAT(28)
C DO 50 I=2,7
50  A(1,I)=DAT(1+44)
C A(1,8)=DM*DAT(35)
C A(1,9)=DM*DAT(39)
C A(1,11)=DM*DAT(38)
C A(1,12)=0.5*DM*DAT(37)*DAT(14)

```

```

A(1,13)=0.5*DM*DAT(37)*DAT(15)
A(1,14)=0.5*DM*DAT(37)
A(1,15)=-DM*DAT(40)
A(1,16)=0.5*DM*DAT(37)*(DAT(3)-DAT(4))/DAT(3)
A(1,18)=DM*(DAT(36)*DAT(18)+DAT(40)*DAT(18)+DAT(41))
DO 60 I=2,7
60  A(I,1)=-DAT(I+44)
A(1,1)=DAT(1+27)/DAT(28)
A(8,1)=1./DAT(5)
A(8,8)=-A(8,1)
A(9,8)=1./DAT(6)
A(9,9)=-A(9,8)
A(10,1)=DAT(42)*DAT(20)
A(10,10)=-DAT(44)
A(11,1)=DAT(43)*DAT(20)-DAT(21)*DAT(20)
A(11,10)=DAT(44)
A(11,11)=-DAT(45)-DAT(20)*DAT(1)
A(12,12)=-1./DAT(7)
A(12,16)=(DAT(3)-DAT(4))/DAT(3)/DAT(7)
A(13,13)=-2./DAT(7)
A(13,18)=2.*DAT(18)/DAT(7)
A(14,1)=2.*DAT(13)/DAT(7)/DAT(1)
A(14,14)=A(13,13)
DL=1./DAT(8)/(DAT(2)+DAT(3))
DM1=DAT(3)*DL
DN=(DAT(12)-DAT(11))*DL
A(15,9)=-DN*DAT(25)
A(15,12)=DM1*DAT(14)
A(15,13)=DM1*DAT(15)
A(15,14)=DM1
A(15,15)=-1./DAT(8)+DN*DAT(26)
A(15,18)=DAT(9)*DAT(18)*DL-DN*(DAT(26)*DAT(18)+DAT(27))
A(16,15)=DAT(52)/DAT(9)
A(16,16)=-1./DAT(9)
A(16,16)=-1./DAT(9)
DP=DAT(16)/DAT(10)/DAT(17)
A(17,9)=DP*DAT(22)
A(17,15)=-DP*DAT(23)
A(17,17)=-1./DAT(10)-DP
A(17,18)=DP*(DAT(23)*DAT(18)+DAT(24)-DAT(19))
A(18,9)=DAT(22)/DAT(17)
A(18,15)=-DAT(23)/DAT(17)
A(18,17)=-1./DAT(17)
A(18,18)=(DAT(23)*DAT(18)+DAT(24)-DAT(19))/DAT(17)
RETURN
END

```

5.30S MARVIK

30701-09

	IDENT	MARVIK
PROGRAM LENGTH	00550	
ENTRY POINTS	MARVIK	00003
EXTERNAL SYMBOLS		08001CT.
00220 SYMBOLS		

LOAD
RUN,3,1500, , ,

PROGRAM NAMES
 0 76720 TEST
 0 73241 080ERRRJR
 0 71513 TSH.

PROGRAM EXTENS.
 NONE

LABELED COMMON
 NONE

NUMBERED COMMON
 NONE

ENTRY POINTS
 0 77777 SENTRY
 0 71010 08QDICT.
 0 71504 STH.
 0 71143 08JHISI.
 0 71413 IOR.
 0 76052 ALLOC.
 0 70110 ALLOCIN.
 0 73352 08QERSET
 0 71067 050TIME.

MARVIK
 IOP.
 IUS.

70150
 0 71670
 0 71070

01057
 00204
 00045

00550
 01351
 00423

75714
 0 71010
 0 71010

ALLUC.
 SLI.
 080ENIRY

00264
 00052
 00000

0 73445
 0 71200

I JH.
 STH.

0 71407
 0 71517
 0 71573
 0 71110
 0 71404
 0 76035
 0 73241
 0 71202

THEVD.
 TST.
 IOP.
 IUS.
 UNDUJUBL.
 IREIUKN.
 03VUFRAC
 U0JCHAIN

02247
 00036

EXECUTION STARTED AT 1053 -35

Appendix 3. Olof Norinder, Atomenergi, Rapport R4-441.



Arbetsrapport

Org enh och nr
AE-

Titel och författare
Matematikmaskinprogrammet MKST för beräkning av
Marvikenreaktorns stabilitet

Olov Norinder

P 9410

AE dec reg	Org enh och nr R4-441 AE- RFR-711
Antal ex/Anlat sid 9	Datum 19.4.68
Godkänd av <i>Olof Carlberg</i>	

Utsändes till

FD	1 ex
FTB-Sthlm	3 ex
FTB-Studsvik	2 ex

Vattenfall 10 ex
 ASEA 10 ex

Ett matematikmaskinprogram benämnt MKST har skapats för bestämning av Marvikenreaktorns stabilitet. Programmet är avsett att väl avbilda förhållandena vid alla effektnivåer. Beräkningstiderna är små.

Principer för beräkning av stabiliteten

I programmets nuvarande utformning undersöks den oreglerade reaktorns dynamiska egenskaper för små svängningar. Alla ekvationer lineariseras genom att variablerna ersättes med stationärvärdet + avvikelse, t.ex. för nukleär effekt $\phi = \phi + \delta\phi$. Det resulterande lineära systemet av ordinära lineära differentialekvationers dynamiska egenskaper beskrivs som bekant fullständigt av systemets egenlösning. I MKST bestäms egenlösningen numeriskt.

Ekvationer

I följande ekvation är ϕ nukleär effekt, λ livslängd för prompta neutroner, δk avvikelsen i k-effektiv från 1, C_i koncentrationer av fördröjda neutroner, λ_i sönderfallskonstanter och β summa andelar fördröjda neutroner. Man får följande ekvation:

$$\frac{d}{dt} \phi = \frac{1}{\lambda} \left[(1 + \delta k)(1 - \beta) - 1 \right] \phi + \sum_i \lambda_i C_i \quad (1)$$

För koncentrationerna av fördröjda neutroner har man (de fördröjda neutronerna räknas som genererade med $k_{eff} = 1$):

$$\frac{d}{dt} C_i = \frac{1}{\lambda} \beta_i \phi - \lambda_i C_i \quad (2)-(7)$$

$i = 1, 6$

Effekten från bränslet till kylmediet, P_f , sammanhänger med tidskonstanten τ_f med den nukleära effekten. Man får:

$$\frac{d}{dt} P_f = \frac{1}{\tau_f} (-P_f + \phi) \quad (8)$$

Kokarelementen antages vara med god marginal hydrodynamiskt stabila. Kanalerna räknas "kvasistationärt". Detta innebär, att jag representerar de kokande kanalerna med en ("kontinuerlig") följd av stationära lösningar. Kanalernas effekt P beskrivs såsom med tidskonstanten τ_{ch} sammanhängande med effekten P_f från bränslet.

$$\frac{d}{dt} P = \frac{1}{\tau_{ch}} (-P + P_f) \quad (9)$$

Ekvationerna för jod, I, och xenon, X, beskrives i det bekväma systemet med mättnadskoncentrationen för xenon som enhet. Således erhålles:

$$\frac{d}{dt} I = \gamma_I \sigma \phi - \lambda_I I \quad (10)$$

$$\frac{d}{dt} X = \gamma_X \sigma \phi + \lambda_I I - \lambda_X X - \sigma \phi X \quad (11)$$

I ovanstående ekvationer är γ_I relativ andel jod, γ_X relativ andel xenon ($\gamma_I + \gamma_X = 1$), σ mikroskopiska tvärsnittet för xenon (obs. att σ skall givas så, att produkten $\sigma \phi$ ger rätt reaktionshastighet), λ_I sönderfallskonstanten för jod och λ_X sönderfallskonstanten för xenon.

Väsentligheterna i vattnets och ångans cirkulation i Marviken framgår av fig 1. Låt M_D vara ångmassa i domen, S_{DC} ångmassflöde kondenserande i domen, S_{UT} ångmassflöde utblåsande från domen, S_K ångmassflöde från kokarkanalerna och p trycket. Då har man för domen:

$$\frac{\delta M_D}{\delta p} \frac{dp}{dt} + S_{DC} + S_{UT}(p) = S_K \quad (12)$$

Kondensation (och avdunstning) i domen beskrives med följande ekvation, där bl.a. tidskonstanten τ_{DC} ingår:

$$\tau_{DC} \frac{dS_{DC}}{dt} + S_{DC} = A \frac{dp}{dt} \quad (13)$$

Mellan mättnadstemperatur T_s och tryck har vi ett funktionssamband $T_s(p)$.

För temperaturen T_u på vattnet utgående till fallspalten har vi följande samband, där mättnadstemperaturen T_s , och moderators utgående temperatur T_{mu} , vätskemassflöde ut från kokarna W_K och moderatormassflöde W_m ingår:

$$T_u = \frac{W_K T_s + W_m T_{mu}}{W_K + W_m} \quad (14)$$

Transporttiden genom fallspalten representeras med en tidskonstant τ_{dc} och man får för temperaturen T'_u efter fallspalten:

$$\tau_{dc} \frac{dT'_u}{dt} + T'_u = T_u \quad (15)$$

För underkylningen har man:

$$\mathcal{J} = T_s - T'_u \quad (16)$$

Vätskemassflödet ut från kokarna W_K , ångmassflödet ut från kokarna S_K och reaktivitetsverkan från void F betraktas som funktioner av total effekt från bränslet till kokarna P , underkylning \mathcal{J} och tryck (mättnadstryck) p . Således har man funktionerna

$$W_K(P, \mathcal{J}, p) \quad S_K(P, \mathcal{J}, p) \quad \text{och} \quad F(P, \mathcal{J}, p).$$

I beräkningarna inkommer huvudsakligen de partiella derivatorerna av funktionerna. Funktionerna och deras derivator bestäms med HYDROST (statiska HYDRO) och voidviktningsprogrammet GEVO.

I verkligheten påverkar dynamiska moderatortemperaturändringar voidbilden även via värmeledningen mellan kokarkanalerna och moderatoren. Emellertid torde inverkan vara ringa. Angiven dynamisk verkan av moderator-temperaturförändringar försummas i MKST.

Moderatormassflödet är sammansatt av delen W_p från matarvattenpumpen och delen W_c från moderatorkylaren genom tempereringspumpen. Således:

$$W_m = W_p + W_c \quad (17)$$

Ingångstemperaturen till moderatoren T_{ni} erhålles från blandning av matarvatten med temperaturen T_p och vatten från moderatorkylaren med temperaturen T_{cu} :

$$T_{ni} = \frac{W_c T_{cu} + W_p T_p}{W_m} \quad (18)$$

Efter moderatororkylaren kvarstår andelen $(1 - C_{cc})$ av temperaturen före kylaren. Dessutom representerar vi transporttiden med en tidskonstant τ_c :

$$\tau_c \frac{dT_{cu}}{dt} + T_{cu} = (1 - C_{cc})(T_s - \theta) \quad (19)$$

Moderatorns ingångstemperatur uppträder tidsförskjuten i ett bidrag i moderatorns utgångstemperatur. Beteckna den tidsförskjutna temperaturen T'_{mi} . Tidsförskjutningen representeras med en tidskonstant τ_m :

$$\tau_m \frac{dT'_{mi}}{dt} + T'_{mi} = T_{mi} \quad (20)$$

Även mättnadstemperaturen ger tidsförskjuten ett bidrag i moderatorns utgångstemperatur. Det tidsförskjutna värdet betecknas T'_s . Förskjutningen beskrives med tidskonstanten $\tau_m/2$:

$$(\tau_m/2) \frac{dT'_s}{dt} + T'_s = T_s \quad (21)$$

Gammavärmning och neutronnedbromsning ger även ett tidsförskjutet temperaturbidrag T'_γ till moderatorns utgångstemperatur. Låt nukleära moderatortemperaturhöjningen vara ΔT_γ och effekten ϕ_0 för jämviktstillståndet. Följande ekvation ansättes för temperaturbidraget:

$$(\tau_m/2) \frac{dT'_\gamma}{dt} + T'_\gamma = \frac{\Delta T_\gamma}{\phi_0} \phi \quad (22)$$

Andelen q_i av moderatorns ingångstemperatur och andelen q_s av mättnadstemperaturen antages återfinnas i moderatorns utgångstemperatur T_{mu} . Således:

$$T_{mu} = q_i T'_{mi} + q_s T'_s + T'_\gamma \quad (23)$$

Det lineariserade systemet för små störningar bildas genom att variablerna ersättes med stationärvärden + tillskott med δ som "tillskottsoperator". Efter enklaste manipulationer erhålles följande ekvationer med tidsderivata:

$$\frac{d}{dt} \delta\phi = -\frac{1}{\ell} \beta \delta\phi + \frac{1}{\ell} (1 - \beta) \phi \delta k + \sum_i \lambda_i \delta C_i \quad (24)$$

$$\frac{d}{dt} \delta C_i = \frac{1}{\ell} \beta_i \delta\phi - \lambda_i \delta C_i \quad (25)-(30)$$

$$i = 1, 6$$

$$\frac{d}{dt} \delta P_f = \frac{1}{\tau_f} (-\delta P_f + \delta\phi) \quad (31)$$

$$\frac{d}{dt} \delta P = \frac{1}{\tau_{ch}} (-\delta P + \delta P_f) \quad (32)$$

$$\frac{d}{dt} \delta I = \gamma_I \sigma \delta\phi - \lambda_I \delta I \quad (33)$$

$$\frac{d}{dt} \delta X = \gamma_X \sigma \delta\phi + \lambda_I \delta I - \lambda_X \delta X - \sigma\phi \delta X - \sigma X \delta\phi \quad (34)$$

$$\frac{d}{dt} \delta T'_{mi} = \frac{1}{\tau_m} (-\delta T'_{mi} + \delta T_{mi}) \quad (35)$$

$$\frac{d}{dt} \delta T'_s = \frac{2}{\tau_m} (-\delta T'_s + \delta T_s) \quad (36)$$

$$\frac{d}{dt} \delta T'_\gamma = \frac{2}{\tau_m} \left(-\delta T'_\gamma + \frac{\Delta T_\gamma}{\phi} \delta\phi \right) \quad (37)$$

$$\frac{d}{dt} \delta T'_u = \frac{1}{\tau_{dc}} (-\delta T'_u + \delta T_u) \quad (38)$$

$$\frac{d}{dt} \delta T_{cu} = \frac{1}{\tau_c} \left(-\delta T_{cu} + (1 - C_{cc}) \delta T'_u \right) \quad (39)$$

$$\frac{d}{dt} \delta S_{DC} = -\frac{1}{\tau_{DC}} \left(1 + \frac{A}{\frac{\partial M_D}{\partial p}} \right) \delta S_{DC} + \frac{A}{\tau_{DC} \cdot \frac{\partial M_D}{\partial p}} (\delta S_K - \delta S_{UT}) \quad (40)$$

$$\frac{d}{dt} \delta p = \frac{1}{\frac{\partial M_D}{\partial p}} (\delta S_K - \delta S_{DC} - \delta S_{UT}) \quad (41)$$

Man har ytterligare ett antal lineära samband mellan introducerade variabler, vilka samband delvis erhålles genom differentiering:

$$T_u \delta W_K + (W_K + W_m) \delta T_u = T_s \delta W_K + W_K \delta T_s + W_m \delta T_{mu} \quad (42)$$

$$\delta T_{mi} = (W_c / W_m) \delta T_{cu} \quad (43)$$

$$\delta T_{mu} = q_i \delta T'_{mi} + q_s \delta T'_s + \delta T'_y \quad (44)$$

$$\delta T_s = \frac{\partial T_s}{\partial p} \delta p \quad (45)$$

$$\delta \psi = \delta T_s - \delta T'_u \quad (46)$$

$$\delta S_K = \frac{\partial S_K}{\partial P} \delta P + \frac{\partial S_K}{\partial \psi} \delta \psi + \frac{\partial S_K}{\partial p} \delta p \quad (47)$$

$$\delta W_K = \frac{\partial W_K}{\partial P} \delta P + \frac{\partial W_K}{\partial \psi} \delta \psi + \frac{\partial W_K}{\partial p} \delta p \quad (48)$$

Utblåst ångmängd är funktion av trycket. Genom derivering erhålles:

$$\delta S_{UT} = \frac{\partial S_{UT}}{\partial p} \delta p \quad (49)$$

Jag introducerar moderatormedeltemperaturen T_m och sätter för denna:

$$\delta T_m = .5 (\delta T_{mi} + \delta T_{mu}) \quad (50)$$

Slutligen uppskrives den viktiga ekvationen för reaktivitetsverkningar. Introducerade reaktivitetskoefficienter framgår av ekvationen:

$$\delta k = \alpha_f \delta P_f + \alpha_s \delta T_s + \alpha_m \delta T_m + \alpha_x \delta X + \frac{\partial F}{\partial P} \delta P + \frac{\partial F}{\partial \nu} \delta \nu + \frac{\partial F}{\partial p} \delta p + \delta R \quad (51)$$

Den sista termen står för ett yttre reaktivitetsbedrag (t.ex. från styrtavar).

Bestämning av stabiliteten

Avvikelseerna som ingår tidsderiverade sammanfattar jag i vektorn $\underline{\zeta}$.

Avvikelseerna som ingår utan tidsderivata sammanfattar jag i vektorn $\underline{\eta}$.

Efter Laplace-transformation och omdefiniering erhålles följande matrisekvation för ekvationerna (24) - (41):

$$\underline{E} \underline{\zeta} + \underline{F} \underline{\eta} - s \underline{I} \underline{\zeta} = 0 \quad (52)$$

Ekvationerna (42) - (51) ger efter Laplace-transformation och omdefiniering matrisekvationen:

$$\underline{C} \underline{\eta} = \underline{D} \underline{\zeta} \quad (53)$$

Stabiliteten studeras genom lösning av följande egenproblem:

$$|(\underline{E} + \underline{F} \underline{C}^{-1} \underline{D}) - s \underline{I}| \underline{\zeta} = 0 \quad (54)$$

Beskrivning av initial input till MKST

Förutom titel ingår i rubr. endast reella tal. Genomgående användes FORMAT (12F6.6) för inläsningen.

På kort 1 läses TITELN på vanligt sätt.

På kort 2 läses data nr 1 till 12, vilka är:

ϕ , W_K , W_m , W_p , τ_f , τ_{ch} , τ_m , τ_{dc} , τ_c , τ_{DC} , T_s och T_u .

På kort 3 läses data nr 13 till 24, vilka är:

ΔT_Y , q_i , q_s , A , $\partial M_D / \partial p$, $\partial \tau_s / \partial p$, $\partial S_{UT} / \partial p$, σ , X , $\partial S_K / \partial P$, $\partial S_K / \partial \nu$ och $\partial S_K / \partial p$.

På kort 4 läses data nr 25 till 34, vilka är $\partial W_K / \partial P$, $\partial W_K / \partial \nu$, $\partial W_K / \partial p$, λ , β_1 , β_2 , β_3 , β_4 , β_5 och β_6 .

På kort 5 läses data nr 35 till 41 och nr 52, vilka är α_j , α_s , α_m , α_x , $\partial F/\partial P$, $\partial F/\partial N$, $\partial F/\partial P$ och $(1-C_{cc})$.

Vissa data är permanent inskrivna i MKST, nämligen:

$\gamma_I = .95$, $\gamma_X = .05$, $\lambda_I = 2.89 \text{ E-}5$, $\lambda_X = 2.09 \text{ E-}5$, $\lambda_1 = .0124$, $\lambda_2 = .0305$, $\lambda_3 = .111$, $\lambda_4 = .301$, $\lambda_5 = 1.14$ och $\lambda_6 = 3.01$.

Beskrivning av icke initial input till MKST:

Frångår omedelbart ur följande avskrift av en subroutine:

```

SUBROUTINE NDA
COMMON TIT(14), DA (70)
DIMENSION IND(6), D(6)
3 READ (5, 1) IG, I, IND, D
1 FORMAT (8I3, 6F.6)
IF(IG.NE.2)GOTO 2
CALL IDA
RETURN
2 DO 4 J = 1,I
  K=IND(J)
4 DA(K)=D(J)
IF(IG.EQ.3)GOTO 3
RETURN
END

```

Subrutinen IDA läser initialdata. K i DO-loopen motsvarar datanummer.

Beskrivning av output från MKST

Outputen förklaras praktiskt taget fullständigt av ingående text. För varje beräkningsfall tryckes ingångsdata uppställt nästan likformigt med initialdata.

FALL NR 1 VID BERAKNINGSTILLFALLET
PK 1.0 X FULL EFFEKT
FALL 1

1.000E C0 1.764E-03 2.360E-02 8.000E C0 2.000E C0 3.800E-02 5.500E C0 2.000E C0 1.000E C2 2.630E-02 2.250E-02
 2.200E C1 1.000E-03 6.000E-01 2.000E C1 1.590E C2 1.250E C0 4.800E C0 1.400E-04 9.200E-01 2.970E C2-6.510E C0 5.300E-01
 -4.560E C2 4.260E C0-5.700E C0 7.000E-04 2.130E-04 1.415E-03 1.266E-03 2.552E-03 7.440E-04 2.710E-04
 -6.000E-03-2.000E-05-1.000E-04-3.400E-02 2.645E-03-1.380E-04-3.000E-05 5.000E-01
 9.500E-01 2.890E-05 5.000E-02 2.090E-05 1.240E-02 3.050E-02 1.110E-01 3.010E-01 1.140E C0 3.010E C0

EGENVARDEN CCH EGENVEKTORER

REAL -5.581522 -2.847861 -C.980448 -0.601188 -0.500000 -0.271088 -0.118225 -0.118225 -0.118225 -0.108909
 IMAG C.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.136700 0.136700 0.136700 C.000000

1 FI 1.224E-04 2.870E-06 5.158E-08 3.195E-09 -3.913E-02 1.732E-10 7.235E-10 1.164E-05 3.476E-10
 8P-FUEL -1.618E-06 -1.314E-07 -7.081E-09 -3.360E-10 4.301E-10 5.872E-10 -5.334E-10 5.436E-10 5.134E-11
 9 P-KAN 8.974E-08 2.850E-08 8.161E-09 4.011E-09 -7.181E-02 7.864E-10 -8.564E-10 2.220E-10 -6.867E-11

10 ICC 8.741E-09 1.031E-08 1.043E-08 1.044E-08 1.041E-05 1.044E-08 9.516E-09 4.299E-05 1.044E-08
 11 XE 1.606E-09 1.727E-10 5.629E-11 5.045E-11 -9.536E-06 4.978E-11 4.610E-11 2.087E-11 4.973E-11
 12T-IMCC -2.614E-05 -2.613E-05 -2.608E-05 -2.604E-05 -1.450E-07 -2.590E-05 -2.348E-05 -1.094E-05 -2.550E-05

13TS-MCC 8.575E-01 8.575E-01 8.575E-01 8.575E-01 0. 8.575E-01 7.814E-01 3.532E-01 8.575E-01
 14 T-GAM -5.145E-01 -5.145E-01 -5.145E-01 -5.145E-01 6.210E-03 -5.145E-01 -4.688E-01 -2.119E-01 -5.145E-01
 15T-UDWA 1.346E-13 -9.180E-13 2.531E-15 1.725E-15 -5.204E-01 2.613E-15 1.600E-15 6.737E-16 1.315E-15

16T-UMCC 0. -0. -0. -0. -0. -0. -0. -0. -0. -0.
 17CCMKNC -2.540E-09 -2.867E-09 -2.425E-09 -1.927E-09 7.540E-02 -6.085E-10 8.749E-10 -1.970E-09 1.655E-09
 18 TRYCK -1.765E-08 -1.925E-08 -1.698E-08 -1.447E-08 3.695E-01 -7.970E-09 0. -1.130E-08 2.354E-09

REAL -C.036587 -C.029510 -C.012021 -C.009791 -C.006159 -0.005263 -0.002632 0.000000 0.000000 C.000162
 IMAG C.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 C.000000

1 FI 3.405E-10 4.190E-10 1.197E-09 -1.297E-10 -7.042E-10 -1.144E-10 -2.125E-06 9.493E-10 -3.708E-10
 8P-FUEL 3.259E-10 4.344E-10 1.290E-09 -1.665E-10 -7.538E-10 -1.295E-10 -2.171E-06 9.563E-10 -3.626E-10
 9 P-KAN 2.969E-10 4.134E-10 1.131E-09 -2.013E-10 -1.043E-09 1.690E-10 -5.889E-06 9.332E-10 1.160E-09

10 ICC 1.045E-08 1.045E-08 1.047E-08 1.045E-08 1.053E-08 1.053E-08 1.073E-07 5.066E-05 7.718E-09
 11 XE 4.928E-11 4.942E-11 5.664E-11 6.705E-11 2.425E-11 3.382E-11 -1.061E-07 -1.005E-10 1.495E-10
 12T-IMCC -2.415E-05 -2.355E-05 -1.882E-05 -1.654E-05 -6.640E-06 1.093E-10 9.919E-01 -5.196E-05 -5.078E-05

13TS-MCC 8.575E-01 8.575E-01 8.575E-01 8.575E-01 8.575E-01 8.575E-01 -1.090E-01 8.575E-01 8.575E-01
 14 T-GAM -5.145E-01 -5.145E-01 -5.145E-01 -5.145E-01 -5.145E-01 6.508E-02 -5.145E-01 -5.145E-01 -5.145E-01
 15T-UDWA 1.277E-15 1.956E-15 1.726E-15 1.948E-15 1.154E-15 1.791E-15 -1.093E-12 1.729E-15 1.810E-15

16T-UMCC 0. 0. 0. 0. -0. -0. -0. -0. -0. -0.
 17CCMKNC 5.436E-09 6.686E-09 4.248E-08 -1.783E-07 5.842E-09 -3.247E-09 1.048E-05 -4.129E-10 -3.202E-10
 18 TRYCK 1.429E-08 1.641E-08 2.895E-08 1.188E-08 -2.671E-08 5.526E-09 -1.468E-04 2.321E-08 2.859E-08

PRCJECTICNS CF UNIT IMPULSE IN REACTIVITY

1 1.107E 07 2 1.256E 07 3 2.525E 11 4-1.387E 17 5-1.885E 17 6-1.284E 24 7-1.736E 29 8-1.584E 36 9-1.515E 35
 10-8.594E 36 11 6.525E 36 12 9.257E 33 13-2.294E 35 14-4.204E 34 15-4.446E 36 16-3.714E 26 17 1.467E 33 18 1.962E 34

Appendix 4. Sato, " On a decomposition technique for
Multilevel systems".

conditions and t . Therefore (6) is valid. The system studied (Fig. 1) is a classical extremum control, and (5) has previously appeared in the literature.⁴

To determine the system behavior during a short interval of time, the nonlinear part can be replaced by a gain $2\alpha x$,⁵ that is, an incremental model can be used. The system differential equation becomes

$$\tau \frac{d^2c}{dt^2} + \frac{dc}{dt} = \epsilon K_1 K_2 2\alpha x \quad (21)$$

where $\alpha > 0$ and $\epsilon = \epsilon_f - \epsilon_i$. This gives

$$C(s) = \frac{2\epsilon K_1 K_2 \alpha x}{s^2(\tau s + 1)} + \frac{c(0^+)[\tau s + 1]}{s(\tau s + 1)} + \frac{c'(0^+)}{s(\tau s + 1)} \quad (22)$$

where $C(s)$ is the Laplace transform of $c(t)$, and $c(0^+)$ and $c'(0^+)$ are the initial conditions. Noting that $c(0^+) = c(0^-)$ and $c'(0^+) = c'(0^-)$, and using the initial value theorem, it is seen that (16) is obtained. For computer results showing d^2c/dt^2 and d^2c/dx^2 see Perret and Rouxel⁴ and Hamza.⁵ The method of control proposed has been extended to the case where the system has a variable pure delay.⁶

MOHAMED H. HAMZA
University of Zurich
Zurich, Switzerland

⁴ R. Perret and R. Rouxel, "Principle and application of an extremal computer," *Proc. 2nd Internat'l Fed. Automatic Control Cong.* (Basle, Switzerland, 1963), London: Butterworths, 1963.

⁵ M. H. Hamza, "Synthesis of extremum-seeking regulators," *Automation and Remote Control*, vol. 25, pt. 8, 1964.

⁶ M. H. Hamza, "Extremum control in the presence of variable pure delay," in *Theory of Self-Adaptive Control Systems*. New York: Plenum Press, 1966.

On a Decomposition Technique for Multilevel Control Systems

The multilevel control approach [1]-[4], which seems to be prominent in optimization problems of large-scale systems, proceeds along the following stages:

1) *Decomposition*: The system is partitioned into a number of subsystems.

2) *Local optimization*: The infimal control units optimize their own subsystems locally.

3) *Coordination*: The supremal control units adjust the parameters in their infimal units to achieve overall optimality.

In this approach, the difficulty in carrying out the stages 2) and 3) depends on how the decomposition 1) is made. Moreover, as the stages 2) and 3) are usually conducted through an iteration algorithm [4], a certain decomposition may result in the situation where the iteration would not converge to its optimal value [1]. Hence, it is significant to choose carefully the suitable scheme of decomposition before proceeding to the following stages 2) and 3).

In this correspondence, a method is described for decomposing a linear dynamic system

$$\dot{x} = Ax + u \quad (1)$$

with a reducible coefficient matrix A and a separable objective function¹ into the subsystems, utilizing the normal form of the reducible matrix A , so that the computations of 2) and 3) are simplified.

COEFFICIENT MATRIX REDUCTION

It is well known that for an $n \times n$ reducible matrix A , there exists an $n \times n$ permutation matrix P , such that

$$PAP^T = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ & A_{22} & \cdots & A_{2k} \\ & & \ddots & \vdots \\ & & & A_{kk} \\ 0 & & & \end{bmatrix} \quad (2)$$

where each square submatrix A_{ii} ($1 \leq i \leq k$) is either irreducible or a 1×1 null matrix. This block triangular form is called the *normal form* of a reducible matrix A [5].

For obtaining the normal form of a reducible matrix, it is not necessary to find matrix P ; instead the method of using a *reachability matrix* [6], [7] can be employed. Thus, a simple matrix multiplication, which can be easily implemented on a computer, reduces a coefficient matrix A of (1) to its normal form. For example, system

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} -2 & 0 & 0 & 0 & 2 \\ 6 & -4 & 2 & 1 & -3 \\ 3 & 0 & -1 & 0 & 0 \\ 0 & -3 & 2 & 0 & 5 \\ -2 & 0 & 4 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} \quad (3)$$

is reduced to two subsystems such as

$$\frac{d}{dt} \begin{bmatrix} x_2 \\ x_4 \\ x_1 \\ x_3 \\ x_5 \end{bmatrix} = \begin{bmatrix} -4 & 1 & 6 & 2 & -3 \\ -3 & 0 & 0 & 2 & 5 \\ 0 & 3 & -1 & 0 & 0 \\ -2 & 4 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \\ x_1 \\ x_3 \\ x_5 \end{bmatrix} + \begin{bmatrix} u_2 \\ u_4 \\ u_1 \\ u_3 \\ u_5 \end{bmatrix} \quad (4)$$

where $\{x_2, x_4\}$ and $\{x_1, x_3, x_5\}$ are the state variables of each subsystem.

According to the theory of graphs, an irreducible submatrix is equivalent to a *strongly connected subgraph* [5]. For instance, in the case of (4), it is obvious that $\{x_2, x_4\}$ and $\{x_1, x_3, x_5\}$ belong to strongly connected subgraphs, respectively, in the

¹ The objective function is called "separable" in terms of control u , when the decomposed objective function of any subsystem does not contain controls which appear in the equations of other subsystems.

directed graph of the coefficient matrix of (3). So reducing the coefficient matrix of (1) to its normal form means decomposing the linear system (1) into several subsystems in which state vectors are strongly coupled with each other.

The above technique of matrix reduction was first introduced by Harary [6] to find eigenvalues of large sparse matrices and later applied by Himmelblau [7] and Steward [8] to solve large systems of algebraic equations. However, no work has been reported on the technique in connection with the problem of large-scale system optimization.

LINEAR DYNAMIC SYSTEMS WITH QUADRATIC OBJECTIVE FUNCTIONS

The system considered here is (1) with the initial condition

$$x(t_0) = x_0 \quad (5)$$

and the objective function

$$J = \frac{1}{2} \int_{t_0}^{t_1} \{ (x - r)^T B (x - r) + u^T C u \} dt \quad (6)$$

where r is a reference vector, B is a symmetric matrix, and C is a diagonal matrix.² The problem is to find control u to minimize J .

1) Decomposition

If A is reducible, then (1) can be reduced to k subsystems as

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1k} \\ & A_{22} & \cdots & A_{2k} \\ & & \ddots & \vdots \\ & & & A_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{bmatrix} \quad (7)$$

For the i th subsystem ($1 \leq i \leq k$), the equation is

$$\dot{x}_i = A_{ii}x_i + u_i + x_i^* \quad (8)$$

and the objective function is

$$J_i = \frac{1}{2} \int_{t_0}^{t_1} \{ (x_i - r_i^*)^T B_{ii} (x_i - r_i^*) + u_i^T C_{ii} u_i \} dt \quad (9)$$

where x_i^* and r_i^* are *intervention parameters* [4] and are given by

$$x_i^* = \sum_{j \neq i}^k A_{ij} x_j \quad (10)$$

$$r_i^* = r_i + B_{ii}^{-1} \sum_{j \neq i}^k \{ (A^T)_{ij} C_{jj} u_j - B_{ij} (x_j - r_j) \} \quad (11)$$

2) Local Optimization

The local optimization problem is generally to find $u_i(x_i^*, r_i^*)$ minimizing J_i ($1 \leq i \leq k$). In this case, since (1) is reduced to (7), one can see in (10)

$$A_{ij} = 0 \quad \text{for } i > j. \quad (12)$$

² This assures the "separable" condition of the objective function in terms of control u .

Utilizing relations (10) and (12), one can compute controls u_i ($1 \leq i \leq k$) in turn starting from the k th subsystem, as the functions of parameters r_j^* ($i \leq j \leq k$) only.

3) Coordination

The coordination problem is to find r_i^* which assures that u_i (r_i^*) for all i ($1 \leq i \leq k$) minimize the overall objective function J under the constraint of (1). Since (1) is reduced to (7), one can see in (11)

$$(A^T)_{ij} = 0 \quad \text{for } i > j. \quad (13)$$

When (13) is used in (11), computation of the term u_i in (11) becomes simplified.³ The actual computation is implemented as the iteration algorithm using past interaction data [4].

The above decomposition technique may be extended to nonlinear systems when one considers its structural matrix [8] in place of a coefficient matrix.

T. SATO

A. ICHIKAWA

Dept. of Control Engrg.

Tokyo Inst. of Tech.

Meguroku, Tokyo, Japan

REFERENCES

- [1] C. F. Sprague, "On the reticulation problem in multivariable control systems," *IEEE Trans. Automatic Control*, vol. AC-9, pp. 398-405, October 1964.
- [2] Y. Takahara, "Multilevel approach to dynamic optimization," Case Inst. of Tech., Cleveland, Ohio, Rept. SRC 59-A-64-21, 1964.
- [3] M. D. Mesarovic, "Conceptual framework for the study of mng systems," Case Inst. of Tech., Cleveland, Ohio, Rept. SRC 77-A-65-29, 1965.
- [4] M. D. Mesarovic et al., "On the synthesis of dynamic multilevel systems," presented at the 3rd Internat'l Fed. Automatic Control Cong. (London, 1966), paper 40.B.
- [5] R. S. Varga, *Matrix Iterative Analysis*. Englewood Cliffs, N. J.: Prentice-Hall, 1962.
- [6] F. Harary, "A graph theoretic method for the complete reduction of a matrix with a view toward finding its eigenvalues," *J. Math. Phys.*, vol. 38, pp. 104-111, 1959.
- [7] D. M. Himmelblau, "Decomposition of large-scale systems I," *Chem. Engrg. Sci.*, vol. 21, pp. 425-438, May 1966.
- [8] D. V. Steward, "Partitioning and tearing systems of equations," *J. SIAM Num. Anal.*, vol. 2, pp. 345-365, 1965.

³ In particular, if the matrix B is diagonal, the term $B_{ij}(x_j - r_j)$ in (11) vanishes and the coordination becomes simpler.

Functional Reproducibility and the Existence of Classical Sensitivity Matrices

In 1964 Cruz and Perkins proposed a sensitivity matrix for multivariable feedback system design.¹ The following year, Brockett and Mesarovic introduced the concept of reproducibility.² The purpose of

Manuscript received March 7, 1967. This work was supported by NSF Grant GK 804.

¹ J. B. Cruz, Jr., and W. R. Perkins, "A new approach to the sensitivity problem in multivariable feedback system design," *IEEE Trans. Automatic Control*, vol. AC-9, pp. 216-223, July 1964.

² R. W. Brockett and M. D. Mesarovic, "The reproducibility of multivariable systems," *J. Math. Anal. and Appl.*, vol. 11, pp. 548-563, July 1965.

this correspondence is to point out that certain of the properties developed by Brockett and Mesarovic appear to be intrinsically related to the existence of the Cruz-Perkins sensitivity matrix.

In the sequel, it is convenient to assume that the plant is a linear, time-invariant dynamical system. Let

$$P(s) = C(sI - A)^{-1}B \quad (1)$$

be the zero-state representation of the triple $\{A, B, C\}$, where

$$\dot{x} = Ax + Bu \quad (2a)$$

$$y = Cx. \quad (2b)$$

The matrices A , B , and C have dimensions $n \times n$, $n \times m$, and $r \times n$, respectively.

In the Cruz-Perkins development, it is assumed that

$$P(s)U(s) = 0 \quad (3)$$

if and only if

$$U(s) = 0 \quad (4)$$

where $U(s)$ is an m vector of input transforms. In the Brockett-Mesarovic development, the homogeneous responses of (2) are said to be functionally reproducible if and only if the relation

$$P^T(s)K(s) = 0 \quad (5)$$

implies that

$$K(s) = 0 \quad (6)$$

where $K(s)$ is an r vector and superscript T denotes transposition. Clearly, $P^T(s)$ may be viewed as the zero-state representation of the triple $\{A^T, C^T, B^T\}$. Accordingly, it is sufficient to state the following theorem.

THEOREM

The linear, time-invariant dynamical system characterized by the triple $\{A, B, C\}$ has a classical sensitivity matrix in the sense of Cruz and Perkins if and only if the corresponding system $\{A^T, C^T, B^T\}$ has functionally reproducible homogeneous responses in the sense of Brockett and Mesarovic.

Algebraic criteria for the above theorem are stated in Brockett and Mesarovic.³ It is of interest to observe, however, that a necessary condition is $r \geq m$ or, equivalently, that the number of controlled inputs is no greater than the number of outputs. If $r = m$, the above properties become equivalent to the existence of $P^{-1}(s)$, which has been studied by Youla and Dorato.³

ACKNOWLEDGMENT

The author wishes to thank Profs. J. B. Cruz, Jr., and W. R. Perkins of the University of Illinois, Urbana, for helpful comments related to this discussion.

MICHAEL K. SAIN

Dept. of Elec. Engrg.

University of Notre Dame

Notre Dame, Ind.

³ D. C. Youla and P. Dorato, "On the inverse of linear dynamical systems," Polytechnic Institute of Brooklyn, N. Y., Electrophysics Memo. PTBMR-1319-66, March 1966.

A General Proof of the P_n Matrix and Its Inverse

INTRODUCTION

In a previous short paper [1] we transform a set of phase-variable differential equations

$$\dot{Y} = BY$$

where

$$B = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \cdot & 1 \\ -b_1 & -b_2 & -b_3 & \cdots & \cdot & -b_n \end{bmatrix} \quad (1)$$

into the corresponding set of Schwarz's variable differential equations

$$\dot{X} = AX$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ -a_1 & 0 & 1 & \cdots & 0 & 0 \\ 0 & -a_2 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & 0 & -a_{n-1} & -a_n \end{bmatrix} \quad (2)$$

by the following transformation P_n :

$$P_n = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ C_{(n-1)2} & \cdot & \cdot & 0 & 0 & 0 \\ C_{(n-1)1} & \cdot & \cdot & 0 & 0 & 0 \\ 0 & \cdot & 0 & 1 & 0 & 0 \\ C_{(n-3)3} & \cdot & C_{32} & 0 & 1 & 0 \\ C_{(n-3)1} & \cdot & C_{31} & 0 & 1 & 0 \\ 0 & \cdot & 0 & C_{22} & 0 & 1 \\ & & & C_{21} & & \end{bmatrix} \quad (3)$$

such that $A = P_n B P_n^{-1}$. The coefficients C_{ij} in P_n are the elements in the Routh array.

Naturally, two questions arise: 1) Is there a simple formulation for P^{-1} also? 2) Is there a general proof for P_n ? This paper will answer these two questions.

FORMULATION OF THE INVERSE MATRIX

First, some rules for formulating the matrix P_n^{-1} are given. The rules are comparable with those given for P_n .

1) P_2^{-1} , P_3^{-1} , P_4^{-1} , \dots , P_{n-1}^{-1} are principal minors of P_n^{-1} from the lower right corner to the upper left corner of the main diagonal.

2) The P^{-1} matrix is always a triangular matrix such that the elements above the main diagonal are zero.

3) Each element of the main diagonal is unity.

4) Each element of the first lower diagonal is zero.

5) Each element of the second lower diagonal is equal to the corresponding element of the second diagonal of P after changing sign. That means, from the lower right corner to the upper left corner, the elements of the second lower diagonal of P^{-1} are:

Manuscript received March 28, 1967. This research was supported in part by NSF Grant GP-6167.