UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA DE CAMPINAS

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ERRO ⇒ canonicas

ANÁLISE E COMPUTAÇÃO DE FORMAS CANÔNICAS PARA SISTEMAS LINEARES MULTIVARIÁVEIS

ARLINDO MOREIRA FARTES FILHO

Orientador - CELSO PASCOLI BOTTURA

Tese de Mestrado apresentada à Faculdade de Engenharia da Universidade Estadual de Campinas

DEZEMBRO - 1977

SUMÁRIO

Neste trabalho um conjunto de formas canônicas relacionadas com o aspecto controlabilidade-observabilidade é apresentado, sendo definidos alguns índices e observadas algumas características intrínsecas das formas.

São propostas formas canônicas alternativas; uma visando contornar determinada dificuldade que surge em uma das formas de Luenberger, outra mantendo a mesma característica quanro à invariância que existe nas formas de Bingulac, e a última que computacionalmente é origem de formas de Luenberger e de Bingulac.

São comparados métodos de obter as diversas formas canônicas ressaltando a versatilidade do método das transformações de similaridade elementares.

Finalmente, como uso desta versatilidade, é aperfeiçoado um método de se obter função de transferência à partir da descrição do sistema em espaço de estado.

ÍNDICE

CAPÍTULO I

INTRODUÇÃO

As formas canônicas desenvolvidas por LUENBERGER, /1/, como uma extensão das formas canônicas controláveis e observáveis de sistemas monovariáveis, e suas derivações obtidas por BINGULAC, /5/, tem encontrado uma grande aplicação na teoria de controle.

Podemos situar que tais formas tem sido usadas no projeto de observadores, no estudo de sistemas multivariáveis, nos problemas de alocação de polos e de desacoplamento, na simplificação de problemas de controle linear quadrático ótimo, na identificação de sistemas, etc.

Como objetivos deste trabalho teremos a análise dos parâmetros das formas canônicas e a computação das mesmas. Para isso no segundo capítulo serão expostas as formas canônicas de LUENBERGER já com alguma evolução em relação ao seu trabalho original tal como classificar quatro formas canônicas, todas de característica diferentes mas bem definidas. E também no segundo capítulo serão apresentadas as formas canônicas alternativas de BINGULAC sendo que para uma delas, que será apresentada como FCCB2 , a apresentação será mais versátil que no seu original.

No capítulo terceiro serão comparados métodos para obter formas canônicas, sendo estes os métodos; tradicional, de JORDAN-SRIDHAR, /2/, e o de DALY, /3/. Neste capítulo toda ênfase será para o aspecto computacional o que notar-se-á pelos diversos algorítmos apresentados.

No capítulo seguinte, o quarto, é demonstrada a existência das formas canônicas de uma maneira computacional, uma vez que LUENBERGER já demonstrou em seu trabalho que as matrizes transformação de similaridade são inversíveis. No entanto as demonstrações procuram ser mais versáteis pois provar a não singularidade das matrizes transformação de similaridade necessariamente nada diz a respeito das formas canônicas. Neste quarto capítulo também serão demonstrados quais os parâmetros das formas canônicas de LUENBERGER e de BINGULAC são sempre nulos e que entre algumas formas o número de elementos obrigatoriamente nulos é invariante. Outro aspecto de invariância que existe em uma das formas de BINGULAC é reproduzido por uma forma proposta, a FCCS2 . Em relação a problemas do tipo alocação de polos ou estimação de estado, uma das formas de LUENBERGER se mostra inadequada, mas com uma variação desta, que será chamada de FCCS1 , contornaremos este problema. Prosseguindo mostraremos mais uma forma, FCCS3, que computacionalmente é origem de outras formas canônicas. Finalmente apresentaremos um uso imediato de FCCS1 ao melhorarmos o método proposto por DALY para obter a função de transferência de um sistema a partir de sua descrição em espaço de estado.

Para encerrar o trabalho teremos no capítulo quinto uma conclusão sobre o que foi desenvolvido.

NOTAÇÃO E NOÇÕES BÁSICAS

Tendo em vista o objetivo deste trabalho, de implementar algorítmos, será usada uma notação bem próxima da linguagem FORTRAN, quando em algorítmos, na seguinte forma:

NOME (.,.) - matriz NOME.

NOME (I,.) - linha I da matriz NOME.

NOME (.,J) - coluna J da matriz NOME.

NOME (I,J) - elemento da linha I e coluna J da matriz NOME.

**,*,/,+,- - potenciação, multiplicação, divisão, adição e subtração, respectivamente.

Por outro lado em algumas descrições analíticas e em algumas explicações será usada a seguinte notação:

MATRIZ - letras maiúsculas

VETOR - letras minúsculas sublinhadas

ELEMENTO - letras minúsculas

$\underline{nome}^{i}$ - linha I da matriz NOME.

$\underline{nome}_{j}$ - coluna J da matriz NOME.

$nome_{i,j}$ - elemento da linha I e coluna J da matriz NOME.

$NOME_{I,J}$ - sub-matriz I,J da matriz NOME.

Também como auxílio a nossa notação usaremos a seguinte equivalência:

$NOMEIN = (NOME)^{-1}$

$IDN = \text{matriz } I_n$

NOME [A,B] = matriz NOME como função das matrizes A e B.

Consideremos agora que um sistema linear seja descrito por:

$$\begin{cases} \dot{\underline{x}} = A\underline{x} + B\underline{u} \\ \underline{y} = C\underline{x} + D\underline{u} \end{cases} \tag{1}$$

onde $\underline{x}$ é um vetor coluna n-dimensional das variáveis de estado, $\underline{u}$ é um vetor coluna m-dimensional das variáveis de entrada, $\underline{y}$ é um vetor coluna L-dimensional das variáveis de saída e as matrizes A, B, C e D são de dimensões apropriadas.

Se fizermos agora uma mudança de base no espaço de estado dada por:

$$\underline{z} = T\underline{x} \tag{2}$$

teremos:

$$\begin{cases} \dot{\underline{z}} = TAT^{-1}\underline{z} + TB\underline{u} \\ \underline{y} = CT^{-1}\underline{z} + D\underline{u} \end{cases} \tag{3}$$

Podemos então representar o sistema (1) e a influência da transformação de similaridade T pelas seguintes equações matriciais:

$$\left[\begin{array}{c} \dot{\underline{x}} \\ \hline \underline{y} \\ \hline \underline{x} \end{array}\right] = \left[\begin{array}{c|c|c} A & B & I_n \\ \hline C & D & 0 \\ \hline I_n & 0 & 0 \end{array}\right] * \left[\begin{array}{c} \underline{x} \\ \hline \underline{u} \\ \hline 0 \end{array}\right] = H * \left[\begin{array}{c} \underline{x} \\ \hline \underline{u} \\ \hline 0 \end{array}\right] \tag{4}$$

$$\begin{bmatrix} \dot{\underline{z}} \\ \hline \underline{y} \\ \hline \underline{x} \end{bmatrix} = \left[\begin{array}{c:c:c} TAT^{-1} & TB & T \\ \hline CT^{-1} & D & 0 \\ \hline T^{-1} & 0 & 0 \end{array}\right] * \begin{bmatrix} \underline{z} \\ \hline \underline{u} \\ \hline 0 \end{bmatrix} \tag{5}$$

onde o efeito de T fica descrito por:

$$\left[\begin{array}{c:c:c} T & 0 & 0 \\ \hline 0 & I_n & 0 \\ \hline 0 & 0 & I_n \end{array}\right] * \left[\begin{array}{c:c:c} A & B & I_n \\ \hline C & D & 0 \\ \hline I_n & 0 & 0 \end{array}\right] * \left[\begin{array}{c:c:c} T^{-1} & 0 & 0 \\ \hline 0 & I_L & 0 \\ \hline 0 & 0 & I_n \end{array}\right] = \left[\begin{array}{c:c:c} TAT^{-1} & TB & T \\ \hline CT^{-1} & D & 0 \\ \hline T^{-1} & 0 & 0 \end{array}\right] \tag{6}$$

O mesmo sistema (1) será dito controlável se a matriz de controlabilidade:

$$R[A,B] = [B \,\vdots\, A*B \,\vdots\, \dots \,\vdots\, A^{n-1}*B] \tag{7}$$

tiver n colunas linearmente independentes.

Da mesma forma o sistema (1) será dito observável se a matriz de observabilidade:

$$S[A,C] = \left[C^T \,\vdots\, (C*A)^T \,\vdots\, \dots \,\vdots\, (C*A^{n-1})^T\right]^T \tag{8}$$

tiver n linhas linearmente independentes.

De acordo com (1) e (3) podemos notar que:

$$R[TAT^{-1},\ TB] = T* R[A,B] \tag{9}$$

$$S[TAT^{-1},\ CT^{-1}] = S[A,C]*T^{-1} \tag{10}$$

e desde que T não seja singular será mantida a mesma dependência entre as colunas de $R[TAT^{-1}, TB]$ e $R[A,B]$ assim como será mantida a mesma dependência entre as linhas de $S[TAT^{-1}, CT^{-1}]$ e $S[A,C]$.

Durante o desenvolvimento do trabalho será suposto que não há redundância de entradas ou de saídas, isto é, as colunas de B são linearmente independentes e da mesma forma o são as linhas de C. Por outro lado este fato será relaxado e posteriormente detectado nos algorítmos que forem realmente usados.

Por fim vamos destacar que é semelhante dizer controlabilidade de $[A,B]$ ou observabilidade de $[A^T, B^T]$, ou que é semelhante dizer observabilidade de $[A,C]$ ou controlabilidade de $[A^T, C^T]$. Por isso o nosso tratamento será feito em somente um dos casos, já que o outro caso mediante à dualidade recai no caso então estudado.

CAPÍTULO II

Neste capítulo introduziremos alguns algorítmos básicos e outros específicos para obtermos as formas canônicas de Luenberger, também serão definidos índices de controlabilidade e de observabilidade de modo a distinguirmos estes diversos índices uma vez que na bibliografia utilizada notamos que tais índices, por uma falta de definição de uso comum, foram usados diversas vezes com significados diferentes.

Para obtermos as formas canônicas de Luenberger para o sistema:

$$\dot{\underline{x}} = A\underline{x} + B\underline{u}$$

$$\underline{y} = C\underline{x} + D\underline{u}$$

vamos construir primeiramente as matrizes de observabilidade e de controlabilidade que podem ser geradas pelos seguintes algorítmos:

<u>ALGORÍTMO 1</u>: Matriz de controlabilidade RAB[A,B].

1.- RAB(.,.) = B(.,.) ; K=1

2.- RAB(.,.) = [B(.,.) ¦ A(.,.)*RAB(.,.)] ; K=K+1

3.- se K<N vá para 2.- ; se não pare

<u>ALGORÍTMO 2</u>: Matriz de observabilidade SAC[A,C].

1.- SAC(.,.) = C(.,.) ; K=1

2.- $\text{SAC}(.,.) = \left[\begin{array}{c} C(.,.) \\ \hline \text{SAC}(.,.)*A(.,.) \end{array}\right]$; K=K+1

3.- se K<N vá para 2.- ; se não pare

Se as colunas de B forem linearmente independentes assim como as linhas de C poderemos substituir o ítem 3.- do algorítmo 1 por K=N-M e do algorítmo 2 por K=N-L se a finalidade for obter RAB, ou SAC, até que o posto de cada uma destas matrizes seja N.

Para dar prosseguimento a este trabalho vamos definir alguns índices tais como:

DEFINIÇÃO 1: Índice de Controlabilidade IC.

Chamaremos de índice de controlabilidade, IC, ao menor valor de K no algorítmo 1 com o qual conseguimos obter N colunas linearmente independentes em RAB.

DEFINIÇÃO 2: Índice de Observabilidade IO.

Chamaremos de índice de observabilidade, IO, ao menor valor de K no algorítmo 2 com o qual conseguimos obter N linhas linearmente independentes em SAC.

Devemos agora definir as matrizes transformação de similaridade que operadas no sistema original nos levem às formas canônicas procuradas.

Estas matrizes geradas a seguir tem como primeira letra C (controlável) ou O (observável) de acordo com o tipo de forma que visa formar. Como segunda letra um indicativo do autor, no caso L (Luenberger) e a seguir um número para diferenciar as matrizes do mesmo autor e com o mesmo objetivo, mas diferentes; portanto:

ALGORÍTMO 3: CL1[RAB,B]

.- Agrupando na forma $\underline{b}_1$, $A\underline{b}_1$, ..., $\underline{b}_2$, ...

1.- K=0 ; DO 3.- I=1,M

2.- DO 3.- J=0,N-1

3.- K=K+1 ; CL1(.,K) = RAB(.,I+J*M)

.- Obtendo CL1 na forma final

4.- selecione as primeiras N colunas linearmente independentes de CL1(.,.).

ALGORĪTMO 4: CL2[RAB,B]

1.- selecione as N primeiras colunas linearmente independentes de RAB(.,.).

2.- rearrume estas colunas de forma a agrupar todas as colunas derivadas de uma mesma coluna de B na ordem em que forem encontradas, isto é, $\underline{b}_1$, $A\underline{b}_1$,...., $\underline{b}_2$... .

ALGORĪTMO 5: OL1[SAC,C]

.- Agrupando na forma $[(\underline{c}^1)^T, (\underline{c}^1 A)^T, \ldots, (\underline{c}^2)^T, \ldots]^T$

1.- K=0 ; DO 3.- I=1,L

2.- DO 3.- J=0,N-1

3.- K=K+1 ; OL1(K,.) = SAC(I+J*L,.)

.- Obtendo a forma final de OL1

4.- selecione as N primeiras linhas linearmente independentes de OL1(.,.).

ALGORĪTMO 6: OL2[SAC,C]

1.- selecione as primeiras N linhas linearmente independentes da matriz de observabilidade SAC(.,.).

2.- rearrume estas linhas de forma a agrupar todas as linhas derivadas de uma mesma linha de C, na ordem em que forem encontradas, isto é, $[(\underline{c}^1)^T, (\underline{c}^1 A)^T, \ldots, (\underline{c}^2)^T, \ldots]^T$.

Com estas matrizes formadas vamos definir alguns índices que serão usadas no nosso trabalho, são eles:

DEFINIÇÃO 3: Índice Relativo de Controlabilidade (IRC).

Chamaremos de índice relativo de controlabilidade da entrada I, IRC(I), ao número de vezes que a coluna B(.,I) aparece em CL1 ou CL2 conforme a matriz usada.

DEFINIÇÃO 4: Índice Relativo de Observabilidade (IRO).

Chamaremos de índice relativo de observabilidade da saída I, IRO(I), ao número de vezes que a linha C(I,.) aparece em OL1 e OL2 conforme a matriz que estiver em uso.

Com as matrizes de transformação de similaridade obtidas e com os índices definidos acima estamos em condição de obtermos algumas formas canônicas de Luenberger.

Com a finalidade de diferenciarmos as diversas formas canônicas usaremos a seguinte convensão, por exemplo em FCCL1, FC de forma canônica, o segundo C de controlável, L de Luenberger e o número para diferenciarmos por exemplo de FCCL4.

FORMA CANÔNICA CONTROLÁVEL FCCL1

$$\left[\begin{array}{c|c|c} \mathrm{CL1IN} & 0 & 0 \\ \hline 0 & \mathrm{IDL} & 0 \\ \hline 0 & 0 & \mathrm{IDN} \end{array}\right] * H * \left[\begin{array}{c|c|c} \mathrm{CL1} & 0 & 0 \\ \hline 0 & \mathrm{IDM} & 0 \\ \hline 0 & 0 & \mathrm{IDN} \end{array}\right] = \left[\begin{array}{c|c|c} \mathrm{ALC1} & \mathrm{BLC1} & \mathrm{TLC1} \\ \hline \mathrm{CLC1} & D & 0 \\ \hline \mathrm{TLC1IN} & 0 & 0 \end{array}\right] = \mathrm{HLC1}$$

FORMA CANÔNICA CONTROLÁVEL FCCL2

$$\left[\begin{array}{c|c|c} \mathrm{CL2IN} & 0 & 0 \\ \hline 0 & \mathrm{IDL} & 0 \\ \hline 0 & 0 & \mathrm{IDN} \end{array}\right] * H * \left[\begin{array}{c|c|c} \mathrm{CL2} & 0 & 0 \\ \hline 0 & \mathrm{IDM} & 0 \\ \hline 0 & 0 & \mathrm{IDN} \end{array}\right] = \left[\begin{array}{c|c|c} \mathrm{ALC2} & \mathrm{BLC2} & \mathrm{TLC2} \\ \hline \mathrm{CLC2} & D & 0 \\ \hline \mathrm{TLC2IN} & 0 & 0 \end{array}\right] = \mathrm{HLC2}$$

FORMA CANÔNICA OBSERVÁVEL FCOL1

$$\left[\begin{array}{c:c:c} OL1 & 0 & 0 \\ \hdashline 0 & IDL & 0 \\ \hdashline 0 & 0 & IDN \end{array}\right] *H* \left[\begin{array}{c:c:c} OL1IN & 0 & 0 \\ \hdashline 0 & IDM & 0 \\ \hdashline 0 & 0 & IDN \end{array}\right] = \left[\begin{array}{c:c:c} AL01 & BL01 & TL01 \\ \hdashline CL01 & D & 0 \\ \hdashline TL01IN & 0 & 0 \end{array}\right] = HL01$$

FORMA CANÔNICA OBSERVÁVEL FCOL2

$$\left[\begin{array}{c:c:c} OL2 & 0 & 0 \\ \hdashline 0 & IDL & 0 \\ \hdashline 0 & 0 & IDN \end{array}\right] *H* \left[\begin{array}{c:c:c} OL2IN & 0 & 0 \\ \hdashline 0 & IDM & 0 \\ \hdashline 0 & 0 & IDN \end{array}\right] = \left[\begin{array}{c:c:c} AL02 & BL02 & TL02 \\ \hdashline CL02 & D & 0 \\ \hdashline TL02IN & 0 & 0 \end{array}\right] = HL02$$

Nestas formas canônicas obteremos a seguintes configurações para as duplas

$[ALC2]$ $[BLC2]$ e $[AL02]^T$ $[CL02]^T$:

$$\left[\begin{array}{c:c:c:c}
00 \ldots 0X & 00 \ldots 0X & \ldots & 00 \ldots 0X \\
10 \ldots 0X & 00 \ldots 0X & \ldots & 00 \ldots 0X \\
01 \ldots 0X & 00 \ldots 0X & \ldots & 00 \ldots 0X \\
\cdots & \cdots & \cdots & \cdots \\
00 \ldots 1X & 00 \ldots 0X & \ldots & 00 \ldots 0X \\
\hdashline
00 \ldots 0X & 00 \ldots 0X & \ldots & 00 \ldots 0X \\
00 \ldots 0X & 10 \ldots 0x & \ldots & 00 \ldots 0X \\
00 \ldots 0X & 01 \ldots 0X & \ldots & 00 \ldots 0X \\
\cdots & \cdots & \cdots & \cdots \\
00 \ldots 0X & 00 \ldots 1X & \ldots & 00 \ldots 0X \\
\hdashline
\cdots & \cdots & \cdots & \cdots \\
\hdashline
00 \ldots 0X & 00 \ldots 0X & \ldots & 00 \ldots 0X \\
00 \ldots 0X & 00 \ldots 0X & \ldots & 10 \ldots 0X \\
00 \ldots 0X & 00 \ldots 0X & \ldots & 01 \ldots 0X \\
\cdots & \cdots & \cdots & \cdots \\
00 \ldots 0X & 00 \ldots 0X & \ldots & 00 \ldots 1X
\end{array}\right]
\qquad
\left[\begin{array}{c}
10 \ldots 0 \\
00 \ldots 0 \\
00 \ldots 0 \\
\cdots \\
00 \ldots 0 \\
\hdashline
01 \ldots 0 \\
00 \ldots 0 \\
00 \ldots 0 \\
\cdots \\
00 \ldots 0 \\
\hdashline
\cdots \\
\hdashline
00 \ldots 1 \\
00 \ldots 0 \\
00 \ldots 0 \\
\cdots \\
00 \ldots 0
\end{array}\right]$$

e para as duplas $[ALC1]$ $[BLC1]$ e $[AL01^T]$ $[CL01^T]$ teremos

$$
\left[\begin{array}{c:c:c:c}
00 \ldots 0X & 00 \ldots 0X & \ldots & 00 \ldots 0X \\
10 \ldots 0X & 00 \ldots 0X & \ldots & 00 \ldots 0X \\
01 \ldots 0X & 00 \ldots 0X & \ldots & 00 \ldots 0X \\
\cdots & \cdots & \cdots & \cdots \\
00 \ldots 1X & 00 \ldots 0X & \ldots & 00 \ldots 0X \\
\hdashline
00 \ldots 00 & 00 \ldots 0X & \ldots & 00 \ldots 0X \\
00 \ldots 00 & 10 \ldots 0X & \ldots & 00 \ldots 0X \\
00 \ldots 00 & 01 \ldots 0X & \ldots & 00 \ldots 0X \\
\cdots & \cdots & \cdots & \cdots \\
00 \ldots 00 & 00 \ldots 1X & \ldots & 00 \ldots 0X \\
\hdashline
\cdots & \cdots & \cdots & \cdots \\
\hdashline
00 \ldots 00 & 00 \ldots 00 & \ldots & 00 \ldots 0X \\
00 \ldots 00 & 00 \ldots 00 & \ldots & 10 \ldots 0X \\
00 \ldots 00 & 00 \ldots 00 & \ldots & 01 \ldots 0X \\
\cdots & \cdots & \cdots & \cdots \\
00 \ldots 00 & 00 \ldots 00 & \ldots & 00 \ldots 0X
\end{array}\right]
\qquad
\left[\begin{array}{c:c}
10 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
\cdots & \cdots \\
00 \ldots 0 & X \ldots X \\
\hdashline
01 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
\cdots & \cdots \\
00 \ldots 0 & X \ldots X \\
\hdashline
\cdots & \cdots \\
\hdashline
00 \ldots 1 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
\cdots & \cdots \\
00 \ldots 0 & X \ldots X
\end{array}\right]
$$

como podemos observar algumas colunas de BLC1, ou $CLC1^T$, não possuem particularidade alguma; estas colunas não aparecem necessariamente como as últimas, mas se permutarmos convenientemente as colunas de B ou as linhas de C obteremos esta forma. Para estas formas a existência de tais colunas pode ser compreendida se atentarmos para o fato de que em CL1, ou OL1, eventualmente uma coluna de B, ou linha de C, não fará parte de CL1 ou OL1 respectivamente.

Para auxiliar na descrição dos próximos algorítmos e para relaxarmos a exigência de que todas as colunas de B ou C^T sejam linearmente independentes, K será o número de colunas de B que participa de CL1 ou CL2 e P será o número de linhas de C que participa de OL1 ou OL2.

Traçadas estas considerações podemos analisar mais atentamente a topologia das formas canônicas já descritas e com isso notar que determinadas colunas de ALC1 e ALC2 são ocupadas por elementos 'X' que podem assumir qualquer valor e que os elementos

'1' de BLC1 e BLC2 se localizam em linhas adequadas. Estas linhas estão relacionadas com os índices relativos de controlabilidade da mesma forma que estão as colunas dos elementos 'X' em ALC1 e ALC2, colunas estas que chamaremos de colunas estruturais. Para o caso das formas observáveis o pensamento será sempre transposto, isto é, trocar onde é dito colunas por linhas e vice-versa e interpretar onde é dito controlabilidade por observabilidade dada a característica de dualidade que envolve tais conceitos, razão pela qual daqui para frente só abordaremos as formas controláveis, sendo imediata a transferência de raciocínio para as formas observáveis.

Para definirmos as linhas ou colunas estruturais usaremos mais alguns índices definidos abaixo:

DEFINIÇÃO 5: Índice Estrutural de Controlabilidade das formas canônicas de Luenberger (IECL).

Chamaremos de índice estrutural de controlabilidade de Luenberger relativo a entrada I, IECL(I), ao valor:

IECL(I) = IRC(1) + IRC(2) + ... + IRC(I)

DEFINIÇÃO 6: Índice Estrutural de Observabilidade das formas canônicas de Luenberger (IEOL)

Chamaremos de índice estrutural de observabilidade de Luenberger relativo a saída I, IEOL(I), ao valor:

IEOL(I) = IRO(1) + IRO(2) + ... + IRO(I)

Com as definições 3 e 5 e sabendo que para cada bloco da diagonal principal de ALC1 e ALC2, os blocos são as submatrizes separadas por traços, tem dimensão IRC(I)xIRC(I), podemos concluir que a I-ésima coluna estrutural é a coluna IECL(I) e que as linhas J onde aparecem os elementos '1' de BLC1 ou BLC2 são as linhas IECL(I) - IRC(I)+1, I=1,2,..,K.

Além destas características podemos notar que ALC1 e ALO1 são matrizes triangulares por blocos, isto é, os blocos de um dos lados da diagonal principal são submatrizes nulas e neste caso o determinante $|sI_n\text{-ALC1}|$, que é o polinômio característico do sistema, pode ser expresso como o produto de determinantes $[sI_r - ALC1_{I,I}]$, $r = IRC(I)$ e $I = 1,2, \ldots , K$, indicando assim que o polinômio característico é formado pelo produto de polinômios de grau menor, cujos coeficientes são identificáveis de modo imediato nos elementos 'X' dos blocos da diagonal principal uma vêz que para uma matriz M em qualquer uma das seguintes formas:

$$\begin{bmatrix} X(1) & X(2) & \ldots & X(n-1) & X(n) \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \quad \text{ou}$$

$$\begin{bmatrix} 0 & 0 & \ldots & 0 & X(n) \\ 1 & 0 & \ldots & 0 & X(n-1) \\ 0 & 1 & \ldots & 0 & X(n-2) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \ldots & 0 & X(1) \end{bmatrix} \quad \text{ou}$$

$$\begin{bmatrix} X(1) & 1 & 0 & \ldots & 0 \\ X(2) & 0 & 1 & \ldots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ X(n-1) & 0 & 0 & \ldots & 1 \\ X(n) & 0 & 0 & \ldots & 0 \end{bmatrix} \quad \text{ou}$$

$$\begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \ldots & \\ X(n) & X(n-1) & X(n-2) & \ldots & X(1) \end{bmatrix}$$

o determinante $|sI_n - M| = p(s)$ é dado por:

$$p(s) = s^n - X(1)\, s^{n-1} - X(2)\, s^{n-2} - \ldots - X(n)$$

Por fim vamos observar que as colunas 1,2, ... , K em BLC1 ou BLC2, das linhas que contêm os elementos '1' destas matrizes, formam a matriz IDK.

Quanto às matrizes CLC1, CLC2, BLO1 e BLO2 , estas não assumem característica especial nenhuma.

As formas canônicas apresentadas até aqui são obtidas de transformações de similaridade cujas matrizes são formadas diretamente das colunas da matriz de controlabilidade RAB , como nos casos de FCCL1 e FCCL2 , ou das linhas da matriz de observabilidade de SAC , como nos casos de FCOL1 e FCOL2. No entanto outras formas de grande utilidade empregam transformações de similaridade cuja obtenção é um pouco mais complexa. Para chegar à estas formas vamos obter algumas matrizes auxiliares construídas pelos algorítmos abaixo.

ALGORÍTMO 7: C1

```
1.-   DO   3.-   I=1,K
2.-   J=IECL(I)   ,   IECL(I) definido em CL1
3.-   C1(I,.) = CL1IN(J,.)
```

ALGORÍTMO 8: C1A

```
1.-   DO   3.-   I=1,K
2.-   J=IECL(I)   ,   IECL(I) definido em CL1
3.-   C1A(I,.) = IDN(J,.)
```

ALGORÍTMO 9: C2

```
1.-   DO   3.-   I=1,K
2.-   J=IECL(I)   ,   IECL(I) definido em CL2
3.-   C2(I,.) = CL2IN(J,.)
```

ALGORÍTMO 10: C2A

```
1.-   DO   3.-   I=1,K
```

2.- J=IECL(I) , IECL(I) definido em CL2

3.- C2A(I,.) = IDN(J,.)

Podemos mostrar que as matrizes C1 e C1A assim como C2 e C2A estão relacionadas por:

C1 * CL1 = C1A

C2 * CL2 = C2A

uma vez que

CL1IN*CL1 = IDN

e CL2IN*CL2 = IDN

Vamos agora com estas matrizes auxiliares obter umas matrizes de observabilidade auxiliares das quais extrairemos as matrizes transformação de similaridade auxiliares.

SAC1[A,B] = SAC[A,C1] - ALGORĪTMO 2

SAC1A[A,B] = SAC[A,C1A]- ALGORĪTMO 2

SAC2[A,B] = SAC[A,C2] - ALGORĪTMO 2

SAC2A[A,B] = SAC[A,C2A]- ALGORĪTMO 2

destas matrizes de observabilidade auxiliares extraimos as seguintes matrizes de transformação de similaridade auxiliares.

CL3[A,B] = OL1[SAC1,C1] - ALGORĪTMO 5

CL3A[A,B] = OL1[SAC1A,C1A] - ALGORĪTMO 5

CL4[A,B] = OL2[SAC2,C2] - ALGORĪTMO 6

CL4A[A,B] = OL2[SAC2A,C2A] - ALGORĪTMO 6

as quais nos levam a:

FORMA CANÔNICA CONTROLÁVEL FCCL3

$$\left[\begin{array}{c|c|c} CL3 & 0 & 0 \\ \hline 0 & IDL & 0 \\ \hline 0 & 0 & IDN \end{array}\right] * H * \left[\begin{array}{c|c|c} CL3IN & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right] = \left[\begin{array}{c|c|c} ALC3 & BLC3 & TLC3 \\ \hline CLC3 & D & 0 \\ \hline TLC3IN & 0 & 0 \end{array}\right] =$$

$$= \left[\begin{array}{c|c|c} CL3A & 0 & 0 \\ \hline 0 & IDL & 0 \\ \hline 0 & 0 & IDN \end{array}\right] * HLC1 * \left[\begin{array}{c|c|c} CL3AIN & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right] = HLC3$$

FORMA CANÔNICA CONTROLÁVEL FCCL4

$$\left[\begin{array}{c|c|c} CL4 & 0 & 0 \\ \hline 0 & IDL & 0 \\ \hline 0 & 0 & IDN \end{array}\right] * H * \left[\begin{array}{c|c|c} CL4IN & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right] = \left[\begin{array}{c|c|c} ALC4 & BLC4 & TLC4 \\ \hline CLC4 & D & 0 \\ \hline TLC4IN & 0 & 0 \end{array}\right] =$$

$$= \left[\begin{array}{c|c|c} CL4A & 0 & 0 \\ \hline 0 & IDL & 0 \\ \hline 0 & 0 & IDN \end{array}\right] * HLC2 * \left[\begin{array}{c|c|c} CL4AIN & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right] = HLC4$$

Entre as formas canônicas do primeiro grupo, FCCL2 e FCCL4 , existem relações não só das formas em si mas de características como número de parâmetros que identificam o sistema e índices estruturais.

Para o par [ALC3] [BLC3] temos como forma topológica:

$$
\left[\begin{array}{c:c:c:c}
010 \ldots 0 & 000 \ldots 0 & \ldots & 000 \ldots 0 \\
001 \ldots 0 & 000 \ldots 0 & \ldots & 000 \ldots 0 \\
000 \ldots 0 & 000 \ldots 0 & \ldots & 000 \ldots 0 \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
XXX \ldots X & XXX \ldots X & \ldots & XXX \ldots X \\
\hdashline
000 \ldots 0 & 010 \ldots 0 & \ldots & 000 \ldots 0 \\
000 \ldots 0 & 001 \ldots 0 & \ldots & 000 \ldots 0 \\
000 \ldots 0 & 000 \ldots 0 & \ldots & 000 \ldots 0 \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
000 \ldots 0 & XXX \ldots X & \ldots & XXX \ldots X \\
\hdashline
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
\hdashline
000 \ldots 0 & 000 \ldots 0 & \ldots & 010 \ldots 0 \\
000 \ldots 0 & 000 \ldots 0 & \ldots & 001 \ldots 0 \\
000 \ldots 0 & 000 \ldots 0 & \ldots & 000 \ldots 0 \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
000 \ldots 0 & 000 \ldots 0 & \ldots & XXX \ldots X
\end{array}\right]
\qquad
\left[\begin{array}{c:c}
0Y \ldots Y & X \ldots X \\
0Y \ldots Y & X \ldots X \\
0Y \ldots Y & X \ldots X \\
\cdots\cdots & \cdots\cdots \\
1Z \ldots Z & X \ldots X \\
\hdashline
00 \ldots Y & X \ldots X \\
00 \ldots Y & X \ldots X \\
00 \ldots Y & X \ldots X \\
\cdots\cdots & \cdots\cdots \\
01 \ldots Z & X \ldots X \\
\hdashline
\cdots\cdots & \cdots\cdots \\
\hdashline
00 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
\cdots\cdots & \cdots\cdots \\
00 \ldots 1 & X \ldots X
\end{array}\right]
$$

quanto a [ALC4] [BLC4] teremos:

$$
\left[\begin{array}{c:c:c:c}
010 \ldots 0 & 000 \ldots 0 & \ldots & 000 \ldots 0 \\
001 \ldots 0 & 000 \ldots 0 & \ldots & 000 \ldots 0 \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
000 \ldots 1 & 000 \ldots 0 & \ldots & 000 \ldots 0 \\
XXX \ldots X & XXX \ldots X & \ldots & XXX \ldots X \\
\hdashline
000 \ldots 0 & 010 \ldots 0 & \ldots & 000 \ldots 0 \\
000 \ldots 0 & 001 \ldots 0 & \ldots & 000 \ldots 0 \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
000 \ldots 0 & 000 \ldots 1 & \ldots & 000 \ldots 0 \\
XXX \ldots X & XXX \ldots X & \ldots & XXX \ldots X \\
\hdashline
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
\hdashline
000 \ldots 0 & 000 \ldots 0 & \ldots & 010 \ldots 0 \\
000 \ldots 0 & 000 \ldots 0 & \ldots & 001 \ldots 0 \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
000 \ldots 0 & 000 \ldots 0 & \ldots & 000 \ldots 1 \\
XXX \ldots X & XXX \ldots X & \ldots & XXX \ldots X
\end{array}\right]
\qquad
\left[\begin{array}{c}
00 \ldots 0 \\
00 \ldots 0 \\
\cdots\cdots \\
00 \ldots 0 \\
1Z \ldots Z \\
\hdashline
00 \ldots 0 \\
00 \ldots 0 \\
\cdots\cdots \\
00 \ldots 0 \\
01 \ldots Z \\
\hdashline
\cdots\cdots \\
\hdashline
00 \ldots 0 \\
00 \ldots 0 \\
\cdots\cdots \\
00 \ldots 0 \\
00 \ldots 1
\end{array}\right]
$$

Em ambos os casos, os 'Z' de BLC3 ou BLC4 ocorrem nas mesmas linhas onde encontramos os 'X' de ALC3 ou ALC4 , sendo que estas linhas são as linhas IECL(I) , I=1,2, ... , K, sendo para FCCL3 os índices estruturais de controlabilidade para FCCL1 e para FCCL4 os índices de FCCL2.

As formas FCCL2 e FCCL4 são normalmente mais citadas nos diversos trabalhos que usam tal espécie de forma canônica. Para FCCL2 a forma de [ALC2] [BLC2] é justificada pelo seguinte raciocínio: se B(.,I) é o J-ésimo vetor da matriz CL2 , após a transformação de similaridade este vetor passará a ser representado por um vetor igual a J-ésima coluna da matriz IDN , com isso o vetor A(.,.)**M*B(.,I) que pertence a CL2 , sendo sua (J+M)-ésima coluna, após a transformação terá a forma da (J+M)-ésima coluna de IDN com M=0,1, ... , IRC(J)-1 e I=1,2, ... , K; com isto justificamos a forma de [BLC2] e as colunas não estruturais de [ALC2]. O vetor A(.,.)**IRC(I)*B(.,I) é o [IRC(I)+1]-ésimo vetor obtido com B(.,I) logo é o primeiro linearmente dependente da sequência com B(.,I) e este vetor aparece em [ALC2] como sendo sua IECL(I)-ésima coluna uma vez que B(.,I) é a [IECL(I)-IRC(I)+1]-ésima coluna de CL2(.,.), consequentemente BLC2(.,I) é a [IECL(I)-IRC(I)+1]-ésima coluna de IDN, e portanto o vetor A(.,.)**IRC(I)*B(.,I) = A(.,.)*[A(.,.)**(IRC(I)-1)*B(.,I)] após a transformação será ALC2(.,.)x[(IECL(I)-IRC(I)+1+IRC(I)-1)-ésima coluna de IDN] = IECL(I)-ésima coluna de ALC2(.,.). Para todos os valores de I obtemos todas as colunas estruturais de [ALC2].

Este mesmo raciocínio é válido no caso de [ALC1] ; só seve ser feita a explicação para o fato desta matriz ser triangular, o que se baseia em que de acordo com CL1, A(.,.)**IRC(I)*B(.,I) vetor de ALC1(.,.) é expresso como combinação linear de A(.,.)**M*B(.,J) para M=0, ... , [IRC(J)-1] com J≤I logo não depende dos vetores com J>I.

Para mostrarmos a consistência das formas que chamaremos do segundo tipo, FCCL3 e FCCL4 , vamos usar uma sequência de idéias computacionais que vai também mostrar algumas características que foram exploradas por DENERY, GUPTA e outros como veremos no capítulo quatro.

Estas características exploradas por DENERY e GUPTA estão identificadas nas formas alternativas de Bingulac, formas estas que definirão mais propriamente índices estruturais, isto é, índices que se dados podem definir totalmente a condição do sistema em termos dos parâmetros que o definem.

A forma FCCB1 é uma forma alternativa de FCCL2 , sendo que em FCCL2 apenas temos a noção dos índices relativos, indicados pelas ordens dos blocos principais, mas não é tão evidente quais os parâmetros 'X' de [ALC2] que são nulos e como veremos mais adiante em FCCL2 , os parâmetros nulos e não nulos estão convenientemente isolados. O número destes parâmetros foi relacionado por Bingulac com um número mínimo de parâmetros que definem uma forma FCCB2 |7|. Também com respeito ao número de parâmetros em FCCB2 mostraremos que em qualquer uma de suas construções o número é sempre o mesmo o que não é tão evidente devido a topologia desta forma canônica.

Para obtermos estas formas vamos construir mais alguns algorítmos e definir alguns índices:

ALGORÍTMO 11: CB1[RAB]

1.- Selecione as N primeiras colunas linearmente independentes de RAB começando pela coluna B(.,1).

DEFINIÇÃO 7: Índice Estrutural de Controlabilidade das formas canônicas de Bingulac.

Chamaremos de índice estrutural de controlabilidade de Bingulac ao valor IECB(I) gerado pelo seguinte algorítmo:

ALGORÍTMO 12

1.- L=0 , J=0

2.- L=L+1 ; I=0

3.- I=I+1

4.- se IRC(I)≥L vá para 6.-
se não siga

5.- se I=K vá para 2.-
se não vá para 3.-

6.- J=J+1

7.- se IRC(I)=L faça IECB(I)=J e siga
se não somente siga

8.- se J=N pare
se não vá para 5.-

Este algorítmo define apenas a localização das colunas estruturais; de acordo com CB1, IECB(I) indica qual é a localização da coluna estrutural derivada de B(.,I); embora estes índices sejam calculados em algorítmos separados, veremos mais adiante que o algorítmo escolhido para obter estas formas obtém implicitamente tais índices, da mesma forma que obterá os índices relativos.

FORMA CANÔNICA CONTROLÁVEL FCCB1

$$\left[\begin{array}{c|c|c} CB1IN & 0 & 0 \\ \hline 0 & IDL & 0 \\ \hline 0 & 0 & IDN \end{array}\right] *H* \left[\begin{array}{c|c|c} CB1 & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right] = \left[\begin{array}{c|c|c} ABC1 & BBC1 & TBC1 \\ \hline CBC1 & D & 0 \\ \hline TBC1IN & 0 & 0 \end{array}\right] =HBC1$$

Com esta forma e tendo inspiração no que seria a forma observável, chegaremos à uma forma alternativa de FCCL4 , a forma FCCB2 , que mantém para BBC2 uma forma parecida com a BBC1 e para ABC2 uma forma parecida com o que seria AB01 , sendo que as linhas estruturais de ABC2 coicidem com as linhas não nulas de BBC2 , sendo esta forma muito útil no problema de alocação de polos e estimação de estado, ao passo que FCCB1 é útil no problema de identificação de parâmetros.

Nestas formas de Bingulac observaremos que os elementos 'X' nulos aparecem agrupados no final das linhas ou colunas estruturais. Foi com base nestas observações que Bingulac estabeleceu o número mínimo de parâmetros que pode representar um sistema, desde que este seja definido por um determinado conjunto de índices relativos.

Voltando a FCCB1 , [ABC1] [BBC1] assume a seguinte forma:

$$\begin{bmatrix} 0 & \cdots & X & \cdots & X & \cdots & X \\ 0 & & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & & \vdots \\ 0 & & \vdots & & \vdots & & \vdots \\ 1 & & \vdots & & \vdots & & \vdots \\ 01 & & \vdots & & \vdots & & \vdots \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ \vdots & & 1X & & \vdots & & \vdots \\ \vdots & & 01 & & \vdots & & \vdots \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ \vdots & & \vdots & & 1X & & \vdots \\ \vdots & & \vdots & & 01 & & \vdots \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1X \end{bmatrix} \qquad \begin{bmatrix} 10 & \cdots & 0 \\ 01 & \cdots & 0 \\ \cdots & \cdots & \cdots \\ 00 & \cdots & 1 \\ 00 & \cdots & 0 \\ \vdots & & \vdots \\ 00 & \cdots & 0 \end{bmatrix}$$

Neste caso podemos verificar pelo próprio algorítmo que gera CB1 , que as colunas estruturais são as colunas IECB(I) , I=1,2, ... , K, sendo que as K primeiras linhas de ABC1 não contém os elementos '1' assim como estas são as únicas linhas não nulas de BBC1 e que sozinhas formam a matriz IDK.

Já neste estágio podemos contar o número de parâmetros que representa a dupla A,B do sistema e este número será o número de 'X' de ABC1 pois BBC1 não tem parâmetros 'X'.

Em cada coluna IECB(I) , que é a J-ésima coluna estrutural de ABC1(.,.) , o número de elementos 'X' é NXE(I) onde:

$$NXE(I) = K + IECB(I) - J$$

a soma de NXE(I) é portanto o número de elementos 'X' de ABC1 e este deve ser também o número de elementos não nulos de ALC2 uma vez que CL2 e CB1 estão relacionadas por uma matriz permutação de colunas.

Este valor terá algo a ver com o número de parâmetros 'X' não nulos de [ABC2] [BBC2] gerados pelos seguintes algorítmos e matrizes:

ALGORÍTMO 13: C3

1.- DO 3 I=1,K

2.- J=IECB(V) para uma sequência qualquer de V entre 1 e K , sendo que é tomado um valor de V para cada valor de I e não se repetem os valores de V.

3.- C3(I,.) = CB1IN(J,.)

ALGORÍTMO 14: C3A

1.- DO 3 I=1,K

2.- J=IECB(V) como no algorítmo anterior

3.- C3A(.,.) = IDN(J,.)

Estas matrizes definem as seguintes matrizes auxiliares de observabilidade

SAC3[A,B] = SAC[A,C3] - ALGORÍTMO 2

SAC3A[A,B] = SAC[ABC1,C3A] - ALGORÍTMO 2

das quais extraimos as seguintes matrizes de transformação de similaridade.

CB2[A,B] = OB1[SAC3,C3]

CB2A[A,B] = OB1[SAC3A,C3A]

sendo OB1 uma matriz equivalente a CB1 no caso observável.

Destas matrizes CB2 e CB2A obtemos finalmente:

FORMA CANÔNICA CONTROLÁVEL FCCB2

$$
\left[\begin{array}{c|c|c} CB2 & 0 & 0 \\ \hline 0 & IDL & 0 \\ \hline 0 & 0 & IDN \end{array}\right] * H * \left[\begin{array}{c|c|c} CB2IN & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & JDN \end{array}\right] = \left[\begin{array}{c|c|c} ABC2 & BBC2 & TBC2 \\ \hline CBC2 & D & 0 \\ \hline TBC2IN & 0 & 0 \end{array}\right] =
$$

$$
= \left[\begin{array}{c|c|c} CB2A & 0 & 0 \\ \hline 0 & IDL & 0 \\ \hline 0 & 0 & IDN \end{array}\right] * HBC1 * \left[\begin{array}{c|c|c} CB2AIN & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right] = HBC2
$$

Com uma transformação de similaridade de permutação conveniente podemos de HBC2 obter HLC4.

Se nos algorítmos 13 a 14 fixarmos uma certa lei para dispormos de V em relação aos índices relativos, então a matriz ABC2 será estruturalmente invariante, independentemente da ordem dos índices relativos. E embora, a nosso conhecimento, nenhum autor já a tenha explorado, esta ordem dos índices relativos afeta somente o número de parâmetros não nulos 'X' de BBC2. No entanto antes de explorarmos este fato vamos colocar o par [ABC2] [BBC2] conforme Bingulac:

$$
\begin{bmatrix}
00 \cdots 010 & \cdots & \cdots & 0 \\
\vdots \quad 1 & & & \vdots \\
\vdots & \ddots & & \vdots \\
\vdots & \quad 1 & & \vdots \\
X \cdots & X0 & \cdots & 0 \\
\vdots & \quad 1 & & \vdots \\
\vdots & & \ddots & \vdots \\
\vdots & & \quad 1 & \vdots \\
X \cdots & \cdots & X0 \cdots & 0 \\
\vdots & & \quad 1 & \vdots \\
\vdots & & \ddots & \vdots \\
\vdots & & & 1 \\
X \cdots & \cdots & \cdots & X
\end{bmatrix}
\qquad
\begin{bmatrix}
0 \cdots 0 \\
\vdots \quad \vdots \\
0 \cdots 0 \\
0 \cdots 1X \\
0 \cdots 0 \\
\vdots \quad \vdots \\
0 \cdots 0 \\
1X \cdots X \\
0 \cdots 0 \\
\vdots \quad \vdots \\
0 \cdots 0 \\
01 \cdots X
\end{bmatrix}
$$

As linhas estruturais nesta forma são definidas por IECB(I) dado pelo seguinte algorítmo:

ALGORÍTMO 15

1.- L=0 ; J=0

2.- L=L+1 ; I=0

3.- I=I+1

4.- se IRC(V) $\geqslant$ L vá para 6.-
(onde V e I mantém a mesma relação do algorítmo 13 ou 14)
se não siga

5.- se I=K vá para 2.-
se não vá para 3.-

6.- J=J+1

7.- se IRC(V) = L faça IECB(V) = J e siga
se não somente siga

8.- se J=N pare
se não vá para 5.-

Para este tipo de construção Bingulac tem usado como lei que rege o índice V dos algorítmo 13, 14 e 15 uma sequência de V que coloque os índices relativos na forma crescente, e para utilização futura quando encontrarmos dois índices iguais, estes serão colocados na ordem em que foram encontrados no vetor IRC(I), I=1,2, ... , K. A vantagem de se colocar nesta forma é que o número de elementos 'X' de ABC2 diminui, pelo menos pelo que sugere a topologia de ABC2, e se sempre mantivermos esta lei então, não importando a ordem em que se apresentam os índices relativos, sempre que tivermos um dado conjunto de índices, a forma de ABC2 será sempre a mesma. Neste ponto é conveniente antecipar que somente para esta lei posso assegurar que nenhum elemento 'X' de ABC2 é necessariamente nulo.

Em resumo, as formas alternativas de Bingulac além de serem de obtenção um pouco menos trabalhosa, evidenciam determinadas características interessantes das formas canônicas que exploram o aspecto controlabilidade - observabilidade.

CAPÍTULO III

Neste capítulo vamos procurar expor o que tem sido feito no sentido de se obter estas formas canônicas com ênfase em precisão;dois trabalhos que abordam este aspecto são os de JORDAN-SRIDHAR |2| e de DALY |3|.

Colocaremos algumas simplificações e observações nos métodos e a partir destes, no capítulo seguinte, serão derivadas algumas características, outras formas alternativas e algorítmos para se obter as formas canônicas ou funções de transferência.

Finalmente vamos ressaltar que a procura de certas facilidades computacionais pode levar a determinados problemas.

Vimos que para obter as formas canônicas FCCL1, FCCL2, FCCL4, FCCB1 e FCCB2 somos obrigados a formar uma matriz de controlabilidade, desta matriz tirar a transformação de similaridade, da inversa desta ultima montar uma matriz auxiliar, da matriz auxiliar construir uma matriz de observabilidade e desta obter a transformação de similaridade que será aplicada ao sistema original. Este procedimento faz uso de muitos cálculos e construções de matrizes nos passos intermediários que nem sempre nos interessam. Jordan e Sridhar, |2|, propuseram um método de se obter a forma FCCL4 de um meio mais preciso e em um número de operações: multiplicações e divisões, de aproximadamente 55% do número de operações referente ao método normal.

O método de Jordan e Sridhar será exposto aqui e aplicado com algumas simplificações às demais formas canônicas.

A análise de Jordan e Sridhar parte da estrutura da forma FCCL4 e usa da equação:

$$B(.,.) = CL4(.,.)*BLC4(.,.) \qquad (1)$$

para deduzir o seguinte sistema de equações.

$$\begin{cases} B(.,1) = CL4(.,IECL(1)) \\ B(.,2) = CL4(.,IECL(2)) + BLC4(IECL(1)),2)*CL4(.,IECL(1)) \\ \text{-----------------------------------} \\ B(.,M) = CL4(.,IECL(M)) + \sum_{I=1}^{M-1} BLC4(IECL(I),M)*CL4(.,IECL(I)) \\ \text{supondo} \quad K=M \end{cases} \tag{2}$$

de onde, resolvendo para CL4(.,IECL(I)) , teríamos:

$$CL4(.,IECL(I)) = \sum_{J=1}^{I} P(J,I)xB(.,J)$$

$$I=1,2, \ldots , M \quad ; \quad e \quad P(I,I)=1. \tag{3}$$

Usando agora a equação:

$$A(.,.)*CL4(.,.) = CL4(.,.)*ALC4(.,.) \tag{4}$$

obtemos as seguintes equações:

$$CL4(.,IECL(L)-J) = A(.,.)*CL4(.,IECL(L)-J+1) - $$
$$- \sum_{I=1}^{M} ALC4(IECL(I),IECL(L)-J+1)*CL4(.,IECL(I))$$

$$L=1,2, \ldots , M \quad e \quad J=1,2, \ldots , IRC(L)-1 \tag{5}$$

$$A(.,.)*CL4(.,IECL(L)-IRC(L)+1) =$$
$$= \sum_{I=1}^{M} ALC4(IECL(I),IECL(L)-IRC(L)+1)*CL4(.,IECL(I))$$

$$L=1,2, \ldots , M \tag{6}$$

se manipularmos as equações (5) e (6) repetidamente teremos:

$$CL4(.,IECL(L)-J) = A(.,.)**J*CL4(.,IECL(L)) -$$

$$- \sum_{I=1}^{M} \sum_{K=0}^{J-1} ALC4(IECL(J),IECL(L)-J+K+1)*A(.,.)**K*CL4(.,IECL(I))$$

$L=1,2,\ ...\ ,M \quad ; \quad J=1,2,\ ...\ ,IRC(L)-1$ (7)

$$A(.,.)**IRC(L)*CL4(.,IECL(L)) =$$

$$= \sum_{I=1}^{M} \sum_{K=0}^{V} ALC4(IECL(I),IECL(L)-IRC(L)+K+1)*A(.,.)**K*CL4(.,ICL(I))$$

$L=1,2,\ ...\ ,M \quad ; \quad V=IRC(L)-1$ (8)

e de (3) em (8) temos:

$$A(.,.)**IRC(L)* \sum_{J=1}^{L} P(J,L)*B(.,J) =$$

$$= \sum_{J=1}^{M} \sum_{I=1}^{M} \sum_{K=1}^{V} ALC4(IECL(J),IECL(L)-IRC(L)+K+1)*P(I,J)*A(.,.)**K*B(.,I)$$

$L=1,2,\ ...\ ,M \quad ; \quad V=IRC(L)-1$ (9)

Aplicando na equação (9) uma versão modificada da sugestão de Kalman;

$$A(.,.)**IRC(L)*B(.,L) = - \sum_{I=1}^{M} \sum_{K=0}^{V} ALFA(L,I,K)*A(.,.)**K*B(.,I)$$

$L=1,2,\ ...\ ,M \quad ; \quad V=IRC(I)-1$ (10)

Jordan e Sridhar obtiveram a seguinte equação:

$$0 = A(.,.)**IRC(L)*\left[\sum_{J=1}^{L} ALFA(L,J,IRC(L))*B(.,J)\right] +$$

$$+ \sum_{I=1}^{M} \sum_{K=0}^{V} ALFA(L,I,K)*A(.,.)**K*B(.,I) \quad ; \quad V = IRC(L)-1 \qquad (11)$$

cuja comparação com a equação (9) sugere o seguinte algorítmo:

ALGORÍTMO 16:

1.- Calcule ALFA(L,J,K) por (11) ; ALFA(L,L,IRC(L))=1

2.- Obtenha P(J,L)=ALFA(L,J,IRC(L)) ; P(J,J)=1

3.- Obtenha os parâmetros de ALC4(.,.) resolvendo

$$\sum_{J=1}^{M} P(I,J)*ALC4(IECL(J),IECL(L)-IRC(L)+K+1)=-ALFA(L,I,K)$$

I=1,2, ... ,M ; L=1,2, ... ,M ; K=0,1,2, ... ,IRC(L)-1

4.- Obtenha os parâmetros de BLC4(.,.) por

$$BLC4(IECL(K),I) + \sum_{J=K+1}^{I-1} P(K,J)*BLC4(IECL(J),I)+P(K,I)=0$$

I=2,3, ... , M ; K=I-1,I-2, ... , 1

5.- tomando agora as equações (3) e (5) obtemos CL4(.,.)

Devemos neste ponto lembrar que foi suposto no trabalho original de Jordan e Sridhar , que os índices relativos estivessem em ordem não crescente, $IRC(I) \geq IRC(J)$, $I<J$, e que neste caso foi dito, dado os vetores que compõe a matriz CL2 , que:

ALFA(I,L,K) = 0 se I≠L mas K>IRC(L) ou
se K=IRC(L) mas I>L

pois interpretando a intenção de Kalman |2| e a equação (11) chegamos a conclusão que:

CL2(.,.)*ALF(.,.) = AN(.,.) (12)

onde a coluna I de AN(.,.) é A(.,.)*IRC(I)*B(.,I) e que -ALFA(L,I,K)=ALF(IECL(I)-IRC(I)+1+K,L) para L=1,2, ... , M; I=1,2, ... , M ; K=0,1, ... , IRC(I)-1, e portanto se formarmos uma matriz ARE(.,.) onde sua I-ésima linha é de ALC4(.,.) a IECL(I)-ésima linha, ou seja ARE(.,.) contém apenas as linhas estruturais de ALC4(.,.), podemos obter ALC4(.,.) e BLC4(.,.) por:

P(.,.)*ARE(.,.) = AFL(.,.) (13)

onde P(.,.) pode ser obtido de ALFA(.,.,.) por

P(I,J) = ALFA(J,I,IRC(J)) para J>I

P(I,I) = 1

P(I,J) = 0 para J<I

e AFL(.,.) também pode ser obtida de ALFA(.,.,.) por

AFL(I,IECL(L)-IRC(I)+K) = -ALFA(L,I,K)

para I=1,2, ... , M ; L=1,2, ... , M ; K=0,1, ... , IRC(L)-1.

De (13) obtemos ARE(.,.), logo estamos obtendo ALC4(.,.) e de PIN(.,.) obtemos as linhas estruturais de BLC4(.,.), ou seja, sendo BRE(.,.) a matriz das linhas estruturais de BLC4(.,.) então:

BRE(.,.) = PIN(.,.)

Relaxando a condição de serem os índices em ordem decrescente podemos obter ALFA resolvendo a seguinte equação matricial:

$$CL2(.,.)*ALF(.,.) = AN(.,.) \tag{12}$$

de onde podemos ver que ALF é a matriz dos coeficientes de dependência das colunas de AN(.,.) ; no caso de um sistema com índices relativos de controlabilidade 3, 1 e 2 teremos:

$$CL2(.,.) * \left[\begin{array}{c|c|c} -ALFA(1,1,0) & -ALFA(2,1,0) & -ALFA(3,1,0) \\ -ALFA(1,1,1) & -ALFA(2,1,1) & -ALFA(3,1,1) \\ -ALFA(1,1,2) & 0 & -ALFA(3,1,2) \\ \hline -ALFA(1,2,0) & -ALFA(2,2,0 & -ALFA(3,2,0 \\ \hline -ALFA(1,3,0 & -ALFA(2,3,0 & -ALFA(3,3,0) \\ -ALFA(1,3,1) & 0 & -ALFA(3,3,1) \end{array}\right] = AN(.,.)$$

que já obtem todos elementos de ALFA e por outro lado a matriz ALF(.,.) aqui já é a matriz com as colunas estruturais de ALC2 , isto é, já obtemos FCCL2 e obteriamos FCCL1 para a mesma distribuição de índices em:

$$CL1(.,.) * \left[\begin{array}{c|c|c} ALF1(1,1) & ALF1(1,2) & ALF1(1,3) \\ ALF1(2,1) & ALF1(2,2) & ALF1(2,3) \\ ALF1(3,1) & ALF1(3,2) & ALF1(3,3) \\ \hline 0 & ALF1(4,2) & ALF1(4,3) \\ \hline 0 & 0 & ALF1(5,3) \\ 0 & 0 & ALF1(6,3) \end{array}\right] = AN(.,.)$$

Voltando ao caso FCCL4 teríamos para (13)

$$
\begin{bmatrix} 1 & ALFA(2,1,1) & ALFA(3,1,2) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} * ARE(.,.) = \left[\begin{array}{c|c|c|c|c|c} -ALFA(1,1,0) & -ALFA(1,1,1) & -ALFA(1,1,2) & -ALFA(2,1,0) & -ALFA(3,1,0) & -ALFA(3,1,1) \\ \hline -ALFA(1,2,0) & 0 & 0 & -ALFA(2,2,0) & -ALFA(3,2,0) & 0 \\ \hline -ALFA(1,3,0) & -ALFA(1,3,1) & 0 & -ALFA(2,3,0) & -ALFA(3,3,0) & -ALFA(3,3,1) \end{array}\right]
$$

de onde obteríamos ARE(.,.) e para BRE(.,.) teríamos:

$$
BRE(.,.) = PIN(.,.) = \begin{bmatrix} 1 & -ALFA(2,1,1) & -ALFA(3,1,2) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

No caso das formas de Bingulac só devemos rearrumar as colunas de CL2 para obter CB1 e consequentemente rearrumarmos as linhas de ALF com uma matriz inversa da que rearrumou as colunas CL2 ; pela característica de ser permutação, a inversa é a transposta da matriz que causou a permutação nas colunas de CL2 . Com isso teríamos no caso 3,1,2 :

$$
CB1(.,.) * \left[\begin{array}{c|c|c} -ALFA(1,1,0) & -ALFA(2,1,0) & -ALFA(3,1,0) \\ -ALFA(1,2,0) & -ALFA(2,2,0) & -ALFA(3,2,0) \\ -ALFA(1,3,0) & -ALFA(2,3,0) & -ALFA(3,3,0) \\ -ALFA(1,1,1) & -ALFA(2,1,1) & -ALFA(3,1,1) \\ -ALFA(1,3,1) & 0 & -ALFA(3,3,1) \\ -ALFA(1,1,2) & 0 & -ALFA(3,1,2) \end{array}\right] = AN(.,.)
$$

e colocando as colunas de ALF(.,.) na sequência dos índices crescentes, ou seja, colunas 2, 3 e 1, teremos as colunas estruturais de ABC1 , logo FCCB1 , e obteremos as linhas estruturais de ABC2 resolvendo:

$$\begin{bmatrix} ALFA(2,1,1) & ALFA(3,1,2) & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * ARE = - \left[\begin{array}{c|c|c|c|c|c} ALFA(1,2,0) & ALFA(2,2,0) & ALFA(3,2,0) & 0 & 0 & 0 \\ \hline ALFA(1,3,0) & ALFA(2,3,0) & ALFA(3,3,0) & ALFA(1,3,1) & ALFA(3,3,1) & 0 \\ \hline ALFA(1,1,0) & ALFA(2,1,0) & ALFA(3,1,0) & ALFA(1,1,1) & ALFA(3,1,1) & ALFA(1,1,2) \end{array}\right]$$

Como se pode observar, o que se propõe aqui é uma metodologia bastante flexível que faz uso dos parâmetros 'X' nulos , não se restringe às formas de Luenberger, mas também é aplicada às formas de Bingulac e a outras formas, que desenvolveremos no capítulo seguinte, sem a relativa complicação que notamos no trabalho de Jordan e Sridhar e que só fica melhor entendida quando as equações (12) e (13) são desenvolvidas.

Para finalizarmos esta metodologia devemos verificar como obter as matrizes de transformação de similaridade, aqui isto só será visto para o caso FCCL4 pois como já foi anteriormente dito, a forma FCCB2 é obtida a partir de alguns rearranjos feitos sobre a forma FCCL4. Escolhemos partir desta forma para as demais por que em (5) e (6) temos uma descrição mais simples além de que na equação (13), por P(.,.) ser triangular, sua inversão para obter as linhas estruturais de BLC4 e ALC4 , ARE e BRE, também é mais simples.

Como uma simplificação final vamos dizer que se operarmos sobre as linhas de [P(.,.)|¦|AFL(.,.)¦IDM] reduzindo P(.,.) a IDM obteremos [IDM¦ARE(.,.)¦BRE(.,.)] pois:

$$[P(.,.)]^{-1} * [AFL(.,.) \vdots IDM] = [ARE(.,.) \vdots BRE(.,.)]$$

Isto é, se formarmos a matriz composta $[P(.,.) \vdots AFL(.,.) \vdots IDM]$ e operarmos sobre suas linhas de tal modo a obter no lugar de P(.,.) a matriz IDM, então a matriz que operou sobre as linhas é PIN(.,.) e ficaremos com $[IDM \vdots ARE(.,.) \vdots BRE(.,.)]$.

Vamos terminar então com um algorítmo que obtenha a forma de Luenberger, ALC4 e BLC4;

<u>ALGORÍTMO 16</u>:

.- Obter ALF(.,.)

1.- ALF(.,.) = CL2IN(.,.)*AN(.,.)

.- Obter AFL(.,.)

2.- DO 5.- KA=1,M

3.- DO 5.- KB=1,M

4.- DO 5.- KC=IECL(KB)-IRC(KB)+1, IECL(KB)

5.- AFL(KA,KC) = ALF(KC,KB)

.- Obter P(.,.)

6.- P(.,.) = IDM

7.- DO 9.- KB = KA+1,M

8.- DO 9.- KB = KA+1,M

9.- se IRC(KA)>IRC(KB) então
P(KA,KB) = -ALF(IECL(KA)-IRC(KA)+IRC(KB)+1 , KB) ;
se não continue.

.- Obter BRE(.,.)

10.- BRE(.,.) = PIN(.,.)

.- Obter ARE(.,.)

11.- ARE(.,.) = PIN(.,.) * AFL(.,.)

.- Final.

12.- de ARE(.,.) obter ALC4(.,.)

13.- de BRE(.,.) obter BLC4(.,.)

Com esse algorítmo 16 obtemos ALC4 e BLC4 , com (3) e (5) obtemos CL4 e de C e CL4IN obtemos CLC4 , obtendo assim FCCL4.

Esta metodologia dificilmente pode ser usada para obter FCCL3 devido aos elementos 'Y' de BCL3 , podendo ser usada quando estes elementos forem nulos, o que vai ocorrer, como veremos mais tarde, quando os índices relativos de controlabilidade ou forem crescentes, ou se forem decrescentes não o sejam, entre um índice e o posterior, de mais do que uma unidade.

TRANSFORMAÇÕES DE SIMILARIDADE ELEMENTARES (TSE)

Com os intuitos de diminuir o número de operações dos tipos multiplicação e divisão, para melhor precisão na obtenção das formas canônicas, e de divulgar um método computacional que possa auxiliar o entendimento do que ocorre quando em uma operação de similaridade estudaremos as T.S.E.. Baseado neste método formas auxiliares foram inicialmente propostas, tais formas serão vistas no capítulo seguinte, assim como obtivemos os elementos 'Y' de FCCL3 e as características dos 'X' nulos nas diversas formas canônicas.

Como transformação de similaridade elementar entenderemos a transformação de similaridade cuja matriz de transformação é uma matriz que representa operações elementares.

Estas matrizes serão:

PE[I,J] : permuta as linhas (colunas) I e J

MU[AA,I] : multiplica a linha (coluna) I por AA

CO[AA,I,J]: soma AA vezes a linha I (coluna J) à linha J (coluna I)

destas matrizes podemos observar que:

$$PE^{-1}[I,J] = PE[I,J] = PEIN[I,J]$$

$$MU^{-1}[AA,I] = MU[1/AA,I] = MUIN[AA,I]$$

$$CO^{-1}[AA,I,J] = CO[-AA,I,J] = COIN[AA,I,J]$$

com esse conhecimento antecipado das inversas, o número de operações diminui pois não precisamos do cálculo das matrizes inversas.

Assim sendo e tomando a configuração original do sistema, chamaremos de matrizes operações elementares sobre linhas às matrizes:

$$PEL[I,J] = \left[\begin{array}{c:c:c} PE[I,J] & 0 & 0 \\ \hdashline 0 & IDL & 0 \\ \hdashline 0 & 0 & IDN \end{array}\right] \text{ - operação: permutação}$$

$$MUL[AA,I] = \left[\begin{array}{c:c:c} MU[AA,I] & 0 & 0 \\ \hdashline 0 & IDL & 0 \\ \hdashline 0 & 0 & IDN \end{array}\right] \text{ - operação: multiplicação}$$

$$COL[AA,I,J] = \left[\begin{array}{c:c:c} CO[AA,I,J] & 0 & 0 \\ \hdashline 0 & IDL & 0 \\ \hdashline 0 & 0 & IDN \end{array}\right] \text{ - operação: combinação linear}$$

que serão pré multiplicadoras da matriz do sistema. De forma análoga podemos também construir as matrizes operações elementares sobre colunas:

$$PEC[I,J] = \left[\begin{array}{c:c:c} PE[I,J] & 0 & 0 \\ \hdashline 0 & IDM & 0 \\ \hdashline 0 & 0 & IDN \end{array}\right] \text{ - operação: permutação}$$

$$MUC[AA,I] = \left[\begin{array}{c|c|c} MU[AA,I] & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right] \quad \text{- operação: multiplicação}$$

$$COC[AA,I,J] = \left[\begin{array}{c|c|c} CO[AA,I,J] & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right] \quad \text{- operação: combinação linear}$$

que serão pós-multiplicadoras da matriz do sistema; ou seja, em uma transformação de similaridade orientada por linhas, isto é, se a operação que se pretende efetuar foi sugerida por uma operação por linhas, teremos:

PEL[I,J] * H(.,.) * PECIN[I,J]

MUL[AA,I] * H(.,.) * MUCIN[AA,I]

COL[AA,I,J] * H(.,.) * COCIN[AA,I,J]

Ās operações de similaridade elementares como sugeridas acima; ou ās orientadas por colunas chamaremos de transformações de similaridade elementares.

Podemos assim obter algumas transformações; se aplicarmos, por exemplo, uma sērie de operações elementares sobre linhas e depois completarmos a transformação com as respectivas inversas, sobre colunas, teremos uma sērie de operações elementares que serā representada por:

PC[I(1),J(1);I(2),J(2);...;I(K),J(K)] =

= PEC[I(1),J(1)]*PEC[I(2),J(2)]*...*PEC[I(K),J(K)]

PL[I(1),J(1);I(2),J(2);...;I(K),J(K)] =

= PEL[I(K),J(K)]*...*PEL[I(2),J(2)]*PEL[I(1),J(1)]

MC[AB(1),I(1);AB(2),I(2);...;AB(K),I(K)] =

= MUC[AB(1),I(1)]*MUC[AB(2),I(2)]*...*MUC[AB(K),I(K)]

ML[AA(1),I(1);AA(2),I(2);...;AA(K),I(K)] =

= MUL[AA(K),I(K)]*...*MUL[AA(2),I(2)]*MUL[AA(1),I(1)]

CC[AB(1),I(1),J(1);AB(2),I(2),J(2);...;AB(k),I(K),J(K)] =

= COC[AB(1),I(1),J(1)]*COC[AB(2),I(2),J(2)]*...*COK[AB(K),I(K),J(K)]

CL[AA(1),I(1),J(1);AA(2),I(2),J(2);...;AA(K),I(K),J(K)] =

= COL[AA(1),I(1),J(1)]*COL[AA(2),I(2),J(2)]*...*COL[AA(K),I(K),J(K)]

onde

PCIN(.,.) = PL(.,.) (14)

MCIN(.,.) = ML(.,.) se AB(I)=1./AA(I) (15)

CCIN(.,.) = CL(.,.) se AB(I)=-AA(I) (16)

Ā este tipo de transformação de similaridade chamaremos de transformação de similaridade por blocos. Para efetuarmos estas operações foram construídas as subrotinas PERMUT, MULTIP e COMBLI.

No intuito de minimizar o conteúdo da memória utilizada e para evitar o uso de COMMON , para tornar a subrotina de uso imediato ao usuário, as subrotinas procuram os elementos da matriz envolvida como eles se encontram na memória, guardando então para a subrotina apenas o endereço do início do "ARRAY" da matriz em questão.

Para isso basta lembrar que o elemento A(I,J) da matriz A pode ser alcançado por A((J-1) * MD + I) onde MD é o valor do primeiro índice relativo a matriz A na declaração DIMENSION do programa de chamada da subrotina.

Podemos agora explicar as subrotinas elementares:

PERMUT($,A,M,N,MD,I,J,IT,IC,IERR)

A: matriz a ser operada

M,N: dimensões de A

MD: primeiro índice de A no DIMENSION principal

I,J: fatores da operação

IT: tipo de operação sobre A:

IT = 1HS → similaridade elementar.

IT ≠ 1HS → operação elementar.

IC: condição de operação.

IC = 1HC → por colunas.

IC ≠ 1HC → por linhas.

IERR: indicador de erro sobre as dimensões ou sobre os operadores, IERR=1.

OBS 1: se MD=0 será assumido MD=M

OBS 2: se houver algum erro, IERR=1, o programa sairá para o "statement" declarado em $.

OBS 3: se I=J não é considerado erro e após os testes de dimensão volta ao programa de chamada sem efetuar nenhuma operação.

Função: PE(I,J)*A(.,.) se IT≠1HS e IC≠1HC

A(.,.)*PE(I,J) se IT≠1HS e IC=1HC

PE(I,J)*A(.,.)*PE(I,J) se IT=1HS e IC=qualquer

MULTIP($,A,M,N,MD,I,AA,IT,IC,EPS,IERR)

A,M,N,MD,IT,IC,IERR,$ como em PERMUT

I,AA: fatores da operação

EPS: precisão

OBS 1: se, |AA-1|<EPS então não há operação pois se considera multiplicação por 1.

OBS 2: se, $|AA+1|<EPS$, então troca-se os elementos em questão pelo seu negativo.

OBS 3: $|AA|<EPS$, anula os elementos se $IT \neq 1HS$ erro se $IT=1HS$

Função: MU(AA,I)*A(.,.) se $IT \neq 1HS$ e $IC \neq 1HC$

A(.,.)*MU(AA,I) se $IT \neq 1HS$ e $IC=1HC$

MU(AA,I)*A(.,.)*MU(1/AA,I) se $IT=1HS$ e $IC \neq 1HC$

MU(1/AA,I)*A(.,.)*MU(AA,I) se $IT=1HS$ e $IC=1HC$

COMBLI($,A,M,N,MD,IA,JA,AA,IT,IC,EPS,IERR)

A,M,N,MD,IT,IC,EPS,IERR,$ como em MULTIP.

IA,JA,AA fatores de operação.

OBS 1: $|AA-1|<EPS \rightarrow AA=1$

$|AA+1|<EPS \rightarrow AA=-1$

OBS 2: $|AA|<EPS$ não opera.

Após esta apresentação das subrotinas elementares devemos fazer uma observação quanto ao requisito memória-operações, que é dizer que o uso das transformações de similaridade elementar embora não utilize muita memória pode efetuar operações redundantes como veremos na seção seguinte; por outro lado, as transformações de similaridade elementares por blocos podem evitar as operações redundantes, mas necessitam memorizar a sequência de operações para depois obterem as inversas convenientemente. Veremos no final do capítulo como usar a idéia de blocos sem emprego de mais posições de memória.

Finalmente colocamos a seguir a notação usada por Daly com relação às transformações de similaridade elementares:

EP[H;I(1),J(1);...] = PL(.,.)*H(.,.)*PL(.,.)

EM[H;AA(1),I(1);...] = ML(.,.)*H(.,.)*MLIN(.,.)

EC[H;AA(1),I(1),J(1);...] = CL(.,.)*H(.,.)*CLIN(.,.)

SOBRE O MÉTODO DE DALY

Finalmente chegamos a um método para obter as formas canônicas que pode ser mais rápido e mais preciso, e cujo número de operações do tipo multiplicação e divisão envolvido é de aproximadamente 60% das operações envolvidas, no método de Jordan e Sridhar, sendo de aproximadamente 25% do número de operações envolvidas, no método original, de Luenberger ou de Bingulac. No caso de só calcularmos as matrizes A, B e C nas formas canônicas, a redução em relação ao método original é de 85% das operações, isto é, obtemos as formas canônicas das matrizes do sistema com um número de operações da ordem de 15% do número de operações envolvidas nos métodos de Luenberger e Bingulac.

Uma outra vantagem com relação ao método de Jordan-Sridhar é que o método de Daly pode obter todas as formas canônicas propostas quando discutimos as formas de Luenberger e de Bingulac.

Daly propos o método para as formas de Luenberger; aqui o estenderemos para as formas de Bingulac e para as formas especiais construídas no capítulo seguinte.

Como apresentação do método é proposto um algorítmo para obtermos as formas controláveis para sistemas de uma só entrada, quando se confundem FCCL1, FCCL2, FCCB1 e também se confundem FCCL3, FCCL4, FCCB2:

ALGORÍTMO 17: FCCL1

.- Inicialização

1.- Obter a matriz do sistema 4(.,.)

2.- I=1 (I=contador de linhas)

3.- J=N+1 (J=contador de colunas)

.- Procura do melhor pivot

4.- L=valor de K onde ocorre max|H(K,J)| para K=I,J+1,...,N

5.- G=H(L,J)

.- Teste de controlabilidade

6.- se G=0 o sistema não é controlável, pare.
se G≠0 siga.

.- Colocando o pivot na linhas I e coluna J.

7.- H(.,.) = EP[H;I,J]

8.- H(.,.) = EM[H,1/G,I]

.- Eliminação por colunas

9.- H(.,.) = EC[H;-H(1,J),I,1;...;-H(I-1,J),I,I-1;
-H(I+1,J),I,I+1;...;-H(N,J),I,N]

.- Teste de fim de algorítmo

10.- se I=N vá para 12.-
se I≠N siga.

11.- J=I; I=I+1; vá para 4.-

12.- Pare

Observamos que quando estamos para iniciar o trabalho na, por exemplo, linha 4 teremos:

$$[\hat{A} \vdots \hat{B}] = \left[\begin{array}{cccccc|c} 0 & 0 & V & X & X & X & 1 \\ 1 & 0 & V & X & X & X & 0 \\ 0 & 1 & V & X & X & X & 0 \\ 0 & 0 & Y & Z & Z & Z & 0 \\ 0 & 0 & Y & Z & Z & Z & 0 \\ 0 & 0 & Y & Z & Z & Z & 0 \end{array}\right]$$

; onde 'X', 'Y', 'Z' e 'V' podem assumir qualquer valor e onde a primeira coluna $\hat{A}$ é $\hat{A}$*$\hat{B}$, pela forma de $\hat{B}$, e a segunda coluna de $\hat{A}$ é $\hat{A}$**2*$\hat{B}$, pela forma de $\hat{A}$*$\hat{B}$, sendo então $\hat{B}$, $\hat{A}$*B e $\hat{A}$**2*$\hat{B}$ linearmente independentes. Dada a forma de $\hat{A}$**2*$\hat{B}$ para que $\hat{A}$**3*$\hat{B}$ = = $\hat{A}$*($\hat{A}$**2*$\hat{B}$) seja também linearmente independente dos demais é necessário que pelo menos um dos 'Y' não seja nulo; se todos 'Y' forem nulos os valores 'V' darão os coeficientes de dependência de $\hat{A}$**3*$\hat{B}$ com os vetores já selecionados, e a submatriz de elementos 'Z' indicará quais os polos não controláveis.

Para obtermos FCCL3 há dois meios, cada um de duas fases:

ALGORÍTMO 18: FCCL3A

.- PRIMEIRA FASE

.- FCCL1

.- SEGUNDA FASE

Nesta segunda fase vamos supor que existe um vetor auxiliar $\bar{C}$ que é a última linha da matriz IDN e vamos obter FCOL1 ou:

2.- J=I-1 (da primeira fase I=N)

3.- H(.,.)=EC(H;H(I,I),I,J;...;H(I,N),N,J)

4.- se I=2 pare; se não siga

5.- I=I-1; vá para 2

ALGORÍTMO 19: FCCL3B

.- PRIMEIRA FASE

.- Inicialização

1.- construir a matriz do sistema H(.,.)

2.- I=1 (I=contador de linhas)

3.- J=N+1 (J=contador de colunas)

.- Procura do melhor pivot.

4.- L=valor de K onde ocorre max|H(K,J)| para K=I,I+1,...,N

5.- G=H(L,J)

.- Teste de controlabilidade.

6.- se G=0 , o sistema não é controlável, pare
se G≠0 , siga

.- Colocando o pivot na linha I e coluna J

7.- H(.,.) = EP[H;I,L]

8.- H(.,.) = EM[H;1/G,I]

.- Eliminando a coluna

9.- H(.,.) = EC[H;-H(I+1,J),I,I+1;...;-H(N,J),I,N]

.- Teste de fim de fase

10.- se I=N vã para 12.-
se I≠N siga

11.- J=I; I=J+1; vã para 4.-

.- SEGUNDA FASE

12.- H(.,.) = EC[H;H(I,I),I,J;...;H(I,N),N,J]

13.- se I=2 pare; se não siga

14.- I=J; J=I-1; vã para 12.-

A forma obtida pelo algorĩtmo 19 ẽ uma modificação da forma de Luenberger original e tem a seguinte forma:

$$[\hat{A} \vdots \hat{B}] = \left[\begin{array}{cccccc:c} X & X & X & X & X & X & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right]$$

os motivos de usarmos este tipo de forma são:

a) não alteramos a localização dos '1'

b) a alteração para irmos do algorĩtmo 17 ã forma final no algorĩtmo 19 ẽ simples.

Nos algorĩtmos 18 e 19, se usarmos na segunda fase o recurso de $\hat{C}$ para obter FCOL1 chegamos ã forma canônica original de Luenberger, sendo que se obedecermos os passos propostos cairemos na forma canônica modificada.

Quanto a diferença entre os algorĩtmos 18 e 19, ela estã no fato de que no algorĩtmo 18 teremos algumas operações redundantes, efetuando assim mais operações que no algorĩtmo 19.

Aqui veremos que usar a T.S.E por blocos não ẽ mais "econômico" que usar a T.S.E simples, e com a mesma metodologia usada para demonstrarmos isso demonstraremos que o algorĩtmo 19 ẽ mais "econômico" do que o algorĩtmo 18.

Para comparar a T.S.E simples com a T.S.E por blocos vamos tomar inicialmente o algorítmo 17 e verificar quais iterações ocorrem, observando que operação de multiplicação ou divisão por ± 1 não será contada, assim como multiplicação por zero e divisão de um número por si mesmo, pois são operações que ou não induzem êrros ou são evitadas.

Tomemos a configuração que existe quando vamos operar sobre a linha I.

$$
H = \begin{array}{cc}
\left[\begin{array}{cccccc|c|cccccc}
0 & 0 & X & X & X & X & 1 & X & X & X & 0 & 0 & 0 \\
1 & 0 & X & X & X & X & 0 & X & X & X & 0 & 0 & 0 \\
0 & 1 & X & X & X & X & 0 & X & X & X & 0 & 0 & 0 \\
0 & 0 & X & X & X & X & 0 & X & X & X & 1 & 0 & 0 \\
0 & 0 & X & X & X & X & 0 & X & X & X & 0 & 1 & 0 \\
0 & 0 & X & X & X & X & 0 & X & X & X & 0 & 0 & 1 \\
\hline
X & X & X & X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]
&
\begin{array}{l}
\\ \} \; I \\ \\ \\ \} \; N-I \\ \\ \\ \\ \\ \\ \\ \\ \\ \\
\end{array}
\\
\underbrace{\quad\quad}_{I-1} \quad \underbrace{\quad\quad}_{N-I+1} \quad\quad \underbrace{\quad\quad}_{I-1} \quad\quad\quad\quad &
\end{array}
$$

Se analizarmos corretamente o algorítmo 17 usando a T.S.E simples vemos que para não incorrermos em operações redundantes devemos eliminar os elementos, na coluna I-1, de 1 a I-2, depois de I+1 a N , ou vice-versa, mas deixando para último lugar a eliminação de $\bar{A}(I-1,I-1)$, onde chamamos de eliminação o fato de que durante o pivoteamento anularmos os demais elementos da coluna do pivot em questão, exceto o pivot que será normalizado.

No caso monovariável veremos que a T.S.E por blocos , com seu inconveniente de ter-se que memorizar a sequência, não faz com que efetuemos menos operações, isto é, se usarmos a T.S.E. por blocos corretamente obteremos uma certa quantia de operações que será igual a de uma determinada sequência usando a T.S.E. simples, já em sistemas multivariáveis veremos que tal não ocorre.

Vamos então efetuar as contagens das operações com o uso da T.S.E. simples e da T.S.E. por blocos nos algorītmos 17, 18 e 19.

Para esta contagem vamos supor que a sequência usada para a T.S.E. simples é a de deixar por último a eliminação correta, sendo que a T.S.E. por blocos fará a eliminação de uma coluna inteira no algorītmo 17, e de uma linha inteira nos algorītmos 18 e 19 sem qualquer restrição quanto a ordem. Verifica-se que em ambos os casos o número de operações é o mesmo, e este número está na tabela abaixo, onde N= e P= significam normalização e pivoteamento respectivamente e os colchetes separam as operações por linha e por coluna em $\hat{A}(.,.)$, na ordem dita acima:

<u>TABELA 1</u>: Estágio I no algorītmo 17

$$A : \left[\begin{array}{l} N = [N-I+1] + [N] \\ P = [(N-1)*(N-I+1)] + [(N-I)*N] \end{array}\right.$$

$$B : \left[\begin{array}{l} N = 0 \\ P = 0 \end{array}\right.$$

$$C : \left[\begin{array}{l} N = L \\ P = (N-1)*L \end{array}\right.$$

$$T : \left[\begin{array}{l} N = I-1 \\ P = (N-1)*I \end{array}\right.$$

$$T^{-1} : \left[\begin{array}{l} N = 0 \\ P = (I-1)*N \end{array}\right.$$

TABELA 2: Totais para o algorītmo 17

A : $(N)(N)(N+1) = N^3+N^2$

B : 0

C : N^2L

T : $(N)(N)(N+1)/2-N = (N^3+N^2)/2-N$

T^{-1}: $(N)(N)(N-1)/2 = (N^3-N^2)/2$

TABELA 3: Estāgio I da fase 2 do algorītmo 18

$$A : \left[\begin{array}{ll} N= & [0] + [0] \\ P= & [0] + [0] \end{array}\right.$$

$$B : \left[\begin{array}{ll} N= & 0 \\ P= & 0 \end{array}\right.$$

$$C : \left[\begin{array}{ll} N= & 0 \\ P= & (N-I+1)*L \end{array}\right.$$

$$T : \left[\begin{array}{ll} N= & 0 \\ P= & (N-I+1)*N \end{array}\right.$$

$$T^{-1} : \left[\begin{array}{ll} N= & 0 \\ P= & (N-I+1)*N \end{array}\right.$$

TABELA 4: Totais para a fase 2 do algorītmo 18

A;B : 0

$$C : (L)(N)(N-1)/2 = (LN^2-LN)/2$$

$$T;T^{-1} : (N)(N)(N-1)/2 = (N^3-N^2)/2$$

TABELA 5: Estágio I da fase 1 do algorītmo 19.

$$A : \begin{bmatrix} N = & [N-I+1] + [N] \\ P = & [(N-I+1)*(N-I)] + [(N-I)*N] \end{bmatrix}$$

$$B : \begin{bmatrix} N = & 0 \\ P = & 0 \end{bmatrix}$$

$$C : \begin{bmatrix} N = & L \\ P = & (N-I)*L \end{bmatrix}$$

$$T : \begin{bmatrix} N = & I-1 \\ P = & (N-I)*I \end{bmatrix}$$

$$T^{-1} : \begin{bmatrix} N = & 0 \\ P = & 0 \end{bmatrix}$$

TABELA 6: Estágio I da fase 2 do algorītmo 19

$$A : \begin{bmatrix} N = & [0] + [0] \\ P = & [0] + [(N-I+1)*(I-1)] \end{bmatrix}$$

$$B : \begin{bmatrix} N = & 0 \\ P = & 0 \end{bmatrix}$$

$$C : \begin{bmatrix} N = & 0 \\ P = & (N-I+1)*L \end{bmatrix}$$

$$T : \begin{cases} N = 0 \\ P = (N-I+1)*N \end{cases}$$

$$T^{-1} : \begin{cases} N = 0 \\ P = (N-I+2)*(N-I+1) \end{cases}$$

TABELA 7: Totais do algorītmo 18

A : $(N)(N)(N+1) = N^3+N^2$

B : 0

C : $N^2L + (L)(N)(N-1)/2$

T : N^3-N

T^{-1} : N^3-N^2

TABELA 8: Totais do algorītmo 19

A : $(N)(N)(N+1) = N^3+N^2$

B : 0

C : N^2L

T : $2(N^3-N)/3$

T^{-1} : $(N^3-N)/3$

Podemos observar pelas tabelas 7 e 8 que para A, B e C o volume de operações ē o mesmo, visto que ao invez de obtermos C pelo algorītmo 18 podemos obte-lo pelo produto da matriz de saīda original por T^{-1}, quando esta for efetuada e este produto usa N^2L

operações. Mas de uma forma geral o algorítmo 19 é mais econômico, em termos de operações, do que o algorítmo 18 pois quando do cálculo de T e T^{-1} vemos que o número de operações envolvidas para o cálculo de T e T^{-1} no algorítmo 19 é o mesmo necessário só para o cálculo de T pelo algorítmo 18.

Dadas estas conclusões e dado o fato de que podemos usar, neste caso, a T.S.E. simples, um algorítmo mais fácil de ser programado, já que evita a memorização das operações, é o seguinte:

ALGORÍTMO 20: FCCL3B

.- PRIMEIRA FASE

.- Inicialização

1.- Construir a matriz do sistema H(.,.)

2.- I=1 (I=contador de linhas)

3.- J=N+1 (J=contador de colunas)

.- Procura do melhor pivot

4.- L=valor de K onde ocorre max|H(K,J)| para K=I,I+1,...,N

5.- G=H(L,J)

.- Teste de controlabilidade

6.- se G=0 , o sistema não é controlável, pare
se G≠0 , siga.

.- Colocando o pivot na linha I e coluna J

7.- H(.,.) = EP[H;I,L]

8.- H(.,.) = EM[H;1/G,I]

.- Eliminando a coluna

9.- H(.,.) = EC[H;-H(K,J),I,K] ; K=I+1,...,N

.- Teste de fim de fase

10.- se I=N vá para 12.-
se I≠N siga

11.- J=I ; I=I+1 ; vá para 4.-

.- SEGUNDA FASE

.- Eliminando a linha

12.- H(.,.) = EC[H;H(I,K),K,J] ; K=N,N-1,.-,I

.- Teste de fim de algorītmo

13.- se I=2 pare ; se não siga

14.- I=J ; J=I+1 ; vá para 12.-

Vamos agora abordar as diversas formas multivariáveis; quanto às modificações introduzidas no algorītmo 24, a seguir em relação ao original de Daly, elas são devidas ao fato de que, por motivo de falha na enumeração das linhas em | |, estas não atingem o que é proposto e também, pelo fato de obtermos as formas modificadas.

<u>ALGORĪTMO 21:</u> FCCL1

.- Inicialização

1.- construir a matriz do sistema H(.,.)

2.- I=1 (I=contador de linhas de H(.,.)

3.- J=N+1 (J=contador de colunas de H(.,.)

4.- JB=1 (JB=contador de colunas de B(.,.)

5.- IV=0 (IV=contador de colunas efetivas de B(.,.)

6.- IECL(K)=0 ; IRC(K)=0 ; K=1,2,...,M

7.- IX=1 (IX=1, operando sobre coluna de B(.,.))

8.- JX=0 (JX=contador para IRC(.))

.- Procurando o melhor pivot

9.- L=valor de K onde ocorre max|H(K,J)|
K=I,I+1,...,N

10.- G=H(L,J)

.- Teste de controlabilidade

11.- se G≠0 vá para 17.-
se G=0 siga

12.- se JB=M o sistema não é controlável, pare.
se não siga

13.- JB=JB+1

14.- se IX=1 , J=N+JB e vá para 9.-
se não siga

.- Índices IRC e IECL

15.- IRC(JB-1) = JX

16.- IECL(JB-1)=J ; J=N+JB ; vá para 7.-

17.- se IX=1 ; IV=N+1 ; IX=0 e siga
se não siga somente

18.- JX=JX+1

.- Colocando o pivot na linha I e coluna J

19.- H(.,.) = EP[H;L,I]

20.- H(.,.) = EM[H;1/G,I]

.- Eliminação por colunas

21.- H(.,.) = EC[H;-H(1,J),I,1;...;-H(I-1,J),I,I-1;
-H(I+1,J),I,I+1;...;-H(N,J),I,N]

.- Teste de fim de algorítmo

22.- se I=N vá para 24.-
se I≠N siga

23.- J=I ; I=I+1 ; vá para 9.-

.- Finalização

24.- IRC(JB) = JX

25.- IECL(JB) = N

26.- o sistema é controlável por IV entradas.

Como podemos observar a obtenção de IECL(.) e de IRC(.) é concomitante com a obtenção da forma canônica.

<u>ALGORÍTMO 22:</u> FCCL3A

.- PRIMEIRA FASE

1.- FCCL1

.- SEGUNDA FASE

Obter FCOL1 com C1A ou

2.- J=I-1 (da primeira fase I=N)

.- Verificando se estamos sobre as linhas estruturais

3.- se H(I,J)=0 vã para 5.-
se não siga

4.- H(.,.) = EC[H;H(I,I),I,J;...;H(I,N),N,J]

5.- se I=2 ; vã para 7.-
se não siga

6.- I=I-1 ; vã para 2.-

.- Índices estruturais

7.- IECL(K) = IECL(K)-IRC(K)+1 se IRC(K)≠0
IECL(K) = 0 se IRC(K)=0
K=1,2,...,M

ALGORĪTMO 23: FCCL3B

.- PRIMEIRA FASE

algorītmo 21 com as seguintes alterações

16.- IECL(JB-1) = I-JX ; vã para 7.-

21.- H(.,.) = EC[H;-H(I+1,J),I,I+1;...;H(N,J),I,N]

25.- IECL(JB) = N-JX+1

.- SEGUNDA FASE

27.- J=I-1 (da primeira fase veio I=N)

.- Verificando se estamos sobre as linhas estruturais.

28.- se H(I,J)=0 vã para 30.-
se não siga

29.- H(.,.) = EC[-H;H(I,I),I,J;...;H(I,N),N,J]

30.- se I=2 ; pare
se não siga

31.- I=I-1 ; vã para 27.-

Para os casos multivariáveis, a obtenção de uma tabela exata só é possível se conhecermos a sequência dos índices relativos e mesmo assim, com alguma atenção, podemos notar que aplicando a T.S.E simples ou a T.S.E por blocos obteremos o mesmo volume de operações, desde que seja respeitada a melhor sequência para a eliminação, ou seja, deixando por último a tarefa de eliminar H(I-1,I-1), por colunas, ou H(I,I) por linhas.

Obtidas as formas canônicas FCCL1 e FCCL3 , vamos tratar de obter FCCL2, FCCL4, FCCB1 e FCCB2 iniciando com um algorítmo para obter as formas FCCL2 e FCCB1 e a seguir propondo dois outros algorítmos para as formas FCCL4 e FCCB2.

<u>ALGORÍTMO 24:</u> FCCB1 e FCCL2

.- Inicialização

1.- construir a matriz do sistema H(.,.)

2.- I=1 (I=indicador de linha em H(.,.))

3.- J=N+1 (J=indicador de coluna em H(.,.))

4.- JB=1 (JB=indicador de coluna de B(.,.))

5.- IV=0 (IV=contador das colunas efetivas de B, ou seja, colunas em que IRC será ≠ 0)

6.- IECB(K)=0 e IRC(K)=0
para K=1,2,...,M

7.- IX=1 (Quando IX=1 estaremos operando sobre alguma coluna de B(.,.))

8.- JX=0 (JX=contador auxiliar)

.- Teste de controlabilidade

9.- se IX=1 vá para 15.-
se não siga

10.- se JB=M faça JX=0 e siga
se não siga

11.- DO 12.- JB=JX+1,M

12.- se IRC(JB)>0 vá para 15.-
se não continue

13.- se JX=0 o sistema não é controlável, pare
se não siga

14.- JX=0 e vá para 11.-

.- Procura do melhor pivot

15.- L= valor de K onde ocorre max|H(K,J)|
K=I,I+1,...,N

16.- G=H(L,J)

.- Teste de dependência linear

17.- se G≠0 vá para 21.-
se não siga

18.- se IX=0 vá para 20.-
se não siga

19.- se J=N+M e IV=0 , sistema não controlável, pare
se J=N+M faça IX=0 ; J=1 ; vá para 11.-
se não faça J=J+1 e vá para 15.-

.- Índices IECL e IRC

20.- IECL(JB)=J
IRC(JB)=-IRC(JB)
J=J+1 ; JX=JB ; e vá para 10.-

21.- se IX=0 vá para 23.-
se não siga

22.- IV=IV+1 ; IRC(J-N)=1 ; vá para 24.-

.- Contando IRC

23.- IRC(JB)=IRC(JB)+1

.- Colocando o pivot na linha I e coluna J

24.- H(.,.) = EP[H;I,L]

25.- H(.,.) = EM[H;1/G;I]

.- Eliminação por colunas

26.- H(.,.) = EC[H;-H(1,J),I,1;,,,;-H(I-1,J),I,I-1;
-H(I+1,J),I,I+1;...;-H(N,J),I,N]

.- Teste de fim de algorítmo

27.- se I=N vá para 30.-
se I≠N siga

28.- I=I+1 ; JX=JB

29.- se J=N+M J=1 ; IX=0 ; vã para 9.-
se não J=J+1 e vã para 9.-

.- Finalização para FCCB1

30.- DO 31.- JB=JX+1,M

31.- se IRC(JX)>0 vã para 34.-
se não continue

32.- se JX=0 vã para 37.-
se não siga

33.- JX=0 e vã para 30.-

34.- J=J+1

35.- IRC(JB)=-IRC(JB)
IECL(JB)=J

36.- se J=M faça JX=0 e vã para 30.-
se não JX=JB e vã para 30.-

37.- DO 37.- JB=1,M

38.- IRC(JB)=-IRC(JB)

39.- O sistema é controlável por IV entradas.

.- Finalização para FCCL2

40.- reordenar H(.,.)

41.- rearrumar IECL

Para obtermos FCCL2 devemos rearrumar, como se estivessemos usando transformações de similaridade elementares do tipo permutação, o que foi então obtido, isto é, FCCB1. Esta tarefa pode ser efetuada de dois modos:

1.- rearrumando os "pivots" '1' de forma a se localizarem na subdiagonal inferior com uma conveniente localização das linhas não nulas da matriz de entrada.

2.- rearrumando a matriz do sistema com uma transformação de similaridade sugerida pelo agrupamento dos elementos de mesmo valor, na mesma forma em que forem encontrados de um vetor auxiliar IORD(.) que pode ser obtido alterando os passos 29.- e 34.-

do algorítmo 24 para:

29.- se J=N+M ; J=1 ; JX=0 ; vá para 9.-
se não IORD(J)=JB ; J=J+1 ; vá para 9.-

34.- J=J+1 ; IORD(J)=JB

Ex: seja um sistema de índices

IRC(1) = 3	IECL(1) = 6
IRC(2) = 1	IECL(2) = 2
IRC(3) = 0	IECL(3) = 0
IRC(4) = 2	IECL(4) = 5

IORD(1) = 1	IORD(2) = 2	IORD(3) = 4
IORD(4) = 1		IORD(5) = 4
IORD(6) = 1		

O rearranjo nos levará a reagrupar as linhas de $\hat{A}$, $\hat{B}$, $\hat{T}$ na sequência 1, 4, 6, 2, 3, 5 e consequentemente as colunas de $\hat{A}$, $\hat{C}$ e $\hat{T}^{-1}$ na mesma sequência uma vez que este tipo de matriz de transformação , P , é de tal forma que $P*P^T=I$.

O par $[\hat{A} \vdots \hat{B}]$ antes de ser rearrumado teria a forma:

$$\left[\begin{array}{cccccc|cccc} 0 & X & 0 & 0 & X & X & 1 & 0 & X & 0 \\ 0 & X & 0 & 0 & X & X & 0 & 1 & X & 0 \\ 0 & X & 0 & 0 & X & X & 0 & 0 & 0 & 1 \\ 1 & X & 0 & 0 & X & X & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & X & X & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & X & X & 0 & 0 & 0 & 0 \end{array}\right] = [ABC1 \vdots BBC1]$$

após a rearrumação teríamos.

$$\left|\begin{array}{cccccc|cccc} 0 & 0 & X & X & 0 & X & 1 & 0 & X & 0 \\ 1 & 0 & X & X & 0 & X & 0 & 0 & 0 & 0 \\ 0 & 1 & X & 0 & 0 & X & 0 & 0 & 0 & 0 \\ 0 & 0 & X & X & 0 & X & 0 & 1 & X & 0 \\ 0 & 0 & X & X & 0 & X & 0 & 0 & 0 & 1 \\ 0 & 0 & X & 0 & 1 & X & 0 & 0 & 0 & 0 \end{array}\right| = |ALC2 \vdots BLC2|$$

Nos dois modos citados podemos usar do conhecimento da estrutura das formas canônicas e através da procura ordenada do "pivot" '1' rearrumar de maneira adequada a matriz do sistema simulando IORD(.) na própria estrutura de FCCB1.

Com isso, o primeiro modo é mais "econômico" do ponto de vista de memória tanto para dados quanto para programação, isto é, na memória de dados deveríamos ter um vetor de ordem N, IORD, e uma matriz quadrada de ordem N que seria a transformação auxiliar e na memória para programação usaríamos um programa de aproximadamente a mesma parte do programa em que simulamos IORD.

<u>ALGORÍTMO 25</u>: Reordenação simulando IORD(.)

.- Inicialização

1.- J=0 (J=contador de colunas de H(.,.))

2.- JB=0 (JB=contador de colunas de B(.,.))

.- Teste de sequência ou fim de algorítmo

3.- se JB=M pare
se não siga

4.- JX=JB+1 ; DO 5.- JB=JX,M

5.- se IRC(JB)>0 vá para 7.-
se não continue

6.- PARE

.- Simulando IORD

7.- DO 8.- L = J+1 , N

8.- se H(L,N+JB) = L vá para 9.-
se não continue

9.- J=J+L ; H(.,.) = EP[H;L,J]

.- Atualizando os índices estruturais

10.- DO 11.- JX=JB,M

11.- se IECL(JX)$\neq$J continue
se não IECL(JX)=L e vá para 12.-

12.- se IECL(JB)=L
faça IECL(JB)=J e vá para 3.-
se não siga

13.- DO 14.- L=N,J+1,-1

14.- se H(L,J)=1 vá para 9.-
se não continue

Neste caso para que possamos trocar a T.S.E. por blocos pela T.S.E. simples devemos observar que como no caso os '1' não ocuparão a subdiagonal principal inferior, o ultimo elemento que deve ser eliminado não é mais o H(I-1,I-1), pois este elemento não pertence à mesma coluna em que estamos operando, mas sim o elemento H(J,J); lembrar que em todos os casos anteriores H(I-1,I-1) = = H(J,J) pois J=I-1. Para o algorítmo 24, assim como na primeira fase dos algorítmos seguintes, usar a T.S.E. por blocos ou a simples vai envolver o mesmo número de operações.

Para finalmente obtermos FCCL4 e FCCB2 vamos fazer uso dos seguintes algorítmos:

<u>ALGORÍTMO 26</u>: FCCL4A, FCCB2A

.- PRIMEIRA FASE: obter FCCB1 eliminando em 20.- IECL (JB)=J e colocando em 22.- IECL(JB)=IV

.- SEGUNDA FASE

2.- JX=I-1 (da primeira fase I=N)

.- Encontrando o '1' pivot

3.- DO 4.- J=1,JX

4.- se H(I,J)=1 vá para 5.-
se não siga

5.- JX=J-1

.- Eliminação por linhas

6.- H(.,.) = EC[H;H(I,J+1),J+1,J;...;H(I,N),N,J]

7.- se I=M+1 vá para 9.-
se não siga

8.- I=I-1 vá para 3.-

9.- reordene como no algorítmo 25 para o caso observável com C3A para obter FCCL4

10.- se usarmos C3A e rearrumarmos de acordo com um algorítmo para obter a forma FCOB1 obteremos FCCB2

A segunda fase deste algorítmo 26, como a segunda fase do algorítmo 27 a seguir, pode ser feita usando C3A e obtendo a forma FCOL2; obtendo com isso FCCL4 , no caso de querermos FCCB2 usaríamos C3A para obter com a matriz do sistema a forma FCOB1; no entanto, devemos notar que obter, por exemplo, FCCB2 conforme esta segunda fase proposta acima ou através da que se encontra no passo 10.- do algorítmo 26, tem uma diferença fundamental que reside no fato de que no passo 10.- do algorítmo 26 não há necessidade de efetuarmos repetidas operações de eliminação por linhas, pois esta já foi efetuada, ao passo que com C3A para obtermos FCOB1 estaremos efetuando várias operações deste tipo, além das operações de permutação, o que implica em operações redundantes e portanto na possibilidade de diminuição da precisão do resultado obtido, assim como implica em um tempo maior de computação.

<u>ALGORÍTMO 27:</u> FCCL4, FCCB2

.- PRIMEIRA FASE

Algorítmo 24 com as seguintes alterações:

20.- elimina IECL(JB)=J

22.- IECL(JB)=IV ; IV=IV+1 ; etc.

26.- H(.,.) = EC[H;-H(I+1,J),I,I+1;...;-H(N,J),I,N] e terminando em 39.-

.- SEGUNDA FASE

Como na segunda fase do algorítmo 26

Para a segunda fase tanto do algorítmo 26, quanto para o algorítmo 27, o método que mostrou envolver menos operações foi uma T.S.E. por blocos, por mais que se escolhesse uma T.S.E. simples conveniente.

Com o objetivo de ilustrarmos as diversas tentativas empregadas para ser encontrada a melhor sequência de operações, no caso da T.S.E. simples, ou o melhor bloco, no caso da T.S.E. por blocos, listamos aqui uma série de casos, com a observação restrita apenas a matriz de estado em um caso de índices relativos de controlabilidade 3, 1 e 2:

(I) indicará o I-ésimo elemento a ser eliminado por T.S.E. simples; no caso da sequência do sistema usado os elementos 'Y' são elementos que deveriam ser nulos mas para este tipo de construção não serão nulos conforme explicaremos mais tarde

X	X	X	X	X	X	Simples	: 67 operações
Y	X	Y	Y	X	X		
Y	X	X	X	X	X	Bloco-Total*	: 54 operações
1	(5)	(4)	(3)	(2)	(1)		
0	0	1	(8)	(7)	(6)	Bloco-Linhas**	: 61 operações
0	0	0	1	(10)	(9)		

X	X	X	X	X	X	Simples	: 61 operações
Y	X	Y	Y	X	X		
Y	X	X	X	X	X	Bloco-Total	: 54 operações
1	(10)	(9)	(8)	(7)	(6)		
0	0	1	(5)	(4)	(3)	Bloco-Linhas	: 57 operações
0	0	0	1	(2)	(1)		

X	X	X	X	X	X	Simples : 57 operações
Y	X	Y	Y	X	X	
Y	X	X	X	X	X	Bloco-Total : 54 operações
1	(7)	(6)	(10)	(9)	(8)	
0	0	1	(3)	(5)	(4)	Bloco-Linhas : 57 operações
0	00	0	1	(1)	(2)	

X	X	X	X	X	X	Simples : 54 operações
Y	X	Y	Y	X	X	
Y	X	X	X	X	X	Bloco-Total : 54 operações
1	(7)	(6)	(10)	(9)	(8)	
0	0	1	(1)	(3)	(2)	Bloco-Linhas : 54 operações
0	0	0	1	(4)	(5)	

X	X	X	X	X	X	
Y	X	Y	Y	X	X	Simples : 61 operações
Y	X	X	X	X	X	
1	(10)	(9)	(8)	(7)	(6)	
0	0	1	(5)	(4)	(3)	Bloco-Grupo*** : 54 operações
0	0	0	1	(2)	(1)	

X	X	X	X	X	X	
Y	X	Y	Y	X	X	Simples : 57 operações
Y	X	X	X	X	X	
1	(10)	(9)	(8)	(7)	(6)	
0	0	1	(1)	(2)	(3)	Bloco-Grupo : 52 operações
0	0	0	1	(4)	(5)	

Melhor configuração

X	X	X	X	X	X	Simples : 54 operações
Y	X	Y	Y	X	X	
Y	X	X	X	X	X	
1	(7)	(6)	(10)	(9)	(8)	Bloco-Grupo : 52 operações
0	0	1	(1)	(3)	(2)	
0	0	0	1	(4)	(5)	

* elimina todos (I) por colunas na sequência apresentada e depois completa a transformação de similaridade por linhas.

** elimina os (I) de cada linha

*** elimina os (I) de cada grupo de linhas separadas pelas linhas tracejadas.

Assim sendo, se optarmos por usar a T.S.E. simples, esta deverá ser feita de modo a que na linha I que estamos eliminando, o último seja H(I,I); se optarmos pela T.S.E. por blocos, o bloco será um grupo de linhas escolhidas de maneira adequada e começaremos sempre da primeira linha de cada bloco.

CAPÍTULO IV

Neste capítulo serão apresentadas algumas formas canônicas alternativas e observadas algumas características destas bem como das de Luenberger e de Bingulac.

De início vamos estabelecer a localização dos parâmetros nulos em FCCL2 uma vez que os de FCCL1 são localizáveis de imediato.

TEOREMA 1 Parâmetros nulos de ALC2

Seja IRC(I)xIRC(J) a dimensão de um dos blocos de ALC2, então os r+1 primeiros elementos da coluna característica serão não nulos se J>I; no caso J≤I os elementos da coluna característica que são não nulos serão os r primeiros, onde r=min [IRC(I), IRC(J)].

PROVA: o vetor A(.,.)**IRC(I)*B(.,I) depende dos vetores A(.,.)**T*B(.,J) onde T assume os valores de 0 à IRC(I)-1 se IRC(J) IRC(I) , e assume o valor IRC(I) se além de IRC(J)> IRC(I) também tivermos J>I . Nos casos onde IRC(I)>IRC(J) teremos T=0,1,...,IRC(J)-1 já que os vetores com T=IRC(J),...,IRC(I) -1 não pertencem a CB2.

Por exemplo um sistema de índices relativos 3,1,5 fica na forma:

$$\left[\begin{array}{ccc|c|ccccc}
0 & 0 & X & X & 0 & 0 & 0 & 0 & X \\
1 & 0 & X & X & 0 & 0 & 0 & 0 & X \\
0 & 1 & X & 0 & 0 & 0 & 0 & 0 & X \\
\hline
0 & 0 & X & X & 0 & 0 & 0 & 0 & X \\
\hline
0 & 0 & X & X & 0 & 0 & 0 & 0 & X \\
0 & 0 & X & 0 & 1 & 0 & 0 & 0 & X \\
0 & 0 & X & 0 & 0 & 1 & 0 & 0 & X \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & X \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & X
\end{array}\right] = \text{ALC2}$$

para o mesmo sistema também temos:

$$\begin{bmatrix} 0 & X & 0 & 0 & 0 & X & 0 & 0 & X \\ 0 & X & 0 & 0 & 0 & X & 0 & 0 & X \\ 0 & X & 0 & 0 & 0 & X & 0 & 0 & X \\ 1 & X & 0 & 0 & 0 & X & 0 & 0 & X \\ 0 & 0 & 1 & 0 & 0 & X & 0 & 0 & X \\ 0 & 0 & 0 & 1 & 0 & X & 0 & 0 & X \\ 0 & 0 & 0 & 0 & 1 & X & 0 & 0 & X \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & X \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & X \end{bmatrix} = ABC1$$

e podemos ver que cada coluna estrutural tem o mesmo número de parâmetros não nulos, ou seja, para a coluna IECL(1)=3 temos 7 parâmetros não nulos da mesma forma que para a coluna IECB(1)=6 , e assim por diante.

Vamos agora provar a existência de FCCL3 e FCCL4 ; para isso vamos construir as seguintes matrizes transformação de similaridade que nos levam às formas modificadas de Luenberger e cuja valia não reside neste fato apenas mas no fato que desta forma poderemos observar os parâmetros nulos de ALC3 e ALC4.

<u>ALGORÍTMO 1:</u> OL1A[ALC1,C1A]

```
1.-   DO   3.-     I=K,1,-1
2.-   DO   3.-     T=0,IRC(I)-1
3.-   OL1A(IECL(I)-T,.)=C1A(I,.)*ALC1(.,.)**T
```

<u>ALGORÍTMO 2:</u> OL2A[ALC2,C2A]

```
1.-   DO   3.-     I=K,1,-1
2.-   DO   3.-     T=0,IRC(I)-1
3.-   OL2A(IECL(I)-T,.)=C2A(I,.)*ALC2(.,.)**T
```

O raciocínio que devemos levar em conta para provar a não singularidade de CL3 e CL4 e mostrarmos as formas de ALC3 e ALC4 será feito através do seguinte exemplo:

Vamos supor que temos um sistema com índices relativos 3,1 e 2 para FCCL1 ; isto nos daria a seguinte forma de ALC1 e OL1A

$$\left[\begin{array}{ccc|c|cc} 0 & 0 & X & X & 0 & X \\ 1 & 0 & X & X & 0 & X \\ 0 & 1 & X & X & 0 & X \\ \hline 0 & 0 & 0 & X & 0 & X \\ \hline 0 & 0 & 0 & 0 & 0 & X \\ 0 & 0 & 0 & 0 & 1 & X \end{array}\right] = \mathrm{ALC1}$$

$$\left[\begin{array}{ccc|c|cc} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] = \mathrm{C1A}$$

$$\left[\begin{array}{c} \mathrm{C1A(1,.)*ALC1(.,.)**2} \\ \mathrm{C1A(1,.)*ALC1(.,.)} \\ \mathrm{C1A(1,.)} \\ \hline \mathrm{C1A(2,.)} \\ \hline \mathrm{C1A(3,.)*ALC1(.,.)} \\ \mathrm{C1A(3,.)} \end{array}\right] = \left[\begin{array}{ccc|c|cc} 1 & X & X & X & 0 & X \\ 0 & 1 & X & X & 0 & X \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & X \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] = \mathrm{OL1A}$$

Se obtivermos uma forma observável entre ALC1 e C1A , como em um algorítmo semelhante a FCOL3 , obteremos a forma:

$$\left[\begin{array}{ccc|c|cc} X & X & X & X & X & X \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & X & 0 & X \\ \hline 0 & 0 & 0 & 0 & X & X \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right] = ALC3'$$

que é uma forma modificada da forma original a qual pode ser obtida se usarmos uma outra transformação de similaridade dada pelos seguintes algorítmos:

<u>ALGORÍTMO 3:</u> P1[OL1A]

```
1.-   DO   2.-   T=1,IRC(1)
2.-   P1(T,.)=IDN(IECL(1)+1-T,.)
3.-   DO   5.-   I=2,K
4.-   DO   5.-   T=1,IRC(I)
5.-   P1(T+IECL(I-1),.)=IDN(IECL(I)+1-T,.)
```

no caso FCCL3 e para a forma FCCL4 teremos:

<u>ALGORÍTMO 4:</u> P2[OL2A]

```
1.-   DO   2.-   T=1,IRC(1)
2.-   P2(T,.)=IDN(IECL(1)+1-T,.)
3.-   DO   5.-   I=2,K
4.-   DO   5.-   T=1,IRC(I)
5.-   P2(T+IECL(I-1),.)=IDN(IECL(I)+1-T,.)
```

Voltando ao nosso exemplo anterior teríamos para P1 a forma:

$$\left[\begin{array}{ccc|c|cc} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right] = P1$$

$$P1*ALC3'*P1IN = P1*ALC3'*P1 = \left[\begin{array}{ccc|c|cc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ X & X & X & X & X & X \\ \hline 0 & 0 & 0 & X & X & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & X & X \end{array}\right] = ALC3$$

TEOREMA 2

Se os índices relativos de controlabilidade forem apresentados em ordem não decrescente não haverá elementos 'Y' e 'Z' de BLC3.

Se os índices relativos de controlabilidade forem apresentados em ordem decrescente mas em uma sequência tal que se I<J e IRC(J)<IRC(I) mas IRC(J)+1 = IRC(I) então os elementos 'Y' continuarão nulos mas elementos 'Z' poderão ser não nulos.

Prova: pelos algorítmo 3 e 1 vemos que P1(.,.)*OL1A (.,.) = CL3A e podemos verificar que CL3A assume a seguinte forma:

$$\left[\begin{array}{c:c:c}
00 \ldots 01 & 00 \ldots 00 & 00 \ldots 00 \\
00 \ldots 1X & 00 \ldots 0X & 00 \ldots 0X \\
\cdots\cdots & \cdots\cdots & \cdots\cdots \\
01 \ldots XX & XX \ldots XX & 0X \ldots XX \\
1X \ldots XX & XX \ldots XX & XX \ldots XX \\
\hdashline
00 \ldots 00 & 00 \ldots 01 & 00 \ldots 00 \\
00 \ldots 00 & 00 \ldots 1X & 00 \ldots 0X \\
\cdots\cdots & \cdots\cdots & \cdots\cdots \\
00 \ldots 00 & 01 \ldots XX & 00 \ldots XX \\
00 \ldots 00 & 1X \ldots XX & 0X \ldots XX \\
\hdashline
\cdots\cdots & \cdots\cdots & \cdots\cdots \\
\hdashline
00 \ldots 00 & 00 \ldots 00 & 00 \ldots 01 \\
00 \ldots 00 & 00 \ldots 00 & 00 \ldots 1X \\
\cdots\cdots & \cdots\cdots & \cdots\cdots \\
00 \ldots 00 & 00 \ldots 00 & 01 \ldots XX \\
00 \ldots 00 & 00 \ldots 00 & 1X \ldots XX
\end{array}\right]$$

recordando agora que BLC3 = CL3A*BLC1 ou seja BLC3 é formado pelas colunas IECL(I)-IRC(I)+1 , I=1,2,...,K ; isto implica que se tomarmos BLC3 na seguinte forma

$$\left[\begin{array}{c:c}
0Y \ldots Y & X \ldots X \\
0Y \ldots Y & X \ldots X \\
0Y \ldots Y & X \ldots X \\
\cdots\cdots & \cdots\cdots \\
1Z \ldots Z & X \ldots X \\
\hdashline
00 \ldots Y & X \ldots X \\
00 \ldots Y & X \ldots X \\
00 \ldots Y & X \ldots X \\
\cdots\cdots & \cdots\cdots \\
01 \ldots Z & X \ldots X \\
\hdashline
\cdots\cdots & \cdots\cdots \\
\hdashline
00 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
\cdots\cdots & \cdots\cdots \\
00 \ldots 1 & X \ldots X
\end{array}\right]
=
\left[\begin{array}{c:c}
 & X \ldots X \\
 & X \ldots X \\
B[1] & X \ldots X \\
 & \cdots\cdots \\
 & X \ldots X \\
\hdashline
 & X \ldots X \\
 & X \ldots X \\
B[2] & X \ldots X \\
 & \cdots\cdots \\
 & X \ldots X \\
\hdashline
\cdots\cdots & \cdots\cdots \\
\hdashline
 & X \ldots X \\
B[K] & X \ldots X \\
 & X \ldots X \\
 & \cdots\cdots \\
 & X \ldots X
\end{array}\right]
= BLC3$$

a coluna J de B[I], J>I , e a primeira coluna do bloco I,J de CL3A e se observarmos veremos que este bloco tem um conjunto de diagonais transversas a principal com elementos não nulos, o que indica que a primeira coluna deste bloco só passará a ter elemento não nulo se IRC(J)$\geq$IRC(I) uma vez que a última coluna deste bloco tem os seus últimos IRC(I)-1 elementos não nulos e isto prova a primeira parte do teorema pois se para J>I tivermos índices não decrescentes, IRC(J)>IRC(I) , então não teremos parâmetros não nulos na primeira coluna do bloco I,J de CL3A o que implica em não ter 'Y' e 'Z' na coluna J de B[I] . Passaremos a ter um elemento não nulo, na última linha da primeira coluna do bloco em questão, se IRC(J) = IRC(I)-1 o que prova a segunda parte.

Para exemplificar este teorema tomemos os seguintes blocos:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & X \\ 0 & 0 & 0 & X & X \\ 0 & 0 & X & X & X \end{bmatrix} (4\times5) \rightarrow \text{primeira coluna} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & X \\ 0 & X & X \\ X & X & X \end{bmatrix} (4\times3) \rightarrow \text{primeira coluna} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ X \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ Z \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 \\ 0 & X \\ X & X \\ X & X \end{bmatrix} (4\times2) \rightarrow \text{primeira coluna} = \begin{bmatrix} 0 \\ 0 \\ X \\ X \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ Y \\ Z \end{bmatrix}$$

Sempre a primeira coluna dos blocos I,J de CL3A terá os r primeiros elementos nulos, onde:

r = min(IRC(I), IRC(J))

Podemos agora determinar quais os elementos 'X' que não serão nulos nas formas ALC3 e ALC4 , uma vez que nenhum dos 'X' de ALC1 vai ser nula apenas por depender da sequência dos índices relativos de controlabilidade.

TEOREMA 3

Na linha estrutural de cada bloco de ALC4 os r primeiros elementos são não nulos assim como esta mesma quantidade de elementos não nulos ocorre na linha estrutural dos blocos de ALC3 que tenha 'X' onde r=min(IRC(I), IRC(J) para o bloco I,J.

PROVA: torna-se evidente se construirmos as matrizes ALC4 e ALC3 de acordo com o procedimento proposto para provar suas respectivas existências.

A forma alternativa que apresentaremos a seguir será parecida com a FCCL3 com a vantagem de que a nova matriz de entrada será como BLC3 com 'Y' e 'Z' nulos, mas com uma desvantagem na forma das submatrizes acima da diagonal principal. No entanto convém ressaltar que esta desvantagem apresentar-se-á irrelevante para os problemas a que se destinará mais tarde. Para alcançarmos està forma alternativa usaremos os seguintes algorítmos

ALGORÍTMO 5: ACL1A[ACL1]

1.- ACL1A(.,.)=ACL1(.,.)

2.- J=0 ; DO 5.- KA=1,K-1

3.- DO 5.- KB=1,IRC(KA) ; I=I+1

4.- DO 5.- KC=IECL(KA)+1,N

5.- ACL1A(.,.) = 0

com este algorítmo ACL1A é formado apenas pelas submatrizes da diagonal principal de ALC1; continuando agora teremos:

SAC15[A,B] = SAC[ALC1A,C1A]

CS1[A,B] = OL1[SAC1S,C1A]

FORMA CANÔNICA CONTROLÁVEL AUXILIAR I FCCS1

$$\left[\begin{array}{c|c|c} CS1 & 0 & 0 \\ \hline 0 & IDL & 0 \\ \hline 0 & 0 & IDN \end{array}\right] *HCL1* \left[\begin{array}{c|c|c} CS1IN & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right] = \left[\begin{array}{c|c|c} ASC1 & BSC1 & TSC1 \\ \hline CSC1 & D & 0 \\ \hline TSC1IN & 0 & 0 \end{array}\right] = HSC1$$

onde [ASC1] [BSC1] é :

$$\left[\begin{array}{c|c|c|c}
010 \ldots 0 & 000 \ldots X & \ldots & 000 \ldots X \\
001 \ldots 0 & 000 \ldots X & \ldots & 000 \ldots X \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
000 \ldots 1 & 000 \ldots X & \ldots & 000 \ldots X \\
XXX \ldots X & 000 \ldots X & \ldots & 000 \ldots X \\
\hline
000 \ldots 0 & 010 \ldots 0 & \ldots & 000 \ldots X \\
000 \ldots 0 & 001 \ldots 0 & \ldots & 000 \ldots X \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
000 \ldots 0 & 000 \ldots 1 & \ldots & 000 \ldots X \\
000 \ldots 0 & XXX \ldots X & \ldots & 000 \ldots X \\
\hline
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
\hline
000 \ldots 0 & 000 \ldots 0 & \ldots & 010 \ldots 0 \\
000 \ldots 0 & 000 \ldots 0 & \ldots & 001 \ldots 0 \\
\cdots\cdots & \cdots\cdots & \cdots & \cdots\cdots \\
000 \ldots 0 & 000 \ldots 0 & \ldots & 000 \ldots 1 \\
000 \ldots 0 & 000 \ldots 0 & \ldots & XXX \ldots X
\end{array}\right]
\quad
\left[\begin{array}{c|c}
00 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
\cdots\cdots & \cdots\cdots \\
00 \ldots 0 & X \ldots X \\
10 \ldots 0 & X \ldots X \\
 & \\
00 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
\cdots\cdots & \cdots\cdots \\
00 \ldots 0 & X \ldots X \\
01 \ldots 0 & X \ldots X \\
\hline
\cdots\cdots & \cdots\cdots \\
\hline
00 \ldots 0 & X \ldots X \\
00 \ldots 0 & X \ldots X \\
\cdots\cdots & \cdots\cdots \\
00 \ldots 0 & X \ldots X \\
00 \ldots 1 & X \ldots X
\end{array}\right]$$

Para finalizar devemos fazer algumas apreciações sobre a flexibilidade de algumas formas canônicas, e essa flexibilidade recai principalmente sobre ALC4 ou AL04. Podemos observar que se os índices relativos de controlabilidade ou de observabilidade, forem decrescentes, crescentes ou sem ordem alguma o número de 'X' nulo será sempre o mesmo, logo se para um grupo de índices nós sempre agrupamos os blocos da diagonal principal numa mesma sequência a forma de ALC4 ou AL04 , será sempre a mesma, isto é, será invariante independentemente da sequência real dos índices relativos; esta sequência terá influência apenas em BLC4 ou CL04. Assim sendo se tomarmos os seguintes algorítmos e matrizes:

ALGORÍTMO 6: C2S

1.- C2S(I,.)=CL2IN(IECL(J),.) ; I=1,2,...M e para uma sequência de J entre 1 e M.

ALGORÍTMO 7: C2SA

1.- C2SA(I,.)=IDN(IECL(J),.) ; I=1,2,...,M e para uma sequência de J entre 1 e M.

ALGORÍTMO 8: C2SB

1.- C2SB(I,.)=IDN(IECL(J)-IRC(J)*1 ; 1 ; J=1,2,...M e para uma sequência de J entre 1 e M.

SDN = matriz primeira subdiagonal superior de ordem n, por exemplo:

$$SD4 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

SAC2S = SAC[A,C2S]

SA2SA = SAC ALC[ALC2, C2SA]

SA2SB = SAC[SDN, C2SB]

CS2 = OL2[SAC2S, C2S]

CS2A = OL2[SA2SA, C2SA]

CS2B = OL2[SA2SB, C2SB]

poderemos obter as seguintes formas

FORMA CANÔNICA CONTROLÁVEL AUXILIAR II FCCS2

$$\left[\begin{array}{c|c|c} CS2 & 0 & 0 \\ \hline 0 & IDL & 0 \\ \hline 0 & 0 & IDN \end{array}\right] * \left[\begin{array}{c|c|c} A & B & IDN \\ \hline C & D & 0 \\ \hline IDN & 0 & 0 \end{array}\right] * \left[\begin{array}{c|c|c} CS2IN & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right] = HSC2 =$$

$$= \left[\begin{array}{c|c|c} CS2A & 0 & 0 \\ \hline 0 & IDL & 0 \\ \hline 0 & 0 & IDN \end{array}\right] *HLC2* \left[\begin{array}{c|c|c} CS2AIN & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right] = \left[\begin{array}{c|c|c} ASC2 & BSC2 & TSC2 \\ \hline CSC2 & D & 0 \\ \hline TSC2IN & 0 & 0 \end{array}\right] =$$

$$= \left[\begin{array}{c|c|c} CS2B & 0 & 0 \\ \hline 0 & IDL & 0 \\ \hline 0 & 0 & IDN \end{array}\right] *HLC4* \left[\begin{array}{c|c|c} CS2BIN & 0 & 0 \\ \hline 0 & IDM & 0 \\ \hline 0 & 0 & IDN \end{array}\right]$$

onde os índices estruturais de controlabilidade agora serão:

IECS(I) = IECL(J) ; I=1,2,...,M e para uma sequência de J entre 1 e M.

Se esta sequência de J for tal que sempre mantenha uma mesma sequência de índices relativos, para um conjunto de índices relativos, não importando a real sequência, então para o mesmo conjunto de índices relativos, a forma de ASC2 é invariante.

ASC2 tem uma forma parecida com ALC4 exceto pelos novos índices estruturais; quanto a BSC2 é obtido de BLC4 reajustando suas linhas não nulas de acordo com os novos índices de estrutura, por exemplo:

Vamos supor que para as realizações que tenham índices {1,2,3}, J será escolhido de tal modo a colocar os índices relativos de controlabilidade em ordem crescente, com isso podemos ter:

<u>CASO 1</u>:

IRC(1)=1 ; IRC(2)=2 ; IRC(3)=3

IECL(1)=1 ; IECL(2) ; IECL(3)=6

$$\left[\begin{array}{c:c:c} X & X0 & X00 \\ \hdashline 0 & 01 & 000 \\ X & XX & XX0 \\ \hdashline 0 & 00 & 010 \\ 0 & 00 & 001 \\ X & XX & XXX \end{array}\right] \left[\begin{array}{c} 100 \\ 000 \\ 010 \\ 000 \\ 000 \\ 001 \end{array}\right] = [ALC4]\ [BLC4] = [ASC2]\ [BSC2]$$

IECS(1)=1 ; IECS(2)=3 ; IECS(3)=6

<u>CASO 2</u>:

IRC(1)=3 ; IRC(2)=2 ; IRC(3)=1

IECL(1)=3 ; IECL(2)=5 ; IECL(3)=6

$$\left[\begin{array}{c:c:c} 010 & 00 & 0 \\ 001 & 00 & 0 \\ XXX & XX & X \\ \hdashline 000 & 01 & 0 \\ XX0 & XX & X \\ \hdashline X00 & X0 & X \end{array}\right] \left[\begin{array}{c} 000 \\ 000 \\ 1XX \\ \hdashline 000 \\ 01X \\ \hdashline 001 \end{array}\right] = [ALC4]\ [BLC4]$$

$$\left[\begin{array}{c:c:c} X & X0 & X00 \\ \hdashline 0 & 01 & 000 \\ X & XX & XX0 \\ \hdashline 0 & 00 & 010 \\ 0 & 00 & 001 \\ X & XX & XXX \end{array}\right] \left[\begin{array}{c} 001 \\ \hdashline 000 \\ 01X \\ \hdashline 000 \\ 000 \\ 1XX \end{array}\right] = [ASC2]\ [BSC2]$$

IECS(1)=1 ; IECS(2)=3 ; IECS(3)=6

<u>CASO 3</u>:

IRC(1)=2 ; IRC(2)=1 ; IRC(3)=3

IECL(1)=2 ; IECL(2) ; IECL(3)=6

$$\left[\begin{array}{c:c:c} 01 & 0 & 000 \\ XX & X & XX0 \\ \hdashline X0 & X & X00 \\ \hdashline 00 & 0 & 010 \\ 00 & 0 & 001 \\ XX & X & XXX \end{array}\right] \left[\begin{array}{c} 000 \\ 1X0 \\ \hdashline 010 \\ \hdashline 000 \\ 000 \\ 001 \end{array}\right] = [ALC4]\ [BLC4]$$

$$\left[\begin{array}{c:c:c} X & X0 & X00 \\ \hdashline 0 & 01 & 000 \\ X & XX & XX0 \\ \hdashline 0 & 00 & 010 \\ 0 & 00 & 001 \\ X & XX & XXX \end{array}\right] \left[\begin{array}{c} 010 \\ \hdashline 000 \\ 1X0 \\ \hdashline 000 \\ 000 \\ 001 \end{array}\right] = [ASC2]\ [BSC2]$$

IECS(1)=1 ; IECS(2)=3 ; IECS(3)=6

como podemos bem ver ASC2 manteve a mesma forma e a influência da sequência IRC(I) só apareceu em BSC2 , e em particular a sequência de J utilizada, ordem crescente dos índices relativos, faz com que todos os 'X' nulos situem-se acima dos blocos da diagonal principal mas o mais interessante diz respeito a BSC2 pois podemos observar que

$$\left[\begin{array}{c} 100 \\ \hdashline 000 \\ 010 \\ \hdashline 000 \\ 000 \\ 001 \end{array}\right] \qquad \left[\begin{array}{c} 001 \\ \hdashline 000 \\ 01X \\ \hdashline 000 \\ 000 \\ 1XX \end{array}\right] \qquad \left[\begin{array}{c} 010 \\ \hdashline 000 \\ 1X0 \\ \hdashline 000 \\ 000 \\ 001 \end{array}\right]$$

<u>CASO 1</u>   <u>CASO 2</u>   <u>CASO 3</u>

e os 'X' só ocorrem embaixo dos '1' à direita do '1' da linha em questão; para casos mais complexos deve-se lembrar que se houver índices repetidos esta idéia só se mostrará válida se mantivermos a ordem em que se encontram naturalmente os índices repetidos, e que a partir da linha que não tem 'X' começa tudo de novo, por exemplo:

$$BSC2 = \begin{bmatrix} 0100 \\ 0000 \\ 1X00 \\ 0000 \\ 0001 \\ 0000 \\ 0000 \\ 001X \end{bmatrix} ; \quad \begin{array}{l} IRC(2) = 1 \\ IRC(1) = 2 \\ IRC(4) = 2 \\ IRC(3) = 3 \end{array}$$

Podemos comparar FCCB2 e FCCS2 quanto ao número de elementos 'X' não nulos em ABC2 e ASC2 que deve ser o mesmo, visto que por uma transformação de similaridade de permutação podemos sair de uma forma e chegar na outra; por outro lado sabemos que o número destes parâmetros 'X' não varia em ASC2 , logo o mesmo deve ocorrer em ABC2 , no entanto este número de 'X' em ASC2 só fica evidente em ABC2 quando obtemos C3 ou C3A , para formar FCCB2 , com os índices J de tal forma a obter a sequência IRC(J) em ordem crescente. Nesta construção, para FCCB2 os parâmetros 'X' não nulos ficam todos agrupados no início de cada linha estrutural e embora não seja tão evidente que este mesmo número de 'X' não nulos ocorra para outras sequências de J este fato deve ser verificado.

Para exemplificar as observações acima tomemos um sistema de índices relativos: IRC(1)=3 ; IRC(2)=1 ; IRC(3)=2 , com isso: ASC2 pode assumir as seguintes formas:

$$\left[\begin{array}{c|cc|ccc} X & X & 0 & X & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ X & X & X & X & X & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ X & X & X & X & X & X \end{array}\right] \quad \langle 1 \rangle \qquad \left[\begin{array}{c|ccc|cc} X & X & 0 & 0 & X & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ X & X & X & X & X & X \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \\ X & X & X & 0 & X & X \end{array}\right] \quad \langle 2 \rangle \qquad \left[\begin{array}{cc|c|ccc} 0 & 1 & 0 & 0 & 0 & 0 \\ X & X & X & X & X & 0 \\ \hline X & 0 & X & X & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ X & X & X & X & X & X \end{array}\right] \quad \langle 3 \rangle$$

$$\left[\begin{array}{cc|ccc|c} 0 & 1 & 0 & 0 & 0 & 0 \\ X & X & X & X & 0 & X \\ \hline 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ X & X & X & X & X & X \\ \hline X & 0 & X & 0 & 0 & X \end{array}\right] \quad \langle 4 \rangle \qquad \left[\begin{array}{ccc|cc|c} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ X & X & X & X & X & X \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ X & X & 0 & X & X & X \\ \hline X & 0 & 0 & X & 0 & X \end{array}\right] \quad \langle 5 \rangle \qquad \left[\begin{array}{ccc|c|cc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ X & X & X & X & X & X \\ \hline X & 0 & 0 & X & X & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 \\ X & X & 0 & X & X & X \end{array}\right] \quad \langle 6 \rangle$$

que sempre tem 14 'X' não nulos.

ABC2 'ã priori' deve assumir as seguintes formas:

$$\begin{bmatrix} X & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ X & X & X & X & X & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ X & X & X & X & X & X \end{bmatrix} \quad \langle 1 \rangle \qquad \begin{bmatrix} X & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ X & X & X & X & X & Y \\ X & X & X & X & X & X \end{bmatrix} \quad \langle 2 \rangle \qquad \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ X & X & X & Y & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ X & X & X & X & X & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ X & X & X & X & X & X \end{bmatrix} \quad \langle 3 \rangle$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ X & X & X & Y & Y & 0 \\ X & X & X & X & X & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ X & X & X & X & X & X \end{bmatrix} \quad \langle 4 \rangle \qquad \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ X & X & X & Y & Y & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ X & X & X & X & X & Y \\ X & X & X & X & X & X \end{bmatrix} \quad \langle 5 \rangle \qquad \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ X & X & X & Y & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ X & X & X & X & X & Y \\ X & X & X & X & X & X \end{bmatrix} \quad \langle 6 \rangle$$

das quais temos a seguinte quantidade de parâmetros do tipo 'X':

<1> → 14   <2> → 15   <3> → 15

<4> → 16   <5> → 17   <6> → 16

no entanto sabemos que cada uma destas formas deve recair na outra, assim como recair em qualquer uma das ASC2 por uma transformação de similaridade de permutação logo não deve alterar o número de 'X' não nulos, então também para os casos ABC2 <2>, <3>, <4> , <5> e <6> devemos ter 14 elementos e verificamos que tal ocorre uma vez que oa parâmetros que foram denotados por 'Y' são nulos.

Dada estas características sobre os parâmetros 'X' nas formas FCCL4, FCCB2 e FCCS2 , podemos obter qualquer uma destas formas pelo método de Jordan e Sridhar , a partir de um mesmo algorítmo e com uma posterior mudança na localização dos parâmetros obtidos. Assim sendo podemos obter as linhas estruturais por, no nosso exemplo;

$$CL2(.,.) * \left[\begin{array}{c:c:c} -ALFA(1,1,0) & -ALFA(2,1,0) & -ALFA(3,1,0) \\ -ALFA(1,1,1) & -ALFA(2,1,1) & -ALFA(3,1,1) \\ -ALFA(1,1,2) & 0 & -ALFA(3,1,2) \\ \hdashline -ALFA(1,2,0) & -ALFA(2,2,0) & -ALFA(3,2,0) \\ \hdashline -ALFA(1,3,0) & -ALFA(2,3,0) & -ALFA(3,3,0) \\ -ALFA(1,3,1) & 0 & -ALFA(3,3,1) \end{array}\right] = AN(.,.)$$

de onde obteremos os ALFA(.,.,.) e os parâmetros 'X' virão de:

$$\begin{bmatrix} 1 & ALFA(2,1,1) & ALFA(3,1,2) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} *ARE(.,.) = \left[\begin{array}{c:c:c:c:c:c} -ALFA(1,1,0) & -ALFA(1,1,1) & -ALFA(1,1,2) & -ALFA(2,1,0) & -ALFA(3,1,0) & -ALFA(3,1,1) \\ \hdashline -ALFA(1,2,0) & 0 & 0 & -ALFA(2,2,0) & -ALFA(3,2,0) & 0 \\ \hdashline -ALFA(1,3,0) & -ALFA(1,3,1) & 0 & -ALFA(2,3,0) & -ALFA(3,3,0) & -ALFA(3,3,1) \end{array}\right]$$

onde ARE(.,.) , que são as linhas estruturais de ALC4(.,.), deve assumir a forma:

$$\begin{bmatrix} X & X & X & X & X & X \\ X & 0 & 0 & X & X & 0 \\ X & X & 0 & 0 & X & X \end{bmatrix} = \text{ARE}(.,.)$$

sendo que se no lugar dos elementos nulos surgirem valores que dentro da precisão utilizada não podem ser supostas nulas, isto será um indício de que algo está incorreto.

Aqui também podemos observar que os parâmetros nulos de AFL(.,.) são os mesmo de ARE(.,.) , isto é, de mesma localização.

Para obter FCCB2 poderíamos rearrumar FCCL4 ou resolver a segunda equação matricial

$$\begin{bmatrix} \text{ALFA}(2,1,1) & \text{ALFA}(3,1,2) & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} X & X & X & 0 & 0 & 0 \\ X & X & X & X & 0 & 0 \\ X & X & X & X & X & X \end{bmatrix} =$$

$$= - \begin{bmatrix} \text{ALFA}(1,2,0) & \text{ALFA}(2,2,0) & \text{ALFA}(3,2,0) & 0 & 0 & 0 \\ \text{ALFA}(1,3,0) & \text{ALFA}(2,3,0) & \text{ALFA}(3,3,0) & \text{ALFA}(1,3,1) & \text{ALFA}(3,3,1) & 0 \\ \text{ALFA}(1,1,0) & \text{ALFA}(2,1,0) & \text{ALFA}(3,1,0) & \text{ALFA}(1,1,1) & \text{ALFA}(3,1,1) & \text{ALFA}(1,1,2) \end{bmatrix}$$

No caso de querermos obter FCCS1 deveríamos em primeiro lugar resolver:

$$\text{CL1}(.,.)*\text{FAL}(.,.) = \text{AN}(.,.)$$

e assim obter FAL(.,.) para com o algorītmo abaixo obtermos FCCS1.

<u>ALGORĪTMO 9</u>: FCCS1

.- Obtendo os parâmetros 'X' em linhas, ou seja dos blocos principais.

```
1.-   DO   4.-     KA=1,K
2.-   DO   4.-     J=1,IRC(KA)
3.-   V = IECL(KA)-IRC(KA)+J
4.-   ASC1(IECL(KA),V) = FAL(KA,V)
```

.- Obtendo os parâmetros 'X' em colunas ou seja, dos blocos acima dos principais.

```
5.-   DO   10.-    KB=2,K
6.-   DO   10.-    KA=1,KB-1
7.-   DO   10.-    J=1,IRC(KA)
8.-   V = IECL(KA)-IRC(KA)+J
9.-   U = IECL(KB)-IRC(KB)+1
10.-  ASC1(IECL(KA)-J+1,U) = FAL(V,KB)
```

Para obtermos T = TSC1IN(.,.) usaremos o seguinte algorītmo:

<u>ALGORĪTMO 10</u>: TSC1IN(.,.)

```
1.-   T(.,IECL(KA)) = B(.,KA) ; KA=1,2,...,K
2.-   DO   5.-     I=1,K
3.-   DO   5.-     J=1,IRC(I)-1          (para IRC(I)≥2)
4.-   V = IECL(I)-J
5.-   T(.,V) = A(.,.)*T(.,V+1)-ASC1(IECL(I),V+1)*T(.,
      IECL(I))
```

Com esses algorītmos obtemos uma forma canônica que mantem o aspectode semi-desacoplamento, pois ASC1 e bloco triangular, e que sempre podemos usar em alocação de polos por realimentação de estado de modo direto quase como com o uso da tradicional FCCL4. Sempre é bom lembrar que explorar o aspecto de ser triangular por blocos também é interessante se lembrarmos que teremos $\sum_{I=1}^{K} [N-IECL(I)]$ parâmetros realmente livres que não vão alterar a localização de polos mas podem ser assinalados de modo que os zeros resultantes forneçam junto com os polos um comportamento desejado do sistema.

Se tivermos como objetivo alocar polos ou estimar estados e decidirmos por utilizar um método do tipo JORDAN-SRIDHAR, o algorítmo mais rápido é este em que se obtem FCCS1 , apesar de certos inconvenientes tais como o de não poder alocar tres pares de polos complexos em um sistema de ordem 6 mas com índices ímpares para os seus dois blocos.

Podemos também obter FCCS1 e em suas duas versões FCCS1A e FCCS1B , acima obtivemos FCCS1A , usando o método das transformações de similaridade elementar.

<u>ALGORÍTMO 11:</u> FCCS1A

.- PRIMEIRA FASE

1.- obter FCCL1A como no algorítmo do capítulo anterior.

.- SEGUNDA FASE

2.- JJ=I (ao fim da primeira fase I=N)

3.- J=I-1

.- Verificando se estamos sobre as linhas estruturais

4.- se H(I,J)=0 faça JJ=J e vá para 5.-
se não siga

5.- H(.,.) = EC[H;H(I,I),I,J;...;H(I,JJ),JJ,J]

6.- se I=2 vá para 8.-
se não siga

7.- I=I-1 ; vá para 3.-

.- Índices estruturais

8.- IECS(P) = IECL(K)-IRC(P)+1 se IRC(P)≠0
IECS(P)=0 se IRC(P)=0
para P=1,2,...,K

<u>ALGORÍTMO 12</u>: FCCS1B

.- PRIMEIRA FASE

primeira fase do algorítmo 23 do capítulo 3

.- SEGUNDA FASE

1.- JJ=I (na fase anterior terminou com I=N)

2.- J=I-1

.- Verificando se estamos sobre as linhas estruturais

3.- se H(I,J)=0 faça JJ-J e vá para 5.-
se não siga

4.- H(.,.) = EC[H;H(I,I),I,J;...;H(I,JJ),JJ,J]

5.- se I=2 pare
se não siga

6.- I=I-1 e vá para 2.-

Podemos observar aqui que para obter as formas FCCS1A ou FCCS1B o volume de operações empregadas em obter ASC1A ou ASC1B pode ser maior que o volume empregado ao obter ALC3A ou ALC3B porém se o interesse nosso for obter estas formas para, por exemplo, alocar polos por realimentação de estado ou estimar estados só estaremos interessados nos blocos principais, logo poderemos não levar em conta as operações que envolvem elementos outros que não os dos blocos principais. Isto, em resumo, nos leva a uma forma de menos operações e que sempre é útil para os problemas propostos acima, o que nem sempre ocorre com as formas FCCL3A e FCCL3B ou FCOL3A e FCOL3B nos casos de realimentação e estimação respectivamente.

Vamos esquematizar aqui os procedimentos pelo método das Transformações de Similaridade Elementar, para as diversas formas canônicas apresentadas, tomando como exemplo um sistema de índices 3,1,2:

I - FCCL1

$$\left[\begin{array}{cccccc:ccc} 0 & 0 & X & X & 0 & X & 1 & 0 & 0 \\ 1 & 0 & X & X & 0 & X & 0 & 0 & 0 \\ 0 & 1 & X & X & 0 & X & 0 & 0 & 0 \\ 0 & 0 & 0 & X & 0 & X & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & X & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & X & 0 & 0 & 0 \end{array}\right]$$

II - FCCB1

$$\left[\begin{array}{cccccc:ccc} 0 & X & 0 & 0 & X & X & 1 & 0 & 0 \\ 0 & X & 0 & 0 & X & X & 0 & 1 & 0 \\ 0 & X & 0 & 0 & X & X & 0 & 0 & 1 \\ 1 & X & 0 & 0 & X & X & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & X & X & 0 & 0 & 0 \end{array}\right]$$

III - FCCL2 (rearrumando FCCB1)

$$\left[\begin{array}{cccccc:ccc} 0 & 0 & X & X & 0 & X & 1 & 0 & 0 \\ 1 & 0 & X & X & 0 & X & 0 & 0 & 0 \\ 0 & 1 & X & 0 & 0 & X & 0 & 0 & 0 \\ 0 & 0 & X & X & 0 & X & 0 & 1 & 0 \\ 0 & 0 & X & X & 0 & X & 0 & 0 & 1 \\ 0 & 0 & X & 0 & 1 & X & 0 & 0 & 0 \end{array}\right]$$

IV - FCCL3A (modificada)

a.- Primeira fase [FCCL1]

b.- Segunda fase

$$\left[\begin{array}{cccccc:ccc} X & X & X & X & X & X & 1 & Z & Z \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & Y & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X & 0 & X & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & X & X & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

V - FCCL3B (modificada)

a.- Primeira fase

$$\left[\begin{array}{cccccc:ccc} X & X & X & X & X & X & 1 & 0 & 0 \\ 1 & X & X & X & X & X & 0 & 0 & 0 \\ 0 & 1 & X & X & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & X & X & X & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & X & X & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & X & 0 & 0 & 0 \end{array}\right]$$

b.- Segunda fase

$$\left[\begin{array}{cccccc:ccc} X & X & X & X & X & X & 1 & Z & Z \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & Y & Y \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X & X & X & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & X & X & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

Podemos ver que além de ser ligeiramente diferente a forma de se obter FCCL3A e FCCL3B , as formas resultantes são diferentes.

Prosseguindo temos:

VI - FCCS1A (modificada)

a.- Primeira fase [FCCL1]

b.- Segunda fase

$$\left[\begin{array}{cccccc:ccc} X & X & X & X & 0 & X & 1 & 0 & 0 \\ 1 & 0 & 0 & X & 0 & X & 0 & 0 & 0 \\ 0 & 1 & 0 & X & 0 & X & 0 & 0 & 0 \\ 0 & 0 & 0 & X & 0 & X & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & X & X & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

VII - FCCS1B (modificada)

a.- Primeira fase = primeira fase de FCCL3B

b.- Segunda fase

$$\left[\begin{array}{cccccc:ccc} X & X & X & X & X & X & 1 & 0 & 0 \\ 1 & 0 & 0 & X & X & X & 0 & 0 & 0 \\ 0 & 1 & 0 & X & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & X & X & X & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & X & X & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

Vamos agora introduzir uma outra forma canônica, a FCCS3 cujo algorítmo será exposto adiante.

VIII - FCCS3A

a.- Primeira fase (FCCB1)

b.- Segunda fase

$$\left[\begin{array}{cccccc:ccc} X & X & X & X & X & X & 1 & X & X \\ 0 & X & 0 & 0 & X & X & 0 & 1 & 0 \\ 0 & X & X & X & X & X & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

IX - FCCS3B

a.- Primeira fase

$$\left[\begin{array}{cccccc:ccc} X & X & X & X & X & X & 1 & 0 & 0 \\ X & X & X & X & X & X & 0 & 1 & 0 \\ X & X & X & X & X & X & 0 & 0 & 1 \\ 1 & X & X & X & X & X & 0 & 0 & 0 \\ 0 & 0 & 1 & X & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & X & X & 0 & 0 & 0 \end{array}\right]$$

b.- Segunda fase

$$\left[\begin{array}{cccccc:ccc} X & X & X & X & X & X & 1 & X & X \\ X & X & X & X & X & X & 0 & 1 & 0 \\ X & X & X & X & X & X & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

X - FCCB2A (rearrumando FCCS3A)

$$\left[\begin{array}{cccccc:ccc} X & X & X & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ X & X & X & X & X & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ X & X & X & X & X & X & 1 & X & X \end{array}\right]$$

XI - FCCB2B (rearrumando FCCS3B)

$$\left[\begin{array}{cccccc:ccc}
X & X & X & X & X & X & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
X & X & X & X & X & X & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
X & X & X & X & X & X & 1 & X & X
\end{array}\right]$$

XII - FCCL4A (rearrumando FCCS3A)

$$\left[\begin{array}{cccccc:ccc}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & X & X & X & 1 & X & X \\
X & 0 & 0 & X & X & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
X & X & 0 & X & X & X & 0 & 0 & 1
\end{array}\right]$$

XIII - FCCL4B (rearrumando FCCS3B)

$$\left[\begin{array}{cccccc:ccc}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
X & X & X & X & X & X & 1 & X & X \\
X & X & X & X & X & X & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
X & X & X & X & X & X & 0 & 0 & 1
\end{array}\right]$$

XIV - FCCS2A (rearrumando FCCS3A)

$$\left[\begin{array}{cccccc:ccc} X & X & 0 & X & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ X & X & X & X & X & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ X & X & X & X & X & X & 1 & X & X \end{array}\right]$$

XV - FCCS2B (rearrumando FCCS3B)

$$\left[\begin{array}{cccccc:ccc} X & X & X & X & X & X & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ X & X & X & X & X & X & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ X & X & X & X & X & X & 1 & X & X \end{array}\right]$$

Destas já havia sido discutido no capítulo anterior que para as formas FCCL1 , FCCL2, FCCS1 , FCCSL3 e FCCB1 usar a T.S.E. simples, desde que corretamente, é equivalente a usar a T.S.E. por blocos, já para as demais podemos encontrar uma T.S.E. por blocos que pode superar qualquer arranjo em T.S.E. simples.

Construímos os algorítmos primeiro descobrindo qual é o bloco que nos oferece menor número de operações; experimentando vários exemplos chegamos a conclusão de que o melhor bloco é, na segunda fase dos algorítmos posto que na primeira um T.S.E. simples é o suficiente, aquele que nenhuma operação T.S.E. simples neste bloco gerasse a necessidade de outra operação T.S.E. simples no mesmo bloco, ou seja, um bloco em que ao se executar as operações de similaridade não seja necessário operar no mesmo bloco. Por exemplo se formos obter FCCS3A de FCCB1A para $[ABC1 \vdots BBC1]$ na forma:

$$\left[\begin{array}{cccccc:ccc} 0 & X & 0 & 0 & X & X & 1 & 0 & 0 \\ 0 & X & 0 & 0 & X & X & 0 & 1 & 0 \\ 0 & X & 0 & 0 & X & X & 0 & 0 & 1 \\ 1 & X & 0 & 0 & X & X & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & X & X & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & X & X & 0 & 0 & 0 \end{array}\right]$$

e tomarmos como um bloco a última linha não teremos como consequência que novamente agir sobre este bloco; no entanto se usarmos como bloco as duas últimas linhas, ou seja, da última linha até a linha do índice da coluna que tem o pivot '1' mais um, pois é para a linha de mesmo índice que a coluna que contém o pivot '1' da linha última do bloco em questão é que irão alguns elementos. Logo se usarmos mais linhas não teremos mais o melhor bloco.

No entanto para programarmos em Fortran, esta tática de T.S.E. por blocos que economizará algumas operações, utilizaremos muita memória do computador e somente será útil para sistemas realmente grandes.

Com a observação acima vamos citar agora uma versão para se obter FCCS3A , que no programa utilizado por nós foi feita de tal modo a obter FCCS3A e FCCS3B por T.S.E. em blocos.

A única diferença básica de agir por T.S.E. por blocos além da pequena diferença em cálculo e razoável diferença em memória está no fato de termos que obter os blocos através do seguinte algorítmo:

ALGORÍTMO: 13 blocos da T.S.E.

1.- J=N

2.- LF=J

3.- J=coluna do pivot '1' na linha LF

4.- se $J \leq M$; LI=M+1 e pare
se não, LI=J+1 e vá para 2.-

Após já ter visto os blocos da T.S.E. vamos finalmente ao algorítmo que nos permitirá obter FCCS3A.

<u>ALGORÍTMO</u>: 14 FCCS3A

1.- obter FCCB1A

.- SEGUNDA FASE

2.- I=N

3.- J= coluna do pivot '1' na linha I

4.- se J+1=I vá para 7.-
se não siga

5.- DO 6.- L=I-1, J+1, -1

6.- H = EC[H;H(I,L),L,J]

7.- DO 8.- L=N,I,-1

8.- H = EC[H;H(I,L),L,J]

9.- I=I-1

10.- se I=M pare
se não vá para 3.-

Deste algorítmo obtemos FCCS3A e podemos obter FCCL4A, FCCS2A e FCCB2A com o uso de transformações elementares de similaridade do tipo permutação apenas.

Completando esta secção vamos colocar as formas canônicas FCCL1 e FCCL1 em uma forma parcial com o objetivo de auxílio na computação da função de transferência e dos fatores comuns por linhas ou colunas.

Para o caso de se obter as colunas da função de transferência G(s) vamos tentar obter:

$$ALC1P = \left[\begin{array}{l|l} 00 \ \ldots \ 0\text{-}Y & XX \ \ldots \ XX \\ 10 \ \ldots \ 0\text{-}Y & XX \ \ldots \ XX \\ 01 \ \ldots \ 0\text{-}Y & XX \ \ldots \ XX \\ \cdots\cdots\cdots & \cdots\cdots\cdots \\ 00 \ \ldots \ 1\text{-}Y & XX \ \ldots \ XX \\ \hline 00 \ \ldots \ 0 \ 0 & XX \ \ldots \ XX \\ 00 \ \ldots \ 0 \ 0 & XX \ \ldots \ XX \\ \cdots\cdots\cdots & \cdots\cdots\cdots \\ 00 \ \ldots \ 0 \ 0 & XX \ \ldots \ XX \end{array}\right] \qquad BLC1P = \left[\begin{array}{c} X \ \ldots \ 1 \ \ldots \ X \\ X \ \ldots \ 0 \ \ldots \ X \\ X \ \ldots \ 0 \ \ldots \ X \\ \cdots\cdots\cdots \\ X \ \ldots \ 0 \ \ldots \ X \\ \hline X \ \ldots \ 0 \ \ldots \ X \\ X \ \ldots \ 0 \ \ldots \ X \\ \cdots\cdots\cdots \\ X \ \ldots \ 0 \ \ldots \ X \end{array}\right]$$

$$CLC1P = \left[\begin{array}{l|l} XX \ \ldots \ XX & ZZ \ \ldots \ ZZ \\ \cdots\cdots\cdots & \cdots\cdots\cdots \\ XX \ \ldots \ XX & ZZ \ \ldots \ ZZ \end{array}\right]$$

pois neste caso podemos chegar a verificar que:

$$\underline{g}_t(S) = [CA] * [sI-A1]_1^{-1}$$

onde:

$$A1 = \begin{bmatrix} 00 \ \ldots \ 0\text{-}Y_r \\ 10 \ \ldots \ 0\text{-}Y_{r-1} \\ 01 \ \ldots \ 0\text{-}Y_{r-2} \\ \cdots\cdots\cdots \\ 00 \ \ldots \ 1\text{-}Y_1 \end{bmatrix} \qquad CA = \begin{bmatrix} XX \ \ldots \ XX \\ \cdots\cdots\cdots \\ XX \ \ldots \ XX \end{bmatrix}$$

e $[sI-A1]_1^{-1}$ é a primeira coluna de $[sI-A1]_1^{-1}$ de onde verificamos que:

$$[sI-A1]_1^{-1} = \frac{1}{\Delta} * \begin{bmatrix} 1 & Y_1 & Y_2 & \ldots & Y_{r-2} & Y_{r-1} \\ 0 & 1 & Y_1 & \ldots & Y_{r-3} & Y_{r-2} \\ 0 & 0 & 1 & \ldots & Y_{r-4} & Y_{r-3} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \ldots & 1 & Y_1 \\ 0 & 0 & 0 & \ldots & 0 & 1 \end{bmatrix} * \begin{bmatrix} s^{r-1} \\ s^{r-2} \\ s^{r-3} \\ \cdots \\ s \\ 1 \end{bmatrix} = \frac{1}{\Delta} * P * S$$

logo as linhas de CA*P nos dão os coeficientes dos numeradores dos elementos da coluna t de G(s) ; lembrando que o denominador comum é $\Delta = s^r + \gamma_1 s^{r-1} + \ldots + \gamma_r$, ou seja se definimos $\underline{q} = [+\gamma_1 + \gamma_2 \ldots +\gamma_{r-1} +\gamma_r]$ teremos:

$$\left[\begin{array}{c|c} CA & 0 \\ \hline 0 & 1 \end{array}\right] * \left[\begin{array}{c} P \\ \hline \underline{q} \end{array}\right] = \left[\begin{array}{c} CA*P \\ \hline \underline{q} \end{array}\right] = \hat{C} * \bar{P} = GT(.,T)$$

onde os primeiros L linhas indicam os coeficientes dos polinômios dos numeradores, e a última linha indica os coeficientes do polinômio do denominador, polinômio este que é comum a todos os elementos desta linha.

Para se obter a função de transferência total refazemos tudo isto para cada uma das entradas.

No caso de obter as linhas da função de transferência usaremos um procedimento análogo mas com a forma parcial de FCOL1.

Como um algorítmo podemos sugerir:

<u>ALGORÍTMO 15</u>:

1.- DO 8.- I=1,M

2.- selecione os vetores linearmente independentes de A(.,.)**K*B(.,I), K=0,1,... . Obtendo a dependência em K

3.- adicione N-K colunas linearmente independentes destas K já selecionadas formando assim CLP1(.,.).

4.- Forma canônica controlável parcial:

$$\left[\begin{array}{c|c} CLP1IN & 0 \\ \hline 0 & IDL \end{array}\right] * \left[\begin{array}{c|c} A & B \\ \hline C & 0 \end{array}\right] * \left[\begin{array}{c|c} CLP1 & 0 \\ \hline 0 & IDM \end{array}\right] = \left[\begin{array}{c|c} ALC1P & BLC1P \\ \hline CLC1P & 0 \end{array}\right]$$

5.- selecione A1 e CA

6.- de A1 obter P, $\underline{q}$ e $\bar{P}$.

7.- de CA obter $\hat{C}$

8.- $GT(.,I) = \hat{C}*\hat{P}$

Veremos que este procedimento é mais rápido com o uso de transformações de similaridade elementares.

Apesar disto vamos ver como obteríamos estas colunas da função de transferência usando além do método apresentado e o da T.S.E. , o de Jordan-Sridhar.

No caso de querermos obter a função de transferência como foi feito no algorítmo anterior devemos observar que os valores 'Y' de ALC1P podem ser obtidos de:

$$CL1P(.,.)*FLA(.) = A(.,.)**K*B(.,T)$$

onde CL1P(.,.) é formada pelas primeiras K colunas de CLP1(.,.) do algorítmo 13 , e sendo que FLA(.) é a última coluna de A1 ; logo ao resolvermos a equação acima já temos condição de obter P, $\underline{q}$ e consequentemente $\hat{P}$.

Para obter CA(.,.) basta tomarmos as K primeiras colunas de CLC1P = C*C2P1 , ou seja, basta tomarmos:

$$CA(.,.) = C(.) * CL1P(.,.) \tag{16}$$

e agora com CA obtemos $\hat{C}$ e de $\hat{C}$ e $\hat{P}$ obtemos a coluna t da função de transferência.

Voltando ao uso das transformações de similaridade elementar vamos desenvolver um algorítmo para obtermos a função de transferência de um sistema sendo dada a sua formulação em espaço de estado.

Este algorítmo é o que obtém FCCL1 truncado, quando obtivermos a primeira dependência entre as colunas da matriz de estado, pois neste estágio teríamos a configuração ALC1P , BLC1P e CLC1P , conseguindo assim obter uma das colunas da função de transferência.

<u>ALGORÍTMO 16</u>: Função de transferência por linhas

.- Inicialização

1.- Construir a matriz do sistema H(.,.)

2.- I=1 (I=contador de linhas de H(.,.))

3.- J=N+1 (J=contador de colunas de H(.,.))

4.- JB=1 (JB=contador de colunas de B(.,.))

5.- IX=1 ; JX=0

.- Procurando o melhor pivot

6.- L=valor de K onde ocorre max|H(K,J)|
K=I,I+1,...,N

7.- G=H(L,J)

.- Detetando a dependência linear

8.- se G≠0 vá para 14.-
se G=0 siga

.- Teste de fim de algorítmo

9.- se JB=M pare
se não siga

10.- JB=JB+1

11.- se IX=0 vá para 12.-
se não J=N+JB ; I=1 e vá para 6.-

12.- obter CA e dada a última coluna de A1 obter, através de CA também a coluna (JB-1)-ésima da função de transferência e o índice JX.

13.- J=N+JB ; I=1 e vá para 5.-

14.- IX=0 ; JX=JX+1

.- Colocando o pivot na linha I e coluna J

15.- H(.,.) = EP[H;I,L]

16.- H(.,.) = EM[H;1/G,I]

.- Eliminação por colunas

17.- H(.,.) = EC[H;-H(K,J),I,K]
K=I+1,...,N,1,...,I-1

18.- J=I ; I=I+1 e vá para 6.-

No caso de obtermos as linhas da função de transferência usaríamos um algorítmo semelhante porém operando entre a matriz de saída e a de estado.

Para este algorítmo sugerido, a matriz de saída, em cada coluna calculada, terá como suas L primeiras linhas, L é o número de saídas, os coeficientes dos polinômios dos numeradores dos respectivos elementos da função de transferência, sendo estes coeficientes dados em sequência dos expoentes decrescentes, isto é, são coeficientes desde o do primeiro termo, de expoente JX-1, até o coeficiente do último termo, o independente. Como última linha teremos os coeficientes do polinômio do denominador dos elementos desta coluna sendo que este polinômio que é uma ordem superior ao dos numeradores tem a unidade como coeficiente do termo de expoente JX .

Este tipo de procedimento com a T.S.E. foi sugerido por DALY e pode ser aperfeiçoado com o uso de FCCS1B ou FCOS1B truncado. Com esta nova alternativa vamos evitar de ter que construir a matriz P , uma vez que:

$$
\mathrm{ASC1P}=\left[\begin{array}{ccccc:ccccc}
-Y & -Y & \dots & -Y & -Y & X & X & \dots & X & X\\
1 & 0 & \dots & 0 & 0 & X & X & \dots & X & X\\
0 & 1 & \dots & 0 & 0 & X & X & \dots & X & X\\
\cdots & & & & & \cdots & & & & \\
0 & 0 & \dots & 1 & 0 & X & X & \dots & X & X\\
\hdashline
0 & 0 & \dots & 0 & 0 & X & X & \dots & X & X\\
0 & 0 & \dots & 0 & 0 & X & X & \dots & X & X\\
\cdots & & & & & \cdots & & & & \\
0 & 0 & \dots & 0 & 0 & X & X & \dots & X & X
\end{array}\right]
=\left[\begin{array}{ccccc:ccccc}
 & & & & & X & X & \dots & X & X\\
 & & & & & X & X & \dots & X & X\\
 & & A1 & & & X & X & \dots & X & X\\
 & & & & & \cdots & & & & \\
 & & & & & X & X & \dots & X & X\\
\hdashline
0 & 0 & \dots & 0 & 0 & X & X & \dots & X & X\\
0 & 0 & \dots & 0 & 0 & X & X & \dots & X & X\\
\cdots & & & & & \cdots & & & & \\
0 & 0 & \dots & 0 & 0 & X & X & \dots & X & X
\end{array}\right];
$$

$$
\left[\begin{array}{ccccc}
X & \dots & 1 & \dots & X\\
X & \dots & 0 & \dots & X\\
X & \dots & 0 & \dots & X\\
\cdots & & & & \\
X & \dots & 0 & \dots & X\\
X & \dots & 0 & \dots & X\\
X & \dots & 0 & \dots & X\\
\cdots & & & & \\
X & \dots & 0 & \dots & X
\end{array}\right] = \mathrm{BSC1P}
$$

$$BSCIP = \left[\begin{array}{cccc|cccc} X & X & \ldots & X \; X & Z & Z & \ldots & Z \; Z \\ \cdots & & & & \cdots & & & \\ X & X & \ldots & X \; X & Z & Z & \ldots & Z \; Z \end{array}\right] = \left[\begin{array}{c|c} CS & CZ \end{array}\right]$$

onde neste caso

$$\left[sI-\Delta_1\right]_1^{-1} \quad \begin{bmatrix} s^{r-1} \\ \cdots \\ 1 \end{bmatrix} * \frac{1}{\Delta} = S(.) * \frac{1}{\Delta}$$

logo $\underline{g}_t(s) = CS(.,.) * [sI-A1]_1^{-1} = CS(.,.) * S(.) * \frac{1}{\Delta}$ e teremos então como coeficientes do polinômio do numerador as linhas da matriz CS , assim obteremos diretamente a função de transferência, uma vez que a primeira linha de A1(.,.) tem os coeficientes do polinômio comum da coluna em questão com o sinal trocado e na sequência dos expoentes decrescentes; em CS(.,.) temos os coeficientes em ordem também dos expoentes decrescentes mas já com o sinal correto.

Assim sendo, o algorítmo final completo é:

ALGORÍTMO 15:

.- PRIMEIRA FASE

.- Inicialização

1.- Construir a matriz do sistema

2.- I=1 (I=contador de linhas de H(.,.))

3.- J=N+1 (J=contador de colunas de H(.,.))

4.- JB=1 (JB=contador de colunas de B(.,.))

5.- IX=1 ; JX=0

.- Procurando o melhor Pivot

6.- L=valor de K onde ocorre max|H(K,J)|

7.- G=H(L,J)

.- Detetando a dependência linear

8.- se $G \neq 0$ vã para 12.-
se $G=0$ siga

.- Teste de fim de algorítmo

9.- se JB=M pare
se não siga

10.- JB=JB+1

11.- se IX=0 vã para 17.-
se não J=N+JB ; I=1 e vã para 6.-

12.- IX=0 ; JX=JX+1

.- Colocando o pivot na linha I e coluna J

13.- H(.,.) = EP[H;I,L]

14.- H(.,.) = EM[H;1/G,I]

.- Eliminação por colunas

15.- H(.,.) = EC[H;-H(K,J),I,K]
K=I+1,...,N

16.- J=I ; J=I+1 e vã para 6.-

.- SEGUNDA FASE

.- Eliminação por linhas

17.- J=I-1

18.- H(.,.) = EC[H;H(I,K),K,J]
K=I,...,JX

19.- se I=2 vã para 21.-
se não siga

20.- I=I+1 vã para 17.-

.- Obtendo a função de transferência

21.- Obter CS(.,.)
e obter da primeira linha de A1(.,.) os coeficientes de Δ

22.- J=N+JB ; I=1 e vã para 5.-

CAPÍTULO V

Finalmente neste capítulo vamos traçar as características gerais do trabalho desenvolvido.

No capítulo II foram apresentadas as formas canônicas de Luenberger e as formas canônicas alternativas de Bingulac. Em particular foi desenvolvida, de melhor modo, a forma FCCL3 , uma vêz que no trabalho original de Luenberger não eram previstos os fatos desta forma ser bloco-triangular e de em BLC3 os parâmetros 'Y' e 'Z' só ocorrerem em locais determinados pela sequência de índices relativos.

Uma vêz com a apresentação topológica destas formas canônicas, no capítulo III foram apresentadas duas técnicas de obtê-las: a de Jordan-Sridhar e a de Daly. Quanto a metodologia de Jordan-Sridhar resumi-mo-la em três equações matriciais que, por exemplo, no caso de FCCL4 são:

$$CL2(.,.)*ALF(.,.) = AN(.,.) \quad (1)$$

$$P(.,.)*ARE(.,.) = AFL(.,.) \quad (2)$$

$$BRE(.,.) = PIN(.,.) \quad (3)$$

onde as diversas matrizes foram definidas no referido capítulo, sendo que as matrizes P(.,.) e AFL(.,.) são obtidas rearrumando os elementos de ALF(.,.) . Dado que trabalhamos sobre idéias computacionais, convém lembrar que para iniciar a resolução deste modo é necessário ter CL2(.,.) , que é obtida de RAB(.,.) ; então podemos sugerir que obter as colunas linearmente independentes de RAB(.,.) , que formam CL2(.,.) , e resolver a equação (1) seja feito de uma só vêz. Isto pode ser alcançado uma vêz que podemos obter as colunas linearmente independentes de RAB(.,.) usando sobre esta matriz o método de eliminação de Gauss, que vai gerar uma matriz pré-multiplicadora para RAB(.,.) e que fará com que as colunas linearmente independentes de RAB(.,.) sejam transformadas em colunas da matriz IDN ; porque aquelas colunas, se rearrumadas convenientemente, são as colunas que formam CL2(.,.) , então a ma triz formada acima é CL2IN(.,.) , de certo modo, e portanto as primeiras colunas linearmente dependentes relativas a cada entrada são, após seus elementos serem reordenados adequadamente,CL2IN(.,.) *AN(.,.) que é ALF(.,.).

Além do método de Jordan-Sridhar, que apresenta como limitação não poder calcular FCCL3 em geral, foi apresentado o método das transformações de similaridade elementares proposto por Daly. Nesta parte do capítulo foram apresentados diversos algorítmos os quais, em geral, obtêm algumas formas de Luenberger ligeiramente modificadas; devemos observar o fato de que muitas formas canônicas foram denotadas por um caracter final A ou B , FCCL3A ou FCCL3B por exemplo, com isso não queríamos apenas diferenciar dois meios de chegar ao mesmo final, mas dois meios que chegam, em geral, a dois finais, os de terminações A são os que obtêm as formas de Luenberger e os outros obtêm uma forma topologicamente igual mas com valores diferentes para os parâmetros 'X' que aqueles encontrados pela sua construção original. Neste ponto localizamos uma falha no trabalho que propos este método, pois no referido trabalho, cujo objetivo é obter as formas canônicas de Luenberger, na realidade são obtidas as formas do tipo B .

Também foi traçado, neste capítulo terceiro, que as transformações de similaridade elementares podem ser feitas basicamente de dois modos: as T.S.E. simples e as T.S.E. por blocos, sendo que em particular se o bloco for uma linha ou uma coluna teremos a idéia de Daly. Foi mostrado que a mais útil para programação é a T.S.E. simples que não necessita de memorizar o bloco de operações efetuadas. Este bloco pode ser armazenado no próprio local em que se encontra uma vêz que após ter sido completada a operação estes elementos deverão se tornar nulos, mas para que estes elementos não sejam afetados devemos operar convenientemente sobre a matriz em questão. Isto apenas trocaria um espaço de memória necessário para memorizar a sequência de operações por um espaço de memória requerido para a programação; em geral podemos obter uma T.S.E. simples que execute o mesmo número de operações que a T.S.E. bloco, a excessão reside apenas nos algorítmos de duas fases em que a primeira fase é obter algo do tipo FCCB1 . Mesmo neste caso vemos que além do bloco ser um conjunto de linhas, ou colunas, que requer um determinado algorítmo para ser obtido e consequentemente espaço na memória do computador, o volume de operações em relação ao uso da T.S.E. simples é pouco menor, justificando ainda o uso da T.S.E. simples.

Neste ponto queremos fazer alguma observação sobre os métodos de Luenberger (Bingulac), Jordan-Sridhar e Daly para obter as formas canônicas vistas até aqui; esta observação será feita tanto no aspecto numérico quanto computacional.

Começaremos comparando numericamente o método de Luenberger, sobre o qual as matrizes inversas são calculadas pelo método de eliminação de Gauss, o método de Jordan e Sridhar com umas correções na tabela proposta em seu trabalho, e o método de Daly, ambos os métodos aplicados no sentido de se obter FCCL4.

Para isso é necessário citar o número de operações envolvidas na inversão de uma matriz e na resolução de $V*X=W$ onde V é uma matriz NxN e W é uma matriz NxM, sendo X a matriz procurada.

No caso de inversão de matrizes são envolvidas N^3 operações, se for inversão de uma matriz triangular serão envolvidas $(N^3-N)/6$ operações; no caso de se obter a matriz X são envolvidas $(N^3-N)/3 + M(N-N)$ operações.

TABELA 1:		Luenberger
CL2	:	$N^2(N-M)$
CL2IN	:	N^3
CL4	:	$N^2(N-M)$
CL4IN	:	N^3
ALC4	:	$2N^3$
BLC4	:	N^2M
TOTAL	:	GN^3-N^2M

TABELA 2:		Jordan e Sridhar
CL2	:	$N^2(N-M)$
AN	:	N^2M

ALF	:	$(N^3-N)/3 + M(N^2-N)$
ARE	:	$MN(N-1)/2$
BRE	:	$(M^3-M)/6$
CL4	:	$N^2(N-M)-NM(M+1)/2$
CL4IN	:	N^3
TOTAL	:	$(10/3)N^3+(M/2)N^2-(3M^2+12M+2)N+(M/6)(M^2-1)$

No caso assintótico, isto é, $N>>M$, para obtermos ALC4, BLC4, CL4, CL4IN teremos

TABELA 3

Luenberger	:	$(18/3)\ N^3$
Jordan e Sridhar	:	$(10/3)\ N^3$
Daly	:	$(6/3)\ N^3$

Se, como no caso de alocação de polos por realimentação de estado, se apenas ALC4 e BLC4 forem necessários teremos:

TABELA 4

Luenberger	:	$(18/3)\ N^3$
Jordan e Sridhar	:	$(4/3)\ N^3$
Daly	:	$(3/3)\ N^3$

No caso de usarmos C2A, CS2A, CS2A ou C3A, teremos mais operações do que no método direto de Luenberger, que usa diretamente C2 , pois recairemos em uma forma intermediária para finalmente obtermos a forma desejada.

No capítulo IV foram demonstrados quais os elementos 'X' das formas de Luenberger são nulos, demonstradas as formas e demonstrado que o número de elementos não nulos em FCCL4 é o mesmo que em FCCL2 . Com a finalidade de explorar a bloco-triangular mente, nos problemas de alocação de polos ou estimação de estado, foi proposta a forma FCCS1 que possui o mesmo número de 'X' não nulos que FCCL1 . No caso de FCCL3 for demonstrado que os índices relativos entre as construções de primeira e segunda fase do artigo de Luenberger permanecem realmente os mesmos, e quais os elementos 'Y' e 'Z' de BLC3 serão nulos dependendo da sequência dos índices relativos de controlabilidade.

Quanto às formas alternativas de Bingulac mostramos que somente em FCCB2 , formado de tal forma que em sua segunda fase selecionemos os índices relativos em ordem crescente, é que os elementos 'X' não são necessariamente nulos e que o caráter de invariância de ABC2 pode ser encontrado em ASC2 também.

Finalmente como um uso de FCCS1 foi melhorado o método proposto por Daly para obter função de transferência.

Obs: nas formas especiais a letra S também se refere a um nome que é o da minha amada esposa SANDRA.

REFERÊNCIAS BIBLIOGRÁFICAS

/01/ D. G. LUENBERGER
"Canonical forms for linear multivariable systems"
IEEE Trans. Aut. Cont.
Vol: AC-12 p.p. 290-293 June 1967

/02/ D. JORDAN B. SRIDHAR
"An efficient algorithm for calculation of the LUENBERGER canonical form"
IEEE Trans. Aut. Cont.
Vol: AC-18 p.p. 292-295 June 1973

/03/ K. C. DALY
"The computation of LUENBERGER canonical forms using elementary similarity transformations"
Int. J. Syst. Sci.
Vol: 7 p.p. 1-15 January 1976

/04/ R. D. GUPTA F. W. FAIRMAN
"LUENBERGER's canonical forms revisited"
IEEE Trans. Aut. Cont.
Vol: AC-19 p.p. 440-441 August 1974

/05/ S. P. BINGULAC C. P. BOTTURA A. M. FARTES Fº
"Algorítmo de esfôrço mínimo para alocação de polos em sistemas lineares multivariáveis"
1º Congresso da S.B.A.

/06/ D. G. DENERY
"Comments on "Luenberger's canonical form revisited"
IEEE Trans. Aut. Cont.
Vol: AC-20 p.p. 444-445 June 1975

/07/ S. P. BINGULAC
"On the minimal number of parameters in linear multivariable systems"

IEEE Trans. Aut. Cont.
Vol: AC-21 p.p. 604-605 August 1976

/08/ K. C. DALY
"A computational procedure for transfer function evaluation"
Int. J. Cont.
Vol: 20 p.p. 569-576 1974

/09/ A. M. FARTES Fº
"Controle digital direto de sistemas multivariáveis"
Relatório final FAPESP

/10/ S. P. BINGULAC
"An aproach to pole placement by constant output feedback"
Submitted at 7-th IFAC Congress, 1978

/11/ C. T. CHEN
Introduction to linear systems theory 1970

/12/ C. BONIVENTO R. GUIDORZI G. MARRO
"Irreducible canonical realizations from external data sequences"
Int. J. Cont.
Vol: 17 p.p. 553-563 nº 3 1973

/13/ W. M. WONHAM
"On pole assignment in multi-input controllable linear systems"
IEEE Trans. Aut. Cont.
Vol: AC-12 p.p. 660-665 December 1967

/14/ K. B. DATTA
"Canonical form of reduced nonfixed parameters"
IEEE Trans. Aut. Cont.
Vol: AC-21 p.p. 412-413 June 1976

/15/ J. RISSANEN
"Basis of invariants and canonical forms for linear dynamic systems"

Automatica
Vol: 10 p.p. 173-182 1974

/16/ S. H. WANG E. J. DAVISON
"Canonical forms of linear multivariable systems"
SIAM J. Cont. Optimi.
Vol: 14 p.p. 236-250 February 1976

/17/ R. D. GUPTA F. W. FAIRMAN
"Canonic-form realisation from general input-output data"
Proc. of the IEE
Vol: 122 p.p. 657-662 June 1975

/18/ J. D. SIMON S. K. MITTER
"A theory of modal control"
Information and Control
Vol: 13 p.p. 316-358 1968

/19/ H. H. ROSENBROCK
State space and multivariable systems 1970
Nelson Londres

/20/ F. R. GANTMACHER
Theory of matrices
Vol: I e II
Chelsea Publishing Co. 1959

/21/ C. P. BOTTURA
Sistemas de controle
Almeida Neves editores - Rio (no prelo)

/22/ C. P. BOTTURA A. M. FARTES Fº
"Controle digital direto de sistemas multivariáveis"
IV Congresso da S.B.S. 1976

APÊNDICE

Aqui apresentamos as matrizes do sistema A, B e C e as diversas matrizes geradas a partir destas. Assim sendo vamos primeiramente obter as matrizes de controlabilidade e observabilidade para destas extrair as transformações que nos levem a FCCL1, FCOL1, FCCL2 e FCOL2. A seguir vamos obter as matrizes auxiliares de FCCL3, esta mesma forma e as matrizes que compõe a demonstração da existência de tal forma canônica, matrizes estas referidas no capítulo quatro. As próximas são as matrizes citadas acima mas no caso FCCL4 . Prosseguindo temos as matrizes que compõe FCCB1, FCOB1, FCCB2, FCCS1 e FCCS2. Finalmente temos uma exemplificação por colunas de função de transferência.

Nas demais folhas seguem exemplos fornecidos pelos algorítmos implementados no computador e estes algorítmos.

MATRIZES DO SISTEMA --- A B C

A

1.00	0.00	0.00	0.00
0.00	2.00	0.00	0.00
0.00	0.00	3.00	0.00
0.00	0.00	0.00	4.00

B

1.00	0.00	1.00
1.00	0.00	0.00
1.00	1.00	1.00
0.00	1.00	0.00

C

1.00	1.00	0.00	1.00
0.00	1.00	1.00	0.00
0.00	0.00	1.00	1.00

MATRIZ DE CONTROLABILIDADE --- RAB

RAB

1.00	0.00	1.00	1.00	0.00	1.00	1.00	0.00	1.00	1.00
1.00	0.00	0.00	2.00	0.00	0.00	4.00	0.00	0.00	8.00
1.00	1.00	1.00	3.00	3.00	3.00	9.00	9.00	9.00	27.00
0.00	1.00	0.00	0.00	4.00	0.00	0.00	16.00	0.00	0.00

0.00	1.00
0.00	0.00
27.00	27.00
64.00	0.00

MATRIZ DE OBSERVABILIDADE --- SAC

SAC

1.00	1.00	0.00	1.00
0.00	1.00	1.00	0.00
0.00	0.00	1.00	1.00
1.00	2.00	0.00	4.00
0.00	2.00	3.00	0.00
0.00	0.00	3.00	4.00
0.00	4.00	0.00	16.00
0.00	4.00	9.00	0.00
0.00	0.00	9.00	16.00
1.00	8.00	0.00	64.00
0.00	8.00	27.00	0.00
0.00	0.00	27.00	64.00

FORMA CANÔNICA CONTROLÁVEL I - FCCL1

CL1 ALC1 BLC1 CLC1

CL1

1.00	1.00	1.00	0.00
1.00	2.00	4.00	0.00
1.00	3.00	9.00	1.00
0.00	0.00	0.00	1.00

ALC1

0.00	0.00	6.00	-1.00
1.00	0.00	-11.00	1.50
0.00	1.00	6.00	-0.50
0.00	0.00	0.00	4.00

BLC1

1.00	0.00	4.00
0.00	0.00	-4.00
0.00	0.00	1.00
0.00	1.00	0.00

FORMA CANÔNICA OBSERVÁVEL I - FCOL1

OL1 ALO1 BLO1 CLO1

OL1

1.00	1.00	0.00	1.00
1.00	2.00	0.00	4.00
1.00	4.00	0.00	16.00
0.00	1.00	1.00	0.00

ALO1

0.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00
8.00	-14.00	7.00	0.00
2.00	-2.50	0.50	3.00

BLO1

2.00	1.00	1.00
3.00	4.00	1.00
5.00	16.00	1.00
2.00	1.00	1.00

CLO1

1.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00
2.33	-3.00	0.67	1.00

FORMA CANÔNICA CONTROLÁVEL II - FCCL2

CL2 ALC2 BLC2 CLC2

CL2

1.00	1.00	0.00	1.00
1.00	2.00	0.00	0.00
1.00	3.00	1.00	1.00
0.00	0.00	1.00	0.00

ALC2

0.00	-4.00	1.00	-2.00
1.00	4.00	-0.50	1.00
0.00	0.00	4.00	0.00
0.00	1.00	-0.50	2.00

BLC2

1.00	0.00	0.00
0.00	0.00	0.00
0.00	1.00	0.00
0.00	0.00	1.00

CLC2

2.00	3.00	1.00	1.00
2.00	5.00	1.00	1.00
1.00	3.00	2.00	1.00

FORMA CANÔNICA OBSERVÁVEL II - FCOL2

OL2 ALO2 BLO2 CLO2

OL2

1.00	1.00	0.00	1.00
1.00	2.00	0.00	4.00
0.00	1.00	1.00	0.00
0.00	0.00	1.00	1.00

ALO2

0.00	1.00	0.00	0.00
-3.50	4.50	-1.50	1.50
0.25	-0.25	2.25	0.75
-0.25	0.25	-0.25	3.25

BLO2

2.00	1.00	1.00
3.00	4.00	1.00
2.00	1.00	1.00
1.00	2.00	1.00

CLO2

1.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00

MATRIZES AUXILIARES DE FCCL3

C1 C1A SAC1 SAC1A

C1

0.50	-1.00	0.50	-0.50
0.00	0.00	0.00	1.00

C1A

0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00

SAC1

0.50	-1.00	0.50	-0.50
0.00	0.00	0.00	1.00
0.50	-2.00	1.50	-2.00
0.00	0.00	0.00	4.00
0.50	-4.00	4.50	-8.00
0.00	0.00	0.00	16.00
0.50	-8.00	13.50	-32.00
0.00	0.00	0.00	64.00

SAC1A

0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00
0.00	1.00	6.00	-0.50
0.00	0.00	0.00	4.00
1.00	6.00	25.00	-3.50
0.00	0.00	0.00	16.00
6.00	25.00	90.00	-18.50
0.00	0.00	0.00	64.00

MATRIZES DE TRANSFORMAÇÃO

CL3   CL3A

CL3

0.50	-1.00	0.50	-0.50
0.50	-2.00	1.50	-2.00
0.50	-4.00	4.50	-8.00
0.00	0.00	0.00	1.00

CL3A

0.00	0.00	1.00	0.00
0.00	1.00	6.00	-0.50
1.00	6.00	25.00	-3.50
0.00	0.00	0.00	1.00

FORMA CANÔNICA CONTROLÁVEL III - FCCL3

ALC3   BLC3   CLC3

ALC3

0.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00
6.00	-11.00	6.00	-3.00
0.00	0.00	0.00	4.00

BLC3

0.00	0.00	1.00
0.00	-0.50	2.00
1.00	-3.50	5.00
0.00	1.00	0.00

CLC3

9.00	-9.00	2.00	3.50
5.00	-7.00	2.00	4.50
2.00	-3.00	1.00	4.00

PRIMEIRA TRANSFORMAÇÃO

OL1A ALCO1 BLCO1 CLCO1

OL1A

1.00	6.00	25.00	-3.50
0.00	1.00	6.00	-0.50
0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00

ALCO1

6.00	-11.00	6.00	-3.00
1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	0.00	4.00

BL01

1.00	-3.50	0.50
0.00	-0.50	2.00
0.00	0.00	1.00
0.00	1.00	0.00

CLCO1

2.00	-9.00	9.00	3.50
2.00	-9.00	5.00	4.50
1.00	-3.00	2.00	4.00

TRANSFORMAÇÃO FINAL

P1 ALC3 BLC3 CLC3

P1

0.00	0.00	1.00	0.00
0.00	1.00	0.00	0.00
1.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00

ALC3

0.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00
6.00	-11.00	6.00	-3.00
0.00	0.00	0.00	4.00

BLC3

0.00	0.00	1.00
0.00	-0.50	2.00
1.00	-3.50	5.00
0.00	1.00	0.00

CLC3

9.00	-9.00	2.00	-3.50
5.00	-7.00	2.00	4.50
2.00	-3.00	1.00	4.00

MATRIZES AUXILIARES DE FCCL4

C2   C2A   SAC2   SAC2A

C2

-0.50	0.00	0.50	-0.50
0.00	0.00	0.00	1.00
0.50	-1.00	0.50	-0.50

C2A

0.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00

SAC2

-0.50	0.00	0.50	-0.50
0.00	0.00	0.00	1.00
0.50	-1.00	0.50	-0.50
-0.50	0.00	1.50	-2.00
0.00	0.00	0.00	4.00
0.50	-2.00	1.50	-2.00
-0.50	0.00	4.50	-8.00
0.00	0.00	0.00	16.00
0.50	-4.00	4.50	-8.00
-0.50	0.00	13.50	-32.00
0.00	0.00	0.00	64.00
0.50	-8.00	13.50	-32.00

SAC2A

0.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00
1.00	4.00	-0.50	1.00
0.00	0.00	4.00	0.00
0.00	1.00	-0.50	2.00
4.00	13.00	-3.50	4.00
0.00	0.00	16.00	0.00
1.00	6.00	-3.50	5.00
0.00	0.00	64.00	0.00
6.00	25.00	-18.50	14.00

MATRIZES DE TRANSFORMAÇÃO

CL4 CL4A

CL4

-0.50	0.00	0.50	-0.50
-0.50	0.00	1.50	-2.00
0.00	0.00	0.00	1.00
0.50	-1.00	0.50	-0.50

CL4A

0.00	1.00	0.00	0.00
1.00	4.00	-0.50	1.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00

FORMA CANÔNICA CONTROLÁVEL IV - FCCL4

ALC4 BLC4 CLC4

ALC4

0.00	1.00	0.00	0.00
-3.00	4.00	-1.50	0.00
0.00	0.00	4.00	0.00
1.00	0.00	-0.50	2.00

BLC4

0.00	0.00	0.00
1.00	-0.50	1.00
0.00	1.00	0.00
0.00	0.00	1.00

CLC4

-5.00	2.00	2.00	-1.00
-3.00	2.00	2.00	-1.00
-1.00	1.00	2.50	0.00

PRIMEIRA TRANSFORMAÇÃO

OL2A ALCO2 BLCO2 CLCO2

OL2A

1.00	4.00	-0.50	1.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00

ALCO2

4.00	-3.00	-1.50	0.00
1.00	0.00	0.00	0.00
0.00	0.00	4.00	0.00
0.00	1.00	-0.50	2.00

BLCO2

1.00	-0.50	1.00
0.00	0.00	0.00
0.00	1.00	0.00
0.00	0.00	1.00

CLCO2

2.00	-5.00	2.00	-1.00
2.00	-3.00	2.00	-1.00
1.00	-1.00	2.50	0.00

TRANSFORMAÇÃO FINAL

P2 ALC4 BLC4 CLC4

P2

0.00	1.00	0.00	0.00
1.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00

ALC4

0.00	1.00	0.00	0.00
-3.00	4.00	-1.50	0.00
0.00	0.00	4.00	0.00
1.00	0.00	-0.50	2.00

BLC4

0.00	0.00	0.00
1.00	-0.50	1.00
0.00	1.00	0.00
0.00	0.00	1.00

CLC4

-5.00	2.00	2.00	-1.00
-3.00	2.00	2.00	-1.00
-1.00	1.00	2.50	0.00

CB1 ABC1 BBC1 CBC1

CB1

1.00	0.00	1.00	1.00
1.00	0.00	0.00	2.00
1.00	1.00	1.00	3.00
0.00	1.00	0.00	0.00

ABC1

0.00	-1.00	-2.00	-4.00
0.00	4.00	0.00	0.00
0.00	-2.50	2.00	1.00
1.00	-0.50	1.00	4.00

BBC1

1.00	0.00	0.00
0.00	1.00	0.00
0.00	0.00	1.00
0.00	0.00	0.00

CBC1

2.00	1.00	1.00	3.00
2.00	1.00	1.00	5.00
1.00	2.00	1.00	3.00

FORMA CANÔNICA OBSERVÁVEL I - FCOB1

OB1 ABO1 BBO1 CBO1

OB1

1.00	1.00	0.00	1.00
0.00	1.00	1.00	0.00
0.00	0.00	1.00	1.00
1.00	2.00	0.00	4.00

ABO1

0.00	0.00	0.00	1.00
0.25	2.25	0.75	-0.25
-0.25	0.25	3.25	0.25
-3.50	-1.50	1.50	4.50

BBO1

2.00	1.00	1.00
2.00	1.00	1.00
1.00	2.00	1.00
3.00	4.00	1.00

CBO1

1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00

MATRIZES AUXILIARES DE FCCB2

C3 C3A SAC3 SAC3A

C3

0.00	0.00	0.00	1.00
0.50	-1.00	0.50	-0.50
-0.50	0.00	0.50	-0.50

C3A

0.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00

SAC3

0.00	0.00	0.00	1.00
0.50	-1.00	0.50	-0.50
-0.50	0.00	0.50	-0.50
0.00	0.00	0.00	4.00
5.00	-2.00	1.50	-2.00
-0.50	0.00	1.50	-2.00
0.00	0.00	0.00	16.00
0.50	-4.00	4.50	-8.00
0.00	0.00	0.00	64.00
0.50	-8.00	13.50	-32.00
-0.50	0.00	13.50	-32.00

SAC3A

0.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00
0.00	4.00	0.00	0.00
0.00	-0.50	2.00	1.00
1.00	-0.50	2.00	4.00
0.00	16.00	0.00	0.00
1.00	-3.50	5.00	6.00
4.00	-3.50	4.00	13.00
0.00	64.00	0.00	0.00
6.00	-18.50	14.00	25.00
13.00	-18.50	13.00	40.00

MATRIZES DE TRANSFORMAÇÃO

CB2 CB2A

CB2

0.00	0.00	0.00	1.00
0.50	-1.00	0.50	-0.50
-0.50	0.00	0.50	-0.50
-0.50	0.00	1.50	-2.00

CB2A

0.00	1.00	0.00	0.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00
1.00	-0.50	1.00	4.00

FORMA CANÔNICA CONTROLÁVEL II - FCCB2

ABC2 BBC2 CBC2

ABC2

4.00	0.00	0.00	0.00
-0.50	2.00	1.00	0.00
0.00	0.00	0.00	1.00
-1.50	0.00	-3.00	4.00

BBC2

0.00	1.00	0.00
0.00	0.00	1.00
0.00	0.00	0.00
1.00	-0.50	1.00

CBC2

2.00	-1.00	-5.00	2.00
2.00	-1.00	-3.00	2.00
2.50	0.00	-1.00	1.00

MATRIZES AUXILIARES DE FCCS1

ACL1A   C1A   SAC1S

ACL1A

0.00	0.00	6.00	0.00
1.00	0.00	-11.00	0.00
0.00	1.00	6.00	0.00
0.00	0.00	0.00	4.00

C1A

0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00

SAC1S

0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00
0.00	1.00	6.00	0.00
0.00	0.00	0.00	4.00
1.00	6.00	25.00	0.00
0.00	0.00	0.00	16.00
6.00	25.00	90.00	0.00
0.00	0.00	0.00	64.00

FORMA CANÔNICA ESPECIAL I - FCCS1

CS1   ASC1   BSC1   CSC1

CS1

0.00	0.00	1.00	0.00
0.00	1.00	6.00	0.00
1.00	6.00	25.00	0.00
0.00	0.00	0.00	1.00

ASC1

0.00	1.00	0.00	-0.50
0.00	0.00	1.00	-1.50
6.00	-11.00	6.00	-4.50
0.00	0.00	0.00	4.00

BSC1

0.00	0.00	1.00
0.00	0.00	2.00
1.00	0.00	5.00
0.00	1.00	0.00

CSC1

9.00	-9.00	2.00	1.00
5.00	-7.00	2.00	1.00
2.00	-3.00	1.00	2.00

MATRIZES DE TRANSFORMAÇÃO

CS2 CS2A CS2B

CS2

0.00	0.00	0.00	1.00
0.50	-1.00	0.50	-0.50
-0.50	0.00	0.50	-0.50
-0.50	0.00	1.50	-2.00

CS2A

0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00
0.00	1.00	0.00	0.00
1.00	4.00	-0.50	1.00

CS2B

0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00
1.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00

FORMA CANÔNICA CONTROLÁVEL ESPECIAL II - FCCS2

ASC2 BSC2 CSC2

ASC2

4.00	0.00	0.00	0.00
-0.50	2.00	1.00	0.00
0.00	0.00	0.00	1.00
-1.50	0.00	-3.00	4.00

BSC2

0.00	1.00	0.00
0.00	0.00	1.00
0.00	0.00	0.00
1.00	-0.50	1.00

CSC2

2.00	-1.00	-5.00	2.00
2.00	-1.00	-3.00	2.00
2.50	0.00	-1.00	1.00

MATRIZES AUXILIARES DE FCCS2

C2S C2SA C2SB SAC2S SAC2SA SAC2SB

C2S

0.00	0.00	0.00	1.00
0.50	-1.00	0.50	-0.50
-0.50	0.00	0.50	-0.50

C2SA

0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00
1.00	0.00	0.00	0.00

C2SB

0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00
1.00	0.00	0.00	0.00

SAC2S

0.00	0.00	0.00	1.00
0.50	-1.00	0.50	-0.50
-0.50	0.00	0.50	-0.50
0.00	0.00	0.00	4.00
0.50	-2.00	1.50	-2.00
-0.50	0.00	1.50	-2.00
0.00	0.00	0.00	16.00
0.50	-4.00	4.50	-8.00
0.00	0.00	0.00	64.00
0.50	-8.00	13.50	-32.00
-0.50	0.00	13.50	-32.00

SAC2SA

0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00
0.00	1.00	0.00	0.00
0.00	0.00	4.00	0.00
0.00	1.00	-0.50	2.00
1.00	4.00	-0.50	1.00
0.00	0.00	16.00	0.00
1.00	6.00	-3.50	5.00
4.00	13.00	-3.50	4.00
0.00	0.00	64.00	0.00
6.00	25.00	-18.50	14.00
13.00	40.00	-18.50	13.00

SAC2SB

0.00	0.00	1.00	0.00
0.00	0.00	0.00	1.00
1.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00
0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00

MATRIZES DO SISTEMA --- A B C

A

1.00	0.00	0.00	0.00
0.00	2.00	0.00	0.00
0.00	0.00	3.00	0.00
0.00	0.00	0.00	4.00

B

1.00	0.00	1.00
1.00	0.00	0.00
1.00	1.00	1.00
0.00	1.00	0.00

C

1.00	1.00	0.00	1.00
0.00	1.00	1.00	0.00
0.00	0.00	1.00	1.00

FUNÇÃO DE TRANSFERÊNCIA POR COLUNAS

PRIMEIRA COLUNA

7.00	15.00	-4.00
15.00	17.00	8.00
10.00	9.00	7.00
------	------	------
-6.00	11.00	-6.00

SEGUNDA COLUNA

1.00	11.00
1.00	10.00
2.00	21.00
------	------
-7.00	12.00

TERCEIRA COLUNA

1.00	0.00
1.00	16.00
1.00	16.00
------	------
-4.00	3.00

```
C         +*+*+*+*+*+*+*+*+*+*
C                 FRCN2
C         +*+*+*+*+*+*+*+*+*+*
          SUBROUTINE FRCN2(S,H,M,N,MD,NA,NB,IV,II,IRC,IECL,NW,NNW,EPS,IERR)
          DIMENSION H(1),IRC(1),IECL(1)
          IERR=0
C
C         TESTES DE DIMENSOES
C
          IF(MD.EQ.0)MD=M
          IF(MD.LT.M.OR.M.LT.NA)GO TO 23
          IF(N.LT.(NA+NB).OR.NA.LE.0)GO TO 23
          IF(NW.LT.1.OR.NB.LE.0)GO TO 23
          IF(NA.EQ.1)RETURN
C
C         INICIALIZACAO
C
          IV=0
          JB=1
          I=1
          J=NA+1
          IC=(NA)*MD
          DO 20 KA=1,NB
          IECL(KA)=0
20        IRC(KA)=0
11        IX=1
          JX=0
          GO TO 1
C
C         PROCURA DO MELHOR PIVOT
C
12        IC=(I-2)*MD
1         KB=IC+I+1
          IF(NNW.NE.0)CALL SAIDA(H,NA+NB,NA,MD,NW,
          1'F1',14,'                 MATRIZ <A!B>')
          KC=IC+NA
          G=ABS(H(KB-1))
          L=I
          IF(I.EQ.NA)GO TO 334
          DO 2 KA=KB,KC
          IF(ABS(H(KA)).LE.G) GO TO 2
          G=ABS(H(KA))
          L=KA-IC
2         CONTINUE
          G=H(IC+L)
C
C         TESTE DE CONTROLABILIDADE
C
334       IF(ABS(G)-EPS)333,333,3
333       IF(II.NE.'FUTR')GO TO 51
          IRC(JB)=JX
          GO TO 50
51        IF(JB.EQ.NB) GO TO 4
          JB=JB+1
          IC=(NA+JB-1)*MD
          IF(IX.EQ.1)GO TO 1
C
C         INDICES IRC E IECL
C
          IRC(JB-1)=JX
```

```
      IECL(JB-1)=J
      GO TO 11
3     IF(IX.EQ.1)IV=IV+1
      JX=JX+1
C
C     COLOCANDO O PIVOT NA LINHA I E COLUNA J
C
      CALL PERMUT($23,H,M,N,MD,I,L,1HS,1HL,IERR)
      CALL MULTIP($23,H,M,N,MD,I,G,1HS,1HC,EPS,IERR)
C
C     ELIMINACAO POR COLUNAS
C
      IF(I.EQ.NA)GO TO 7
      DO 9 KA=(I+1),NA
      CALL COMBLI($23,H,M,N,MD,KA,I,-H(IC+KA),1HS,1HL,EPS,IERR)
9     CONTINUE
7     IF(IX.EQ.1)GO TO 25
      IF(II.EQ.'CL1A'.OR.II.EQ.'CS1A'.OR.II.EQ.'CL3A')GO TO 25
      GO TO 8
25    IF(I.EQ.1)GO TO 8
      DO 10 KA=1,(I-1)
      CALL COMBLI($23,H,M,N,MD,KA,I,-H(IC+KA),1HS,1HL,EPS,IERR)
10    CONTINUE
8     IX=0
C
C     TESTE DE FIM DE FASE OU DE FIM DE ALGORITMO
C
      IF(I.EQ.NA)GO TO 19
      J=I
      I=I+1
      GO TO 12
19    IRC(JB)=JX
      IECL(JB)=NA
      IF(II.EQ.'CL1A')GO TO 13
C
C     SEGUNDA FASE
C
50    IF(I.EQ.1)RETURN
      JJ=I
22    J=I-1
      IF(NNW.NE.0)CALL SAIDA(H,NA+NB,NA,MD,NW,
     1'F1',14,'                    MATRIZ <A!B>')
C
C     VERIFICANDO SE ESTAMOS SOBRE AS LINHAS ESTRUTURAIS
C
      IF(H((J-1)*MD+I).NE.0)GO TO 21
      IF(II.EQ.'CS1A'.OR.II.EQ.'CS1B')JJ=J
      GO TO 16
C
C     ELIMINACAO SOBRE LINHAS
C
21    DO 18 KA=JJ,I,-1
18    CALL COMBLI($23,H,M,N,MD,KA,I-1,-H((KA-1)*MD+I),1HS,1HC,EPS
     1,IERR)
C
C     TESTE DE FIM DE ALGORITMO
C
16    IF(I.EQ.2)GO TO 14
      I=I-1
      GO TO 22
```

```
C
C       INDICES ESTRUTURAIS
C
14      DO 15 KA=1,NB
        IF(IRC(KA).EQ.0)GO TO 15
        IECL(KA)=IECL(KA)-IRC(KA)+1
15      CONTINUE
C
C       SAIDA DOS INDICES; RELATIVOS E ESTRUTURAIS
C
13      IF(NNW.NE.0)CALL SAIDI(IECL,0,NB,0,NW,
        1'I1',19,'                INDICE ESTRUTURAL')
        IF(NNW.NE.0)CALL SAIDI(IECL,0,NB,0,NW,
        1'I1',17,'                INDICE RELATIVO')
        RETURN
C
C       SAIDA SE O SISTEMA NAO FOR CONTROLAVEL
C
4       KB=NA-I+1
        WRITE(NW,5)KB
5       FORMAT(//,5X,'O SISTEMA APRESENTADO TEM',I3,'  MODOS NAO ',/,5X,
     *  'CONTROLAVEIS')
        RETURN 1
C
C       SAIDA DA SUBROTINA EM CASO DE ERRO DE PARAMETROS DIMENSIONAIS
C
23      IF(IERR.EQ.0)IERR='FRCN'
        RETURN 1
        END
```

```
C         +*+*+*+*+*+*+*+*+*+*+*
C                FRCN1
C         +*+*+*+*+*+*+*+*+*+*+*
          SUBROUTINE FRCN1($,H,M,N,MD,NA,NB,IV,II,IRC,IECL,NW,NNW,EPS,IERR)
          DIMENSION H(1),IRC(1),IECL(1)
C
C         TESTES DE DIMENSOES
C
          IF(MD.EQ.0)MD=M
          IF(MD.LT.M.OR.M.LT.NA)GO TO 25
          IF(N.LT.(NA+NB).OR.NA.LE.0)GO TO 25
          IF(NW.LT.1.OR.NB.LE.0)GO TO 25
          IF(NA.EQ.1)RETURN
C
C         INICIALIZACAO
C
          IV=0
          JB=1
          I=1
          J=NA+1
          DO 20 KA=1,NB
          IECL(KA)=0
 20       IRC(KA)=0
          IX=1
          JX=0
          GO TO 1
C
C         TESTE DE CONTROLABILIDADE
C
 24       IF(IX.EQ.1)GO TO 1
 21       IF(JB.EQ.NB)JX=0
 8        DO 11 JB=(JX+1),NB
          IF(IRC(JB).GT.0)GO TO 1
 11       CONTINUE
          IF(JX.EQ.0)GO TO 4
          JX=0
          GO TO 8
C
C         PROCURA DO MELHOR PIVOT
C
 1        G=ABS(H((J-1)*MD+I))
          IF(NNW.NE.0)CALL SAIDA(H,NA+NB,NA,MD,NW,
         1'F1',14,'                 MATRIZ <A!B>')
          L=I
          IF(I.EQ.NA)GO TO 12
          DO 2 KA=I+1,NA
          IF(ABS(H((J-1)*MD+KA)).LE.G) GO TO 2
          G=ABS(H((J-1)*MD+KA))
          L=KA
 2        CONTINUE
          G=H((J-1)*MD+L)
C
C         TESTE DE DEPENDENCIA LINEAR
C
 12       IF(ABS(G)-EPS)27,27,84
 27       IF(IX.EQ.0)GO TO 19
          IF(J.EQ.(NA+NB).AND.IV.EQ.0)GO TO 4
          IF(J.EQ.(NA+NB))GO TO 18
          J=J+1
          JB=JB+1
```

```
      GO TO 1
18    IX=0
      J=1
      GO TO 8
C
C     INDICES IRC E IECL
C
19    IRC(JB)=-IRC(JB)
C     IF(II.NE.'CB1A')GO TO 38
      IECL(JB)=J
38    J=J+1
      JX=JB
      GO TO 21
84    IF(IX.EQ.0)GO TO 22
      IV=IV+1
      IF(II.EQ.'CL4A'.OR.II.EQ.'CL4B')IECL(JB)=IV
      IF(II.EQ.'CS4A'.OR.II.EQ.'CS4B')IECL(JB)=IV
      IF(II.EQ.'CS3A'.OR.II.EQ.'CS3B')IECL(JB)=IV
      IF(II.EQ.'CS5A'.OR.II.EQ.'CS5B')IECL(JB)=IV
      IRC(JB)=1
      JB=JB+1
      GO TO 3
C
C     CONTANDO IRC
C
22    IRC(JB)=IRC(JB)+1
C
C     COLOCANDO O PIVOT NA LINHA I E COLUNA J
C
3     CALL PERMUT($25,H,M,N,MD,I,L,1HS,1HL,IERR)
      CALL MULTIP($25,H,M,N,MD,I,G,1HS,1HC,EPS,IERR)
C
C     ELIMINACAO POR COLUNAS
C
      IF(I.EQ.NA)GO TO 7
      DO 9 KA=(I+1),NA
      CALL COMBLI($25,H,M,N,MD,KA,I,-H((J-1)*MD+KA),1HS,1HL,EPS,IERR)
9     CONTINUE
7     IF(IX.EQ.1)GO TO 82
      IF(II.EQ.'CL2A'.OR.II.EQ.'CB1A'.OR.II.EQ.'CL4A')GO TO 29
      IF(II.EQ.'CB2A'.OR.II.EQ.'CS2A')GO TO 29
      IF(II.EQ.'CS3A'.OR.II.EQ.'CS4A')GO TO 29
      IF(II.EQ.'CS5A')GO TO 29
      GO TO 26
29    IF(I.EQ.1)GO TO 26
      DO 10 KA=J+1,I-1
      CALL COMBLI($25,H,M,N,MD,KA,I,-H((J-1)*MD+KA),1HS,1HL,EPS,IERR)
10    CONTINUE
      DO 83 KA=1,J
      CALL COMBLI($25,H,M,N,MD,KA,I,-H((J-1)*MD+KA),1HS,1HL,EPS,IERR)
83    CONTINUE
      GO TO 26
82    IF(I.EQ.1)GO TO 26
      DO 85 KA=1,I-1
      CALL COMBLI($25,H,M,N,MD,KA,I,-H((J-1)*MD+KA),1HS,1HL,EPS,IERR)
85    CONTINUE
C
C     TESTE DE FIM DE FASE OU DE FIM DE ALGORITMO
C
26    IF(I.EQ.NA)GO TO 28
```

```
      JX=JB
      I=I+1
      IF(J.EQ.(NA+NB))GO TO 23
      J=J+1
      GO TO 24
23    J=1
      IX=0
      GO TO 21
C
C     FINALIZACAO PARA IRC E IECL (PARA PRIMEIRA FASE)
C
28    LI=J
30    JX=JB
      IF(JB.EQ.NB)JX=0
      DO 32 JB=JX+1,NB
      IF(IRC(JB).GT.0)GO TO 33
32    CONTINUE
      IF(JX.EQ.0)GO TO 34
35    JX=0
      GO TO 30
33    J=J+1
      IRC(JB)=-IRC(JB)
      IF(II.EQ.'CS2A'.OR.II.EQ.'CS2B')IECL(JB)=J
      IF(II.EQ.'CB2A'.OR.II.EQ.'CB2B')IECL(JB)=J
      IF(II.EQ.'CB1A'.OR.II.EQ.'CL2A')IECL(JB)=J
      IF(JB.EQ.NB)GO TO 35
      JX=JB
      GO TO 30
34    DO 36 JB=1,NB
36    IRC(JB)=-IRC(JB)
C
C     FINALIZACAO PARA FCCB1
C
      IF(II.EQ.'CB1A')GO TO 13
C
C     FINALIZACAO PARA FCCL2
C
      IF(II.NE.'CL2A')GO TO 37
      KI=0
      KB=0
      I=0
108   DO 100 JB=KI+1,NB
      IF(IRC(JB).NE.0)GO TO 101
100   CONTINUE
      GO TO 13
101   IF(I.EQ.NA)GO TO 13
      I=I+1
      DO 102 KA=KB+1,NA
      IF(H((NA+JB-1)*MD+KA).EQ.1.)GO TO 103
102   CONTINUE
103   CALL PERMUT(S25,H,M,N,MD,KA,I,1HS,1HL,IERR)
      KK=0
      DO 104 KJ=1,NB
      IF(IECL(KJ).EQ.I)GO TO 105
104   CONTINUE
      GO TO 107
105   IF(IECL(KJ).EQ.KA)GO TO 112
      IECL(KJ)=KA
      KK=KJ
107   DO 106 KJ=1,NB
```

```
      IF(KK.EQ.KJ)GO TO 106
      IF(IECL(KJ).NE.KA)GO TO 106
      IECL(KJ)=I
112   KI=JB
      KB=IECL(KJ)
      GO TO 108
106   CONTINUE
      I=I+1
      DO 109 KA=I,NA
      IF(H((I-2)*MD+KA).EQ.1.)GO TO 103
109   CONTINUE
      GO TO 13
C
C     SEGUNDA FASE
C
C
C     OBTENDO OS PIVOTS DO BLOCO
C
37    LF=NA
56    IF(LI.LT.IV)LI=IV
      IF(NNW.NE.0)CALL SAIDA(H,NA+NB,NA,MD,NW,
      1'F1',14,'                  MATRIZ <A!B>')
      JP=0
      DO 41 I=LI+1,LF
      DO 42 J=JP+1,NA
      IF(H((J-1)*MD+I).EQ.1.)GO TO 43
42    CONTINUE
C
C     ELIMINACAO NO BLOCO : OPERANDO POR COLUNAS
C
43    JP=J
      DO 41 KJ=J+1,NA
      DO 41 KI=1,M
C
C     EVITANDO AS LINHAS JA CHAMADAS
C
      IF(KI.GT.LI.AND.KI.LE.NA)GO TO 41
      H((KJ-1)*MD+KI)=H((KJ-1)*MD+KI)-H((KJ-1)*MD+I)*H((J-1)*MD+KI)
41    CONTINUE
C
C     ELIMINACAO NO BLOCO : OPERANDO POR LINHAS
C
C
C     OBTENDO OS PIVOTS DO BLOCO NOVAMENTE
C
      JJ=0
      DO 46 I=LI+1,LF
      DO 47 J=JJ+1,NA
      IF(H((J-1)*MD+I).EQ.1.)GO TO 48
47    CONTINUE
C
C     ATUANDO SOBRE T
C
48    JJ=J
      KP=0
      DO 49 KI=J+1,NA
      AA=H((KI-1)*MD+I)
      IF(ABS(AA-1.)-EPS)91,91,92
91    AA=1.
92    IF(ABS(AA+1.)-EPS)93,93,94
```

```
93        AA=-1.
94        IF(ABS(AA)-EPS)96,96,95
95        IF((NA+NB).GE.N)GO TO 52
          DO 50 KJ=(NA+NB+1),N
50        H((KJ-1)*MD+J)=H((KJ-1)*MD+J)+AA*H((KJ-1)*MD+KI)
C
C         ATUANDO SOBRE A E B
C
52        IF((KI.LE.IV).AND.(II.EQ.'CS3A'.OR.II.EQ.'CS3B'))GO TO 49
          IF((KI.LE.IV).AND.(II.EQ.'CS5A'.OR.II.EQ.'CS5B'))GO TO 49
          IF(KI.GT.LI)GO TO 51
          DO 53 KJ=1,NA+NB
53        H((KJ-1)*MD+J)=H((KJ-1)*MD+J)+AA*H((KJ-1)*MD+KI)
          H((KI-1)*MD+I)=0.
          GO TO 49
C
C         ENCONTRANDO O PIVOT DAS LINHAS JA CHAMADAS
C
51        DO 55 LJ=KP+1,NA
          IF(H((LJ-1)*MD+KI).EQ.1.)GO TO 54
55        CONTINUE
54        H((LJ-1)*MD+J)=H((LJ-1)*MD+J)+AA
          KP=LJ
96        H((KI-1)*MD+I)=0.
49        CONTINUE
46        CONTINUE
          IF(LI.EQ.IV)GO TO 58
C
C         OBTENDO O PROXIMO BLOCO
C
          LF=LI
          DO 44 J=1,NA
          IF(H((J-1)*MD+LI).EQ.1.)GO TO 45
44        CONTINUE
45        LI=J
          GO TO 56
C
C         FINALIZACAO PARA FCCS3
C
58        IF(II.EQ.'CS3A'.OR.II.EQ.'CS3B')GO TO 13
C
C         FINALIZACAO PARA FCCS4
C
          IF(II.EQ.'CS4A'.OR.II.EQ.'CS4B')GO TO 13
C
C         FINALIZACAO PARA FCCL4
C
          IF(II.EQ.'CL4A'.OR.II.EQ.'CL4B')GO TO 70
C
C         FINALIZACAO PARA FCCS5
C
          IF(II.EQ.'CS5A'.OR.II.EQ.'CS5B')GO TO 70
C
C         OBTENDO FCCB2 OU FCCS2  (ORDEM CRESCENTE)
C
          I=0
          J=1
          KB=0
68        IF(KB.EQ.NA)GO TO 63
C
```

```
C         ORDENANDO PELAS COLUNAS ESTRUTURAIS DE A
C
          KB=KB+1
          DO 61 KA=1,NB
          IF(IECL(KA).EQ.KB)GO TO 67
 61       CONTINUE
          GO TO 68
 67       IECL(KA)=0
          KJ=KB
C
C         ORDENACAO
C
 66       CALL PERMUT($25,H,M,N,MD,KJ,J,1HS,1HL,IERR)
          J=J+1
          GO TO 68
 63       I=I+1
C
C         ORDENANDO COMO SE FOSSE FCOB1
C
          IF(J.GT.NA)GO TO 62
          DO 65 KJ=J,NA
          IF(H((KJ-1)*MD+I).EQ.1.)GO TO 66
 65       CONTINUE
 62       K=0
C
C         INDICE IECB
C
          DO 64 KA=1,NB
          IF(IRC(KA).EQ.0.OR.IECL(KA).NE.0)GO TO 64
C
C         OBTENDO A ORDENACAO CRESCENTE
C
          IF(K.NE.0.AND.IRC(KA).GE.K)GO TO 64
          K=IRC(KA)
          KI=KA
 64       CONTINUE
C
C         FINALIZANDO O INDICE IECB
C
          IECL(KI)=I
          IF(I.EQ.NA)GO TO 69
          GO TO 63
C
C         FINALIZACAO PARA FCCB2
C
 69       IF(II.EQ.'CB2A'.OR.II.EQ.'CB2B')GO TO 13
C
C         OBTENDO FCCS2 OU FCCL4 OU FCCL2 OU FCCS5
C
 70       I=1
          GO TO 73
 72       CALL PERMUT($25,H,M,N,MD,KA,I,1HS,1HL,IERR)
C
C         ATUALIZANDO OS INDICES IECL
C
          DO 97 JB=1,NB
          IF(IECL(JB).EQ.I)GO TO 98
 97       CONTINUE
          GO TO 73
 98       IECL(JB)=KA
```

```
73      IF(I.EQ.NA)GO TO 76
        I=I+1
79      DO 74 KA=I,NA
C
C       VERIFICANDO SE ESTAMOS SOBRE AS LINHAS ESTRUTURAIS
C
        DO 75 KB=1,NB
        IF(IECL(KB).EQ.KA)GO TO 74
75      CONTINUE
        IF(H((I-2)*MD+KA).EQ.1.)GO TO 72
74      CONTINUE
        DO 77 KA =I,NA
        DO 77 KB=1,NB
        IF(IECL(KB).NE.KA)GO TO 77
        IECL(KB)=I
        CALL PERMUT($25,H,M,N,MD,KA,I,1HS,1HL,IERR)
        GO TO 73
77      CONTINUE
C
C       FINALIZACAO PARA FCCS2
C
76      IF(II.EQ.'CS2A'.OR.II.EQ.'CS2B')GO TO 13
C
C       SAIDA DOS INDICES; RELATIVOS E ESTRUTURAIS
C
13      IF(NNW.NE.0)CALL SAIDI(IECL,0,NB,0,NW,
        1'I1',19,'                 INDICE ESTRUTURAL')
        IF(NNW.NE.0)CALL SAIDI(IRC,0,NB,0,NW,
        1'I1',17,'                 INDICE RELATIVO')
        RETURN
C
C       SAIDA SE O SISTEMA NAO FOR CONTROLAVEL
C
4       KB=NA-I+1
        WRITE(NW,5)KB
5       FORMAT(//,5X,'O SISTEMA APRESENTADO TEM',I3,'  MODOS NAO ',/,5X,
     *  'CONTROLAVEIS')
        RETURN 1
C
C       SAIDA DA SUBROTINA EM CASO DE ERRO DE PARAMETROS DIMENSIONAIS
C
25      IF(IERR.NE.0)RETURN 1
        IERR='FRCN'
        RETURN 1
        END
```

```
C         +*******************
C                 PERMUT
C         +*******************
          SUBROUTINE PERMUT($,A,M,N,MD,I,J,IT,IC,IERR)
          DIMENSION A(1)
          IERR=0
          IF(M.LE.0.OR.N.LE.0.OR.I.LE.0.OR.J.LE.0)GO TO 5
          IF(MD.EQ.0)MD=M
          IF(MD.LT.M)GO TO 5
          IF(IT.NE.1HS)GO TO 6
          IF(I.GT.M.OR.J.GT.M)GO TO 5
8         IF(I.GT.N.OR.J.GT.N)GO TO 5
          GO TO 7
6         IF(IC.EQ.1HC)GO TO 8
          IF(I.GT.M.OR.J.GT.M)GO TO 5
7         IF(I.EQ.J)RETURN
          IF(IT.EQ.'S')GO TO 2
          IF(IC.EQ.1HC)GO TO 3
2         DO 1 KA=1,N
          C=A((KA-1)*MD+I)
          A((KA-1)*MD+I)=A((KA-1)*MD+J)
1         A((KA-1)*MD+J)=C
          IF(IT.NE.1HS)RETURN
3         DO 4 KA=1,M
          C=A((I-1)*MD+KA)
          A((I-1)*MD+KA)=A((J-1)*MD+KA)
4         A((J-1)*MD+KA)=C
          RETURN
5         IERR=1
          RETURN 1
          END
```

```
C         +*+*+*+*+*+*+*+*+*+*+*
C                 MULTIP
C         +*+*+*+*+*+*+*+*+*+*+*
          SUBROUTINE MULTIP(S,A,M,N,MD,I,AA,IT,IC,EPS,IERR)
          DIMENSION A(1)
          IERR=0
          IF(M.LE.0.OR.N.LE.0.OR.I.LE.0)GO TO 7
          IF(MD.EQ.0)MD=M
          IF(MD.LT.M)GO TO 7
          IF(IT.NE.1HS)GO TO 8
          IF(ABS(AA)-EPS)7,7,6
6         IF(I.GT.M.OR.I.GT.N)GO TO 7
8         IF(IC.NE.1HC)GO TO 9
          IF(IT.GT.N)GO TO 7
          GO TO 10
9         IF(I.GT.M)GO TO 7
10        IF(ABS(AA-1)-EPS)1,1,11
11        BB=AA
          IF(ABS(AA+1)-EPS)13,13,14
13        BB=-1.
14        IF(IC.EQ.'C')GO TO 2
5         DO 3 KA=1,N
3         A((KA-1)*MD+I)=A((KA-1)*MD+I)*BB
          IF(IT.NE.1HS)RETURN
          IF(IC.EQ.1HC)RETURN
          BB=1./BB
2         DO 4 KA=1,M
4         A((I-1)*MD+KA)=A((I-1)*MD+KA)*BB
          IF(IT.NE.1HS)RETURN
          IF(IC.NE.'C')RETURN
          BB=1./BB
          GO TO 5
1         RETURN
7         IERR=0
          RETURN 1
          END
```

```
C         +*+*+*+*+*+*+*+*+*+*+*
C                 COMBLI
C         +*+*+*+*+*+*+*+*+*+*+*
          SUBROUTINE COMBLI(S,A,M,N,MD,IA,JA,AA,IT,IC,EPS,IERR)
          DIMENSION A(1)
          IERR=0
          IF(M.LE.0.OR.N.LE.0.OR.IA.LE.0.OR.JA.LE.0)GO TO 6
          IF(MD.EQ.0)MD=M
          IF(MD.LT.M)GO TO 6
          IF(IT.NE.1HS)GO TO 7
          IF(IA.GT.M.OR.JA.GT.M)GO TO 6
9         IF(IA.GT.N.OR.JA.GT.N)GO TO 6
          GO TO 8
7         IF(IC.EQ.1HC)GO TO 9
          IF(IA.GT.M.OR.JA.GT.M)GO TO 6
8         IF(ABS(AA)-EPS)1,1,10
10        I=IA
          J=JA
          BB=AA
          IF(ABS(AA-1)-EPS)11,11,12
11        BB=1.
12        IF(ABS(AA+1)-EPS)13,13,14
13        BB=-1.
14        IF(IC.EQ.'C')GO TO 2
5         DO 3 KA=1,N
3         A((KA-1)*MD+I)=A((KA-1)*MD+I)+BB*A((KA-1)*MD+J)
          IF(IT.NE.1HS)RETURN
          IF(IC.EQ.1HC)RETURN
          BB=-BB
          II=I
          I=J
          J=II
2         DO 4 KA=1,M
4         A((I-1)*MD+KA)=A((I-1)*MD+KA)+A((J-1)*MD+KA)*BB
          IF(IT.NE.'S')RETURN
          IF(IC.NE.'C')RETURN
          BB=-BB
          II=I
          I=J
          J=II
          GO TO 5
1         RETURN
6         IERR=1
          RETURN 1
          END
```

ENTRE COM O TIPO DA MATRIZ DO SISTEMA: * 6

MATRIZ TOTAL

0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
1.00	2.00	1.00	2.00	1.00	0.00	1.00	2.00	0.00	3.00	0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	4.00	2.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	2.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCL1A

MATRIZ TOTAL

0.00	0.00	1.00	2.00	0.00	3.00	1.00	0.00	0.00	3.00	-2.00	-1.00	1.00	0.00	0.00	-0.50
1.00	0.00	2.00	-8.00	0.00	-15.00	0.00	0.00	0.00	0.00	-1.00	1.00	0.00	-2.00	0.00	2.00
0.00	1.00	1.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	4.00	0.00	8.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	-1.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50

0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	1.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	3.00	2.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCS1A

MATRIZ TOTAL

1.00	2.00	1.00	0.00	0.00	-12.00	1.00	0.00	0.00	3.00	0.00	0.00	1.00	-2.00	0.00	1.50
1.00	0.00	0.00	-6.00	0.00	-15.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	-2.00	0.00	2.00
0.00	1.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	4.00	0.00	8.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	-1.00
0.00	0.00	0.00	0.00	1.00	2.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.50
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50

0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	2.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	-1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCS1B

MATRIZ TOTAL

1.00	2.00	1.00	0.00	-3.00	-1.00	1.00	0.00	0.00	3.00	0.00	0.00	1.00	-2.00	0.00	0.00
1.00	0.00	0.00	-6.00	-4.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	-2.00	0.00	0.00
0.00	1.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	4.00	2.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	2.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.50
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50

0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	-1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCL3A

MATRIZ TOTAL

1.00	2.00	1.00	2.00	1.00	3.00	1.00	2.00	0.00	3.00	0.00	0.00	1.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	4.00	0.00	8.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	-1.00
0.00	0.00	0.00	0.00	1.00	2.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.50
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	-1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCL30

MATRIZ TOTAL

1.00	2.00	1.00	2.00	1.00	-1.00	1.00	2.00	0.00	3.00	0.00	0.00	1.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	4.00	2.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	2.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.50
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	-1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCB1A

MATRIZ TOTAL

0.00	0.00	0.00	-1.00	8.00	3.00	1.00	0.00	0.00	3.00	-3.00	-1.00	1.00	0.00	0.00	-0.50
0.00	0.00	0.00	-2.00	12.00	8.00	0.00	1.00	0.00	0.00	-2.00	0.00	0.00	1.00	0.00	-1.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.00
1.00	0.00	0.00	4.00	-18.00	-15.00	0.00	0.00	0.00	0.00	3.00	1.00	0.00	-2.00	0.00	2.00
0.00	1.00	0.00	0.50	1.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50
0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	2.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	2.00	0.00	1.00	8.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	4.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCL2A

MATRIZ TOTAL

0.00	-1.00	0.00	8.00	0.00	3.00	1.00	0.00	0.00	3.00	-3.00	-1.00	1.00	0.00	0.00	-0.50
1.00	4.00	0.00	-18.00	0.00	-15.00	0.00	0.00	0.00	0.00	3.00	1.00	0.00	-2.00	0.00	2.00
0.00	-2.00	0.00	12.00	0.00	8.00	0.00	1.00	0.00	0.00	-2.00	0.00	0.00	1.00	0.00	-1.00
0.00	0.50	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50

0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	2.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	2.00	8.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	4.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCS4A

MATRIZ TOTAL

4.00	-18.00	-15.00	-1.00	3.00	3.00	1.00	0.00	0.00	3.00	0.00	3.00	1.00	-8.00	0.00	0.00
0.50	1.00	0.00	-2.00	12.00	8.00	0.00	1.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	2.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.50
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.00	1.00	0.00	-2.00	0.00	2.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50

0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	2.00	0.00	-4.00	24.00	16.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	-0.50	3.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	-1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCS4B

MATRIZ TOTAL

1.00	-18.00	-15.00	11.00	2.00	-9.00	1.00	-6.00	-4.00	-5.00	0.00	0.00	1.00	-8.00	-4.00	-2.00
0.50	4.00	2.00	-2.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	2.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.50
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	-2.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50
0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	6.00	4.00	-4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	-0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	-1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCB2A

MATRIZ TOTAL

0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	3.00	1.00	0.00	-2.00	0.00	2.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50
-1.00	8.00	3.00	4.00	-10.00	-15.00	1.00	0.00	0.00	3.00	0.00	3.00	1.00	-8.00	0.00	0.00
-2.00	12.00	8.00	0.50	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00
0.00	0.00	2.00	0.00	0.00	1.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.50
0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-4.00	24.00	16.00	1.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-0.50	3.00	2.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	-1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCB2B

MATRIZ TOTAL

0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	-2.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50
11.00	2.00	-9.00	1.00	-18.00	-15.00	1.00	-6.00	-4.00	-5.00	0.00	0.00	1.00	-8.00	-4.00	-2.00
-2.00	0.00	0.00	0.50	4.00	2.00	0.00	1.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00
0.00	0.00	2.00	0.00	0.00	1.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.50
0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-4.00	0.00	0.00	1.00	8.00	4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
-0.50	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	-1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCL4A

MATRIZ TOTAL

4.00	-1.00	-18.00	8.00	-15.00	3.00	1.00	0.00	0.00	3.00	0.00	3.00	1.00	-8.00	0.00	0.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.00	1.00	0.00	-2.00	0.00	2.00
0.50	-2.00	1.00	12.00	0.00	8.00	0.00	1.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	2.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.50
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50
0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	-4.00	2.00	24.00	0.00	16.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	-0.50	1.00	3.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	-1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCL4B

MATRIZ TOTAL

1.00	11.00	-18.00	2.00	-15.00	-9.00	1.00	-6.00	-4.00	-5.00	0.00	0.00	1.00	-8.00	-4.00	-2.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	-2.00	0.00	0.00
0.50	-2.00	4.00	0.00	2.00	0.00	0.00	1.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	2.00	0.00	0.00	1.00	2.00	0.00	0.00	0.00	0.00	1.00	0.50
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50

0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	-4.00	8.00	0.00	4.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	-0.50	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	-1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

ENTRE COM O TIPO DA MATRIZ DO SISTEMA: * 6

MATRIZ TOTAL

1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	2.00	0.00	0.00	0.00	0.00	1.00	0.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
0.00	0.00	3.00	0.00	0.00	0.00	1.00	0.00	1.00	1.00	0.00	0.00	1.00	0.00	0.00	0.00
0.00	0.00	0.00	4.00	0.00	0.00	0.00	1.00	1.00	1.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	6.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCL4A

MATRIZ TOTAL

0.00	-11.00	6.00	0.00	24.00	-84.00	1.00	0.00	1.00	3.00	0.50	-4.00	4.50	0.00	-37.50	36.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	-2.00	1.50	0.00	-7.50	6.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	-1.00	0.50	0.00	-1.50	1.00
0.00	0.00	0.00	4.00	0.00	2.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	1.00	-2.00	1.00
0.00	0.00	0.00	0.00	11.00	-30.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	-5.00	6.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-1.00	1.00

1.00	-5.00	6.00	0.00	-3.00	6.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	-4.00	3.00	0.00	-3.00	3.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	-3.00	2.00	0.00	-2.00	-8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.00	1.00	-7.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	-6.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.00	-5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCS2A

MATRIZ TOTAL

4.00	0.00	2.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	1.00	-2.00	1.00
0.00	11.00	-30.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	-5.00	6.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-1.00	1.00
0.00	24.00	-64.00	6.00	-11.00	6.00	1.00	0.00	1.00	3.00	0.50	-4.00	4.50	0.00	-37.50	36.00
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	-2.00	1.50	0.00	-7.50	6.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.50	-1.00	0.50	0.00	-1.50	1.00

0.00	-3.00	6.00	1.00	-5.00	6.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	-3.00	3.00	1.00	-4.00	3.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	-2.00	-8.00	1.00	-3.00	2.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	-7.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	-6.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	-5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCL4B

MATRIZ TOTAL

6.00	13.00	-78.00	0.00	24.00	-84.00	1.00	0.00	1.00	3.00	0.50	-4.00	4.50	0.00	-37.50	36.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	-2.00	1.50	0.00	-7.50	6.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	-1.00	0.50	0.00	-1.50	1.00
-2.00	19.00	-42.00	4.00	7.00	-26.00	0.00	1.00	1.00	0.00	1.00	0.00	-1.00	1.00	0.00	0.00
-1.00	11.00	-30.00	0.00	11.00	-30.00	0.00	0.00	0.00	1.00	-0.50	2.00	-1.50	0.00	2.50	0.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	-0.50	1.00	-0.50	0.00	0.50	0.00
1.00	-6.00	12.00	0.00	-3.00	6.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	-7.00	6.00	0.00	-3.00	3.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	-5.00	-6.00	0.00	-2.00	-8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	3.00	-18.00	1.00	1.00	-14.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	-6.00	0.00	1.00	-6.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	-5.00	0.00	1.00	-5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

FORMA CANONICA: FCCS2B

MATRIZ TOTAL

4.00	7.00	-26.00	-2.00	19.00	-42.00	0.00	1.00	1.00	0.00	1.00	0.00	-1.00	1.00	0.00	0.00
[illegible]	11.00	-30.00	-1.00	11.00	-30.00	0.00	0.00	0.00	1.00	-0.50	2.00	-1.50	0.00	2.50	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.50	1.00	-0.50	0.00	0.50	0.00
[illegible]	24.00	-84.00	6.00	13.00	-78.00	1.00	0.00	1.00	3.00	0.50	-4.00	4.50	0.00	-37.50	36.00
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	-2.00	1.50	0.00	-7.50	6.00
0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.50	-1.00	0.50	0.00	-1.50	1.00
0.00	-3.00	6.00	1.00	-8.00	12.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	-3.00	5.00	1.00	-7.00	6.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	-2.00	-8.00	1.00	-5.00	-6.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	1.00	-14.00	0.00	3.00	-18.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	-6.00	0.00	1.00	-5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	-5.00	0.00	1.00	-5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00