

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DESENVOLVIMENTO E IMPLEMENTAÇÃO DE ALGORITMOS
NUMERICAMENTE ESTÁVEIS PARA POSICIONAMENTO DE
PÓLOS POR REALIMENTAÇÃO DE ESTADOS EM SISTEMAS
LINEARES MULTIVARIÁVEIS

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA
PARA OBTENÇÃO DE GRAU DE MESTRE EM ENGENHARIA

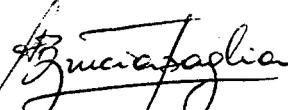
ADHEMAR DE BARROS FONTES

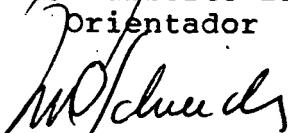
FLORIANÓPOLIS, JULHO - 1988

DESENVOLVIMENTO E IMPLEMENTAÇÃO DE ALGORITMOS NUMERICAMENTE
ESTÁVEIS PARA POSICIONAMENTO DE PÓLOS POR REALIMENTAÇÃO DE
ESTADOS EM SISTEMAS LINEARES MULTIVARIÁVEIS

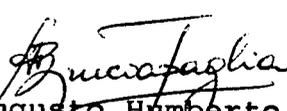
ADHEMAR DE BARROS FONTES

ESTA DISSERTAÇÃO FOI JULGADA PARA A OBTENÇÃO DO TÍTULO DE
MESTRE EM ENGENHARIA - ESPECIALIDADE ENGENHARIA ELÉTRICA E
APROVADA EM SUA FORMA FINAL PELO CURSO DE PÓS-GRADUAÇÃO.

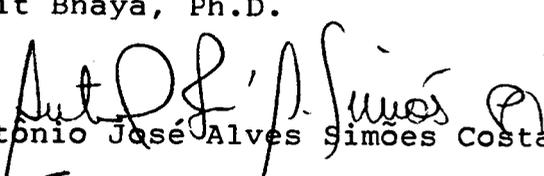

Prof. Augusto Humberto Bruciapaglia, Dr. Ing.
Orientador

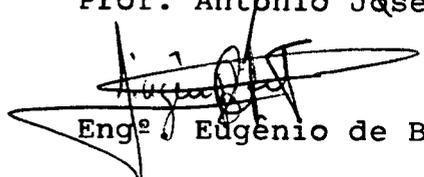

Prof. Márcio Cherem Schneider, Dr.
Coordenador do Curso de Pós-graduação
em Engenharia Elétrica

BANCA EXAMINADORA


Prof. Augusto Humberto Bruciapaglia, Dr. Ing.
Orientador

A. Bhaya -
Prof. Amit Bhaya, Ph.D.


Prof. Antônio José Alves Simões Costa, Ph.D.


Eng. Eugênio de Bona Castelan Neto, M. Eng.

À minha esposa Célia
e meus filhos Abraão,
Adhemar e Ricardo.

À meus pais e
meus irmãos.

AGRADECIMENTOS

Gostaria de manifestar neste pequeno espaço reservado à expressão de sentimentos, os meus sinceros e profundos agradecimentos a todos aqueles que, de uma forma ou de outra, contribuíram para a realização deste trabalho. Em especial ao Prof. Augusto Bruciapaglia, pela orientação, dedicação e amizade demonstrada no decorrer do desenvolvimento deste trabalho, e aos colegas Eugênio de Bona Castelan e Álvaro Renato Rodrigues que, conjuntamente, num trabalho de equipe, colaboraram e tornaram possível a realização deste.

A minha esposa Alexandrina Célia pelo incentivo, dedicação e paciência.

A UFBA, UFSC e a CAPES pelo apoio financeiro que permitiram a realização deste trabalho.

SUMÁRIO

RESUMO.....	ix
ABSTRACT.....	x
CAPÍTULO 1 - INTRODUÇÃO.....	1
CAPÍTULO 2 - PRELIMINARES.....	4
2.1 - Introdução.....	4
2.2 - Formulação do Problema.....	4
2.3 - Formas de Abordagem da Solução.....	6
2.4 - Aspectos Numéricos e Critérios de Avaliação da Solução.....	8
2.4.1 - Aspectos de Análise Numérica.....	8
2.4.2 - Critérios de Avaliação da Solução.....	8
2.5 - Verificação da Controlabilidade.....	12
2.6 - Algoritmos Numericamente Estáveis para Solução do Problema.....	13
2.6.1 - Caso Monovariável.....	16
2.6.2 - Caso Multivariável.....	16
2.7 - Conclusão.....	22
CAPÍTULO 3 - ALGORITMO DE LUENBERGER.....	31
3.1 - Introdução.....	32
3.2 - Redução à FBH: Cálculo dos Índices de Controlabilidade.....	33
3.3 - Redução à Forma Bloco de Frobenius.....	34
3.4 - Redução à Forma Canônica Controlável Multivariável.....	36
3.5 - Cálculo da MRE.....	37

3.6 - Conclusão.....	38
CAPÍTULO 4 - ALGORITMO DE PATEL.....	40
4.1 - Introdução.....	40
4.2 - Redução à FBSIH e Cálculo dos Índices de Controlabilidade.....	40
4.3 - Redução à Forma Intermediária.....	44
4.3.1 - Pré-realimentação de Estado.....	45
4.3.2 - Redução por Rotação de Plano.....	47
4.4 - Redução à FBTI.....	50
4.5 - Cálculo da MRE (K).....	52
4.6 - Cálculo da MRE (K) em Relação à Base Original.....	52
4.7 - Conclusão.....	53
CAPÍTULO 5 - ALGORITMO PROPOSTO.....	55
5.1 - Introdução.....	55
5.2 - Redução à Forma Intermediária (FI).....	56
5.3 - Redução à FCM.....	60
5.4 - Cálculo da MRE.....	62
5.5 - Conclusão.....	64
CAPÍTULO 6 - IMPLEMENTAÇÃO E ANÁLISE DE RESULTADOS.....	66
6.1 - Introdução.....	66
6.2 - Aspectos de Implementação.....	67
6.2.1 - Verificação da Controlabilidade.....	67
6.2.2 - Algoritmo de Luenberger.....	70
6.2.3 - Algoritmo de Patel.....	72
6.2.4 - Algoritmo Proposto.....	75
6.3 - Análise de Resultados.....	78

6.4 - Conclusão.....	84
CAPÍTULO 7 - CONCLUSÕES E PERSPECTIVAS.....	85
REFERÊNCIAS BIBLIOGRÁFICAS.....	88

RESUMO:

A presente dissertação apresenta um estudo dos aspectos teóricos e numéricos do problema de posicionamento de pólos por realimentação de estado em Sistemas Lineares Multivariáveis, invariantes no tempo. Nesta também se desenvolvem e implementam algoritmos numericamente estáveis para solução do problema.

Propõe-se também neste trabalho, um algoritmo baseado na abordagem de desacoplamento com distribuição do esforço de controle entre todas as entradas. Através da análise de resultados obtidos com este e outros algoritmos, comprova-se que este algoritmo é numericamente confiável, apresentando geralmente soluções adequadas para o problema.

Os algoritmos foram implementados em microcomputador compatível IBM-PC, em simples precisão, usando-se linguagem Fortran-77. Adotou-se um conjunto de regras básicas para a codificação estruturada, foi pré-definido um plano de testes para a integração e validação de cada um dos algoritmos e utilizou-se, na maior parte dos casos, procedimentos numericamente confiáveis e computacionalmente eficientes.

ABSTRACT

The present dissertation is a study of the theoretical and numerical aspects of the problem by pole placement by state-feedback in Multivariable Linear Time-invariant Systems. It also deals with the development and implementation of numerically stable algorithms to solve this problem.

In the present work, an algorithm based on the approach of decoupling with control-effort distribution among all the inputs is proposed. Through the analysis of the results obtained with this and other algorithms, it is verified that this algorithm is numerically reliable and generally presents suitable solutions for the problem.

The algorithms were implemented in compatible IBM-PC microcomputers, in single-precision, using Fortran-77 language. For the structured codification a set of basic rules was adopted and a test-schedule was pre-defined for the integration and validation of each algorithm. In most of the cases numerically reliable and computationally efficient procedures were used.

Capítulo 1

INTRODUÇÃO:

Os avanços tecnológicos na área de informática, colocam à disposição do engenheiro de controle, atualmente, ferramentas poderosas que viabilizam a implementação de resultados teóricos já conhecidos e, ainda, induzem novos desenvolvimentos na área de controle e automação de sistemas complexos. Para atender a estas necessidades, desenvolveram-se, paralelamente, pacotes computacionais orientados à análise, projeto e implementação de sistemas de controle (Jamishidi,1985), com o objetivo de auxiliar o projetista na escolha das opções mais adequadas à solução de um dado problema. Nestes pacotes, o esforço computacional associado à implementação de uma metodologia é transparente ao usuário, que exige, em contrapartida, qualidade nos resultados. Assim, os algoritmos colocados à sua disposição devem ser numericamente robustos, computacionalmente eficientes e solucionar problemas com diferentes graus de complexidade.

O problema central em teoria de controle linear, enfoque por variáveis de estados, é o de encontrar uma Matriz de Realimentação de Estados (MRE) tal que os pólos do sistema em malha fechada tenham valores pré-estabelecidos. Este problema caracteriza-se por apresentar várias soluções para um mesmo conjunto de autovalores, devido aos vários graus de liberdade, para o caso multivariável, encontrados na solução do problema. Neste contexto, várias formas de abordagem da solução são possíveis, dentre as quais destacam-se:

- a redução à um único problema monovariável;
- a redução à um determinado número de problemas monovariáveis;
- a solução através da equação de Lyapunov;
- e a solução direta do problema.

Para cada uma destas abordagens, vários algoritmos, na sua maioria matematicamente corretos, têm sido propostos. No entanto, por se tratar basicamente de um problema inverso ao do cálculo de autovalores, onde estão envolvidas todas as questões inerentes à autoestrutura de matrizes (Stewart,1973), os resultados numéricos apresentados pela maioria podem diferir, em maior ou menor grau, da solução correta.

O objetivo deste trabalho é estudar os aspectos teóricos e numéricos do problema de posicionamento de pólos por realimentação de estados em Sistemas Lineares Multivariáveis, invariantes no tempo e desenvolver e implementar algoritmos numericamente confiáveis para solução do problema. Com este objetivo apresenta-se, no próximo capítulo, a formulação do problema, seus aspectos teóricos e numéricos e, devido à sua importância no contexto do problema, apresentam-se, também, os aspectos numéricos e computacionais da verificação da controlabilidade. Ainda neste capítulo, descrevem-se e discutem-se os principais algoritmos numericamente estáveis para a solução dos casos monovariável e multivariável. A seguir, descrevem-se nos capítulos 3, 4 e 5 os algoritmos de "Luenberger", apresentado em (Chen,1984) como Método II, o algoritmo de Patel apresentado em (Patel,1985) e o algoritmo proposto, respectivamente. Estes três algoritmos utilizam como

estratégia de solução, a abordagem do desacoplamento com distribuição do esforço de controle entre todas as entradas. Embora o objetivo deste trabalho seja o desenvolvimento de algoritmos numericamente estáveis para a solução do problema de posicionamento de pólos por realimentação de estados e apesar do algoritmo de Luenberger ser numericamente instável, sua descrição neste trabalho justifica-se, tendo em vista que este é a base conceitual para o desenvolvimento dos algoritmos de Patel e Proposto. No Capítulo 6 apresentam-se os detalhes de implementação computacional destes algoritmos e, através de resultados obtidos, avalia-se o comportamento numérico destes e dos algoritmos de Petkov, apresentado em (Petkov,1986) e de Kautsky, apresentado em (Kautsky e outros,1985), e a robustez numérica das soluções encontradas. Esta avaliação é feita utilizando-se como critérios: os desvios entre os autovalores efetivamente posicionados e os desejados; a sensibilidade dos autovalores da matriz de malha fechada e a norma de Frobenius da MRE encontrada na base original do sistema. Com esta avaliação, comprova-se a eficiência numérica do algoritmo proposto e a sua superioridade em termos de posicionamento de pólos, com relação aos demais algoritmos que utilizam a mesma forma de abordagem da solução.

Capítulo 2

PRELIMINARES

2.1- Introdução:

O problema de posicionamento de pólos de sistemas lineares multivariáveis é formulado neste capítulo, apresentando-se os seus aspectos teóricos e numéricos. A verificação computacional da controlabilidade é abordada, e descreve-se e discute-se os principais algoritmos numericamente estáveis para a solução dos casos monovariável e multivariável.

2.2- Formulação do Problema:

Considere o sistema linear multivariável, invariante no tempo, descrito pelo sistema de equações :

$$p(x(t)) = Ax(t) + Bu(t) \quad (2.1)$$

$$y(t) = Cx(t) \quad (2.2)$$

onde: "p" denota o operador diferencial (d/dt) ou o operador avanço, para sistemas contínuos e discretos no tempo, respectivamente, e $x(t) \in R^n$, $u(t) \in R^p$, $y(t) \in R^q$, $A \in R^{n \times n}$, $B \in R^{n \times p}$ e $C \in R^{q \times n}$.

O posicionamento de pólos por realimentação de estados (PPRE) consiste em encontrar uma matriz de realimentação de estados (MRE), $K \in R^{p \times n}$, da lei de controle:

$$u(t) = Kx(t) + r(t) \quad (2.3)$$

tal que o conjunto de pólos do sistema em malha fechada:

$$p(x(t)) = (A + BK)x(t) + Br(t) \quad (2.4)$$

se encontre em valores pré-definidos no plano complexo, tomando-se os pólos complexos em pares conjugados.

A solução do problema existe, e não é única no caso multivariável, se e somente se o par (A,B) é controlável, ou se os modos não controláveis pertencem ao conjunto dos autovalores desejados (Wonham,1969).

No caso em que o vetor de estado não é totalmente acessível, supondo-se o par (A,C) observável, utiliza-se um observador de estado adequado. Neste caso a solução do problema de PPRE é calculada independentemente do problema de observação (**Princípio da Separação**), podendo-se, com base no Teorema da Dualidade, utilizar a mesma ferramenta para o projeto do observador (Chen,1984).

O PPRE em sistemas lineares multivariáveis caracteriza-se por apresentar várias MRE como solução para um mesmo conjunto de autovalores, devido aos vários graus de liberdade encontrados na solução do problema. Isto se deve às várias combinações possíveis que se pode realizar, entre as entradas acopladas, na ação sobre uma determinada variável de estado. No caso monovariável, o acoplamento inexistente e a solução não apresenta graus de liberdade.

2.3- Formas de Abordagem da Solução:

Dentre as formas possíveis de abordagem da solução do problema multivariável, sob a hipótese de controlabilidade, destacam-se (Chen, 1984):

1) a redução à um único problema monovariável. Por exemplo, através de uma MRE arbitrária K , torna-se a matriz $\hat{A} = A + BK$ cíclica (Chen, 1984) e, pela escolha de um vetor $v \in R^r$, também arbitrário, torna-se o par monovariável (\hat{A}, Bv) controlável por uma única entrada.

2) a redução a um determinado número de problemas monovariáveis, correspondente a um desacoplamento do sistema. Este desacoplamento está relacionado com um determinado critério de escolha de uma matriz transformação, construída a partir de n colunas linearmente independentes da matriz de controlabilidade U a seguir:

$$U = (B : AB : \dots : A^{n-1} B) \quad (2.5)$$

3) a solução através da equação de Lyapunov. Neste caso, escolhe-se uma matriz cíclica $F \in R^{n \times n}$ e uma matriz arbitrária $\bar{K} \in R^{p \times n}$, tais que o par (F, \bar{K}) seja observável, o espectro de F , $\sigma(F)$, seja o conjunto de autovalores desejados e $\sigma(F) \cap \sigma(A) = \emptyset$. Seguramente existe uma matriz $T \in R^{n \times n}$ não singular, tal que:

$$T^{-1} (A + B\bar{K}T^{-1}) T = F \quad (2.6)$$

A partir de (2.6) obtém-se a equação de Lyapunov seguinte, cuja solução é a matriz T desejada:

$$AT - TF = -B\bar{K} \quad (2.7)$$

onde: $\bar{K} = KT$,

e conseqüentemente:

$$K = \bar{K}T^{-1} \quad (2.8)$$

4) a **solução direta do problema**, geralmente reduzindo-se o par (A,B) a uma forma adequada.

Para cada uma destas abordagens vários algoritmos, na sua maioria matematicamente corretos, têm sido propostos. No entanto, a solução para sistemas de dimensão média ou elevada, só é viável através do uso de computadores e, devido à precisão finita destes, os resultados apresentados pela maioria podem diferir, em maior ou menor grau, da solução correta. Por outro lado, a não unicidade da solução conduz à obtenção de diferentes MRE, dentro do conjunto de soluções possíveis, através de diferentes algoritmos confiáveis. Neste contexto, é necessário que certos critérios de avaliação da solução sejam adotados, conduzindo a um conjunto de soluções mais restrito e qualitativamente melhor.

2.4- Aspectos Numéricos e Critérios de Avaliação da Solução:

Neste item são apresentados primeiramente alguns conceitos básicos de análise numérica inerentes ao problema (Stewart,1973; Wilkinson,1965) e, logo após, critérios de avaliação da solução.

2.4.1- Aspectos de Análise Numérica:

- **Condicionamento de um problema:** o condicionamento de um problema mede o comportamento das soluções em face de perturbações nos dados. Assim, um problema é considerado **bem-condicionado**, quando, apesar de perturbações nos dados, devido, por exemplo, às limitações práticas de representação destes, é possível encontrar uma solução próxima da correta; caso contrário, o problema é **mal-condicionado**. O condicionamento é uma característica importante, inerente ao problema, e está relacionada à sua representação matemática (Wilkinson,1965).

Para a avaliação do condicionamento de um problema representado na forma matricial, em relação à sua auto-estrutura, destacam-se as medidas que indicam o grau de sensibilidade global do problema e as que indicam a sensibilidade de cada autovalor independentemente (Wilkinson,1965). Utiliza-se como medida de condicionamento global de uma matriz $A \in R^{n \times n}$ o **número de condicionamento** k_1 , definido a partir da matriz X dos autovetores pela relação:

$$k_1(A) = \|X\|_2 \cdot \|X^{-1}\|_2 \quad (2.9)$$

onde: $\|.\|_2$ representa a norma-2 de uma matriz.

Como medida da sensibilidade de um autovalor λ_i de $A \in \mathbb{R}^{n \times n}$ utiliza-se o número de condicionamento $|s_i^{-1}|$, definido a partir dos autovetores normalizados à direita e à esquerda de A , y_i e x_i , respectivamente, associados com o referido autovalor (Wilkinson, 1965), pela relação:

$$|s_i^{-1}| = |y_i^T x_i|^{-1} \quad (2.10)$$

onde: $\|x_i\|_2 = \|y_i\|_2 = 1$

- **Estabilidade numérica de um algoritmo:** a estabilidade numérica de um algoritmo mede o desvio da solução encontrada pelo algoritmo adotado em relação à solução exata. Um algoritmo é **numericamente estável** quando produz uma solução próxima à exata, para o problema com os dados ligeiramente perturbados; caso contrário, o algoritmo é **numericamente instável** (Stewart, 1973).

A estabilidade de um algoritmo pode ser avaliada utilizando-se o método de análise de erro (Wilkinson, 1965; Stewart, 1973), através do qual se contabiliza o erro total de arredondamento acumulado nas operações aritméticas executadas no computador.

Os dois conceitos apresentados, o de condicionamento de um problema e o de estabilidade numérica, são independentes e as seguintes considerações são importantes (Chen, 1984):

- a utilização de algoritmos numericamente estáveis para resolver problemas bem-condicionados geralmente apresentará bons resultados, o que não é garantido no caso de problemas mal-condicionados;

- quando o método é numericamente instável, o resultado deve ser cuidadosamente examinado, independentemente do condicionamento do problema;

- no projeto ou na implementação dos algoritmos, caso haja a necessidade de se utilizar procedimentos numericamente não-confiáveis ou instáveis, deve-se fazê-lo tão ao final do algoritmo quanto possível.

- **Transformações ortogonais:** com o objetivo de simplificar o problema, a maioria dos algoritmos propostos reduzem a representação original do sistema a formas mais adequadas, através de transformações de similaridade sobre o par (A,B) . Tais reduções são, em geral, fontes de instabilidade numérica. No entanto, uma classe de transformações, chamadas ortogonais, são numericamente estáveis. Estas são realizadas utilizando-se matrizes ortogonais que apresentam as seguintes características (Stewart, 1973):

- para uma matriz ortogonal Q , tem-se: $Q^{-1} = Q^T$;

- não modificam o condicionamento do problema;

- facilitam a análise de erro, pois as normas espectral e euclidiana são invariantes sob estas transformações, possibilitando a fácil verificação da estabilidade numérica de um determinado algoritmo.

Geometricamente, estas características podem ser facilmente visualizadas pois através deste tipo de transformações mantém-se a representação do sistema ainda numa base ortogonal. As transformações ortogonais usadas neste trabalho, são (Wilkinson, 1965; Stewart, 1973):

a) **Rotação de Plano:** dado um vetor $v \in R^n$, a rotação de plano que anula o elemento v_j com o elemento v_i é realizada através de uma matriz ortogonal $G \in R^{n \times n}$, tal que:

$$G_{i,j} (v_1 \dots v_i \dots v_j \dots v_n)^T = (v_1 \dots \bar{v}_i \dots 0 \dots v_n)^T \quad (2.11)$$

onde:

$$G_{i,j} = \begin{array}{cccccccc} & & & & \begin{array}{c} i \\ \downarrow \end{array} & \begin{array}{c} j \\ \downarrow \end{array} & & \\ & & & & 0 & \dots & 0 & \dots & 0 \\ & & & & \vdots & & \vdots & & \vdots \\ & & & & \vdots & & \vdots & & \vdots \\ & & & & 0 & \dots & 1 & \dots & 0 \\ & & & & \vdots & & \vdots & & \vdots \\ & & & & \vdots & & \vdots & & \vdots \\ & & & & 0 & \dots & 0 & \dots & \alpha & \dots & \beta & \dots & 0 & \leftarrow i \\ & & & & \vdots & \\ & & & & 0 & \dots & 0 & \dots & -\beta & \dots & \alpha & \dots & 0 & \leftarrow j \\ & & & & \vdots & \\ & & & & 0 & \dots & 1 \end{array}$$

$$\alpha = v_i / (v_i^2 + v_j^2)^{1/2} \quad (2.12)$$

$$\beta = v_j / (v_i^2 + v_j^2)^{1/2} \quad (2.13)$$

b) **Transformação de Householder:** é realizada através de uma matriz ortogonal e simétrica $H \in R^{n \times n}$, tal que:

$$Hv = \tau \|v\|_2 \cdot e_i \quad (2.14)$$

onde: $e_i = (1 \ 0 \ \dots \ 0)^T$;

$$H = I - (uu^T) / \rho \quad (2.15)$$

$$u = e_i + v / \tau \|v\|_2 \quad (2.16)$$

$$\rho = 1 + v_i / \tau \|v\|_2 \quad (2.17)$$

$$\tau = \text{sgn}(v_i) \quad (2.18)$$

A transformação de Householder, comparativamente à rotação de plano, é computacionalmente mais eficiente.

2.4.2- Critérios de Avaliação da Solução:

Adota-se, neste trabalho, como critérios de avaliação da solução:

a) **Estabilidade numérica do algoritmo:** tem por objetivo avaliar a solução apresentada pelo algoritmo com relação à uma solução exata. Os algoritmos que utilizam procedimentos numericamente estáveis são, em geral, numericamente estáveis. A utilização de um procedimento numericamente instável, torna o algoritmo também numericamente instável. Este critério conduz à uma validação da solução encontrada baseada na proximidade dos autovalores da matriz de malha fechada com os desejados.

b) **Robustez da solução:** avalia a sensibilidade dos autovalores da matriz de malha fechada, obtida a partir de uma MRE encontrada, com respeito a perturbações nos coeficientes das

matrizes do sistema e/ou da MRE, garantindo a eficiência da Lei de Controle no que se refere ao posicionamento de pólos. Neste sentido, pequenas perturbações nos parâmetros do sistema não deverão provocar grandes desvios nos autovalores alocados. Um aspecto importante deste critério é que a minimização da sensibilidade dos autovalores da matriz de malha fechada garante também a minimização dos limites superior e inferior da resposta transitória do sistema (Kautsky,1985).

c) **Norma da MRE:** com este critério seleciona-se uma MRE de menor norma, tal que o esforço dispendido na ação de controle seja minimizado. Deve-se ressaltar a vinculação deste critério com o anterior, pois o atendimento do primeiro limita a norma espectral da MRE (Kautsky,1985).

A adoção desses critérios conduz à redução do número de algoritmos recomendáveis de serem utilizados na solução do problema via computador.

2.5- Verificação da Controlabilidade:

Outro aspecto importante do ponto de vista teórico e numérico, na solução do problema, é o da verificação da controlabilidade. Dentre os métodos utilizados para verificação computacional destacam-se (Chen,1984):

1) **cálculo do posto da matriz de controlabilidade (2.5):** se $\text{posto}(U) = n$, então o sistema é controlável.

2) cálculo dos postos das matrizes $Z_i = (B : A - \lambda_i I)$ para $i = 1, 2, \dots, n$: se $\text{posto}(Z_i) = n, \forall i$, então o sistema é controlável.

3) comparação entre os autovalores das matrizes A e $(A + BF)$, onde $F \in \mathbb{R}^{p \times n}$ é uma matriz arbitrária: o sistema é controlável se e somente se existir no mínimo uma matriz F tal que os autovalores de A sejam distintos dos de $(A + BF)$.

4) redução do par (A, B) à forma bloco de Hessenberg (FBH) através de transformações ortogonais.

Todos os procedimentos desenvolvidos a partir dos três primeiros métodos, em vários casos, apresentam resultados não confiáveis. Isto se justifica pelos seguintes motivos:

- embora o posto possa ser calculado através da "Decomposição em Valores Singulares" (Stewart, 1973), que é um método numericamente estável, o condicionamento da matriz U deteriora-se à medida que a dimensão n do problema aumenta. Neste caso, os ângulos entre as últimas colunas de U tendem a diminuir à medida que n aumenta, dificultando a análise computacional da dependência linear.

- apesar do método QR para cálculo dos autovalores ser numericamente estável, os resultados não são confiáveis em casos mal condicionados (Wilkinson, 1965).

O método baseado na redução do par (A,B) para a FBH (A_H, B_H) , também denominada **forma canônica ortogonal**, foi sugerido por vários pesquisadores, entre eles Van Dooren (1981), Paige (1981), Eising (1981) e Patel (1982). Embora outras transformações ortogonais sejam possíveis, como garantia de bom comportamento e eficiência computacional, utiliza-se a transformação ortogonal de Householder. A FBH é da seguinte forma:

$$A_H = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1,k-1} & A_{1,k} \\ A_{21} & A_{22} & \dots & A_{2,k-1} & A_{2,k} \\ 0 & A_{32} & \dots & A_{3,k-1} & A_{3,k} \\ \vdots & \vdots & & \vdots & \vdots \\ \vdots & \vdots & & \vdots & \vdots \\ 0 \dots 0 & \dots & A_{k,k-1} & A_{k,k} \end{pmatrix} ; B_H = \begin{pmatrix} A_{10} \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \quad (2.19)$$

onde: $A_{i,j} \in R^{n_i \times n_j}$;

posto $(A_{i,i-1}) = n_i$, $i = 1 \dots k$ e

k é o menor inteiro tal que $n_k = 0$ ou $\sum_{i=1}^k n_i = n$.

A controlabilidade se verifica quando $\sum_{i=1}^k n_i = n$,

o que é equivalente ao posto da matriz de controlabilidade ser máximo. Os inteiros n_i e k são denominados, respectivamente, **índices de Kronecker** e **índice de controlabilidade do sistema**.

Este procedimento também permite separar os modos controláveis do sistema, efetuando uma decomposição equivalente à decomposição de "Kalman" (Chen, 1984) e, além disso, é utilizado em vários algoritmos como passo inicial das transformações necessárias.

2.6- Algoritmos Numericamente Estáveis para Solução do Problema:

São apresentados, a seguir, algoritmos para solução do caso monovariável e, posteriormente, os utilizados no caso multivariável. Os primeiros podem ser usados indistintamente quando o problema multivariável é reduzido a um ou mais casos monovariáveis.

2.6.1- Caso Monovariável:

Considere o sistema linear monovariável, invariante no tempo, controlável, descrito pela seguinte equação:

$$p(x(t)) = Ax(t) + bu(t) \quad (2.20)$$

onde: $x \in R^n$, $u(t) \in R$, $A \in R^{n \times n}$ e $b \in R^n$.

É necessário achar um vetor linha $k \in R^n$, tal que os autovalores da matriz de malha fechada $A_c = A + bk$ sejam os desejados. Como já mencionado anteriormente, a solução existe e é única.

Dentre os algoritmos propostos que utilizam somente transformações numericamente estáveis, para solução deste problema, destacam-se os apresentados por Miminis (Miminis,1981), Patel (Patel,1984) e Petkov (Petkov e outros,1986). Estes algoritmos têm em comum os seguintes aspectos:

1) reduzem o par (A,b) para a Forma Superior Irreduzível de Hessenberg (FSIH), (\tilde{A},\tilde{b}) , através de transformações ortogonais de Householder. A FSIH é como segue:

$$\tilde{A}=PAP^{-1} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1,n-1} & a_{1,n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2,n-1} & a_{2,n} \\ 0 & a_{32} & a_{33} & \dots & a_{3,n-1} & a_{3,n} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & a_{n,n-1} & a_{n,n} \end{pmatrix} ; \tilde{b}=Pb = \begin{pmatrix} a_{1e} \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (2.21)$$

onde : $a_{i,i-1} \neq 0$, para $i = 1,2,\dots,n$, uma vez que o par (A,b) é controlável;

2) reduzem a matriz de malha fechada $\tilde{A}_c = \tilde{A} + \tilde{b}\tilde{k}$ à Forma Real de Schur (FRS) (Miminis,1982) a seguir, onde os blocos 2x2 e os escalares da diagonal contém os autovalores desejados:

$$A_c = \begin{pmatrix} \lambda_1 & x & \dots & \dots & \dots & x & x \\ \cdot & \cdot & & & & \cdot & \cdot \\ \cdot & & \cdot & & & \cdot & \cdot \\ \cdot & & & \cdot & & \cdot & \cdot \\ 0 & & & \lambda_r & x & x & x \\ 0 & \dots & \dots & 0 & \alpha_1 & \beta_1 & x \\ 0 & \dots & \dots & 0 & \gamma_1 & \delta_1 & x \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \dots & \dots & 0 & 0 & 0 & \dots \\ 0 & \dots & \dots & 0 & 0 & 0 & \dots \end{pmatrix} \quad (2.22)$$

3) os dois primeiros algoritmos citados são baseados no método QR para cálculo de autovalores.

a) Algoritmo de Petkov:

A idéia central do algoritmo proposto por Petkov (Petkov e outros,1986) é a seguinte:

- desde que as matrizes \tilde{A} e $\tilde{A}_c = \tilde{A} + \tilde{b}\tilde{k}$ estão na forma de Hessenberg e diferem somente na primeira linha, é possível encontrar um autovetor da matriz \tilde{A}_c , associado a um autovalor desejado, antes de calcular \tilde{k} .

Neste algoritmo, na alocação de um autovalor λ ; real, é efetuado um conjunto de rotações de plano com o objetivo de anular todos os elementos do autovetor associado, exceto, evidentemente, o primeiro. Estas transformações são aplicadas nas matrizes \tilde{A} e \tilde{A}_c e, pela condição de alocação de autovalor adotada, obtém-se uma equação para um elemento da matriz \tilde{k} . No caso de um autovalor complexo, um procedimento similar baseado em aritmética real é utilizado e dois elementos de \tilde{k} são determinados. Como \tilde{A} e \tilde{A}_c permanecem na forma de Hessenberg, é possível repetir o procedimento até a alocação de todos os autovalores. Em cada etapa o algoritmo trabalha com um subsistema de dimensão uma ou duas unidades menor e as rotações de plano são determinadas para um autovetor deste. No final do processo obtém-se uma matriz ortogonal Q , e um vetor de realimentação de estados \tilde{k} , tal que:

$$Q^T(\tilde{A} + \tilde{b}\tilde{k})Q = S \quad (2.23)$$

onde: Q é a matriz que representa todas as rotações efetuadas;

S é a matriz final na FRS.

O vetor de realimentação de estado em relação à base original é calculado por:

$$k = \tilde{k}PQ \quad (2.24)$$

onde: P é a matriz que reduz o par (A,b) para a FSIH.

b) Algoritmo de Patel:

O algoritmo proposto por Patel (Patel,1984;1985) baseia-se no método QR com deslocamento implícito, permitindo, como no caso anterior, a utilização de aritmética real na alocação de autovalores complexos. O deslocamento é efetuado sobre a matriz de malha aberta \tilde{A} , onde se calcula uma matriz ortogonal P; a cada alocação de um autovalor, que anula o elemento $a_{n,n-1}$ da n-ésima linha da matriz $(\tilde{A} - \lambda;I)$, no caso de λ ; real, e os elementos $a_{n,n-2}$ e $a_{n,n-1}$, também da n-ésima linha do produto das matrizes $(\tilde{A} - \lambda;I)$ e $(\tilde{A} - \lambda^*I)$, quando λ ; e λ^* são complexos conjugados. Na alocação de um autovalor real ou par de autovalores complexos conjugados, semelhantemente ao algoritmo anterior, determina-se um ou dois elementos do vetor de realimentação de estados \tilde{k} , respectivamente. No entanto, devido às transformações efetuadas à cada alocação, é necessário que se utilize um processo de atualização do vetor \tilde{k} , referenciando-o sempre ao sistema de coordenadas definidas para o par (\tilde{A},\tilde{b}) . Quando o último autovalor alocado é real ou quando os últimos alocados são um par de complexos conjugados, onde não são efetuadas transformações, é necessário também, contrariamente ao proposto por Patel em (Patel,1984), que se referencie o elemento, ou os elementos, de \tilde{k} encontrado(s), ao sistema de coordenadas do par (\tilde{A},\tilde{b}) . No final do processo de alocação, obtém-se equações similares a (2.23) e (2.24).

c) Algoritmo de Miminis:

O algoritmo proposto por Miminis (Miminis,1981), baseia-se também no método QR com deslocamento implícito, que é, no entanto, efetuado sobre a matriz de malha fechada \tilde{A}_c , que se encontra na forma de Hessenberg e tem sua primeira linha desconhecida. Este algoritmo é executado em duas etapas:

1) **varredura direta:** reduz-se a matriz \tilde{A}_c à FRS, com os autovalores desejados alocados em blocos 2x2 ou escalares da diagonal e com os elementos acima desta desconhecidos. Semelhantemente ao algoritmo de Patel, na alocação de um autovalor real λ_i ou par de autovalores complexos conjugados λ_i e λ_i^* , determina-se uma matriz ortogonal P_i que anule o elemento $a_{n,n-1}$ da n-ésima linha da matriz $(\tilde{A} - \lambda_i I)$ ou os elementos $a_{n,n-2}$ e $a_{n,n-1}$, também da n-ésima linha da matriz $(\tilde{A} - \lambda_i I) \times (\tilde{A} - \lambda_i^* I)$, respectivamente. A cada alocação, uma transformação representada por P_i é aplicada em \tilde{A}_c .

2) **varredura inversa:** onde determina-se a linha desconhecida de \tilde{A}_c . Isto é realizado, recursivamente, a partir do (n,n)-ésimo elemento da matriz na FRS, no caso do último autovalor alocado ter sido real, ou a partir da primeira linha do último bloco da referida matriz, no caso dos últimos autovalores alocados terem sido complexos. Neste processo, utiliza-se os elementos correspondentes às transformações efetuadas.

A seguir, calcula-se \tilde{k} , como segue:

$$\tilde{k}_i = (a_{c,i} - a_{i,i}) / a_{i,i}, \quad \text{com } i = 1, 2, \dots, k \quad (2.25)$$

onde: $a_{c,i}$ representa o i -ésimo elemento da primeira linha de \tilde{A}_c ;

$a_{i,i}$ representa o i -ésimo elemento da primeira linha de \tilde{A} ;

\tilde{k}_i representa o i -ésimo elemento do vetor \tilde{k} .

2.6.1.1- Considerações Sobre os Algoritmos:

Algumas considerações sobre os algoritmos para solução do caso monovariável, apresentados anteriormente, devem ser destacadas:

1) No algoritmo de Petkov os elementos dos autovetores são calculados de forma recursiva: para sistemas de dimensão elevada e mal-condicionados, este processo traz problemas numéricos.

2) No algoritmo de Miminis são efetuadas divisões por β (onde β representa um elemento da matriz de rotação de plano) na varredura inversa: em problemas mal-condicionados, este elemento pode se tornar bastante pequeno, provocando instabilidade numérica. Em (Miminis, 1984) é apresentado um procedimento para o cálculo de k , onde se elimina o problema anterior, sendo que, no entanto, há necessidade de armazenamento de todas as transformações executadas durante o processo de alocação. Este fato torna este procedimento desvantajoso do ponto de vista de ocupação de memória.

3) O algoritmo de Patel apresenta propriedades numéricas satisfatórias, além de ser conceitualmente simples e fácil de ser implementado.

2.6.2- Caso Multivariável:

A seguir são discutidos os principais algoritmos numericamente estáveis para solução do caso multivariável.

a) Algoritmo de Patel:

Dois algoritmos propostos por Patel (Patel e Misra, 1984; Patel, 1985) ambos com características numéricas desejáveis, utilizam a abordagem de desacoplamento do sistema original como meio de solucionar o problema. Como citado em 2.3, a estratégia de desacoplamento está relacionada com a forma de escolha de n colunas linearmente independentes da matriz U . Assim, entre os algoritmos propostos por Patel existe uma diferença fundamental no desacoplamento: no primeiro (Patel, 1984), este é obtido concentrando o esforço de controle entre as primeiras entradas, enquanto no segundo (Patel, 1985), as ações de controle são distribuídas entre todas as entradas. Após o desacoplamento, a MRE pode ser obtida utilizando-se qualquer um dos algoritmos apresentados anteriormente para o caso monovariável.

a.1) no primeiro algoritmo (Patel, 1984), obtém-se o desacoplamento reduzindo-se a representação do sistema original

à uma forma bloco triangular superior (FBTS), através de uma transformação ortogonal, representada aqui por T, tal que:

$$\bar{A} = TAT' = \begin{vmatrix} \bar{A}_{11} & \bar{A}_{12} & \dots & \bar{A}_{1,p} \\ 0 & \bar{A}_{22} & \dots & \bar{A}_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{A}_{p,p} \end{vmatrix}; \quad \bar{B} = TB = \begin{vmatrix} \bar{b}_{11} & \bar{b}_{12} & \dots & \bar{b}_{1,p} \\ 0 & \bar{b}_{22} & \dots & \bar{b}_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \bar{b}_{p,p} \end{vmatrix} \quad (2.26)$$

onde: $A_{i,i}$, $i = 1, 2, \dots$, são matrizes $\bar{\mu}_i \times \bar{\mu}_i$, na Forma Superior Irredutível de Hessenberg (FSIH);

$b_{i,i}$, com $i = 1, 2, \dots$, são vetores $\bar{\mu}_i \times 1$, que têm a seguinte estrutura: $b_{i,i} = (x \ 0 \ \dots \ 0)^T$;

p representa o menor número de entradas, a partir da primeira, com os quais é possível controlar o sistema;

$\bar{\mu}_i$ representa o número de variáveis de estado controladas pela entrada μ_i , nesta estratégia de desacoplamento.

a.2) no segundo algoritmo (Patel, 1985), obtém-se o desacoplamento reduzindo-se o par (A,B) em (2.1) para (\bar{A}, \bar{B}) numa Forma Bloco Triangular Inferior (FBTI), utilizando-se os seguintes procedimentos numericamente estáveis: rotação de plano, pré-realimentação de estados, permutação de linhas e colunas e transformações ortogonais de Householder. Este algoritmo será apresentado detalhadamente no capítulo 4.

b) Algoritmo de Cavin-Bhattacharyya:

O algoritmo proposto por Cavin e Bhattacharyya (Cavin e Bhattacharyya, 1982) utiliza a terceira abordagem de solução apresentada no item 2.3, usando os graus de liberdade para tornar a matriz T , solução da equação de Lyapunov (2.7), a melhor condicionada quanto possível. Com isto, a sensibilidade às perturbações da matriz de malha fechada é minimizada.

c) Algoritmo de Kautsky:

O algoritmo proposto por Kautsky (Kautsky e outros, 1982) também utiliza os graus de liberdade disponíveis na solução do problema com o mesmo objetivo do anterior. Neste caso o problema é solucionado diretamente e sua idéia central é:

- seja Λ uma matriz bloco diagonal, cujos blocos contêm os autovalores desejados. Se não existe um autovalor de multiplicidade maior que o número de entradas efetivas, então existe uma matriz X , não singular (Davison, 1970), tal que:

$$(A + BK) = X \Lambda X^{-1} \quad (2.27)$$

onde: X é a matriz formada pelos autovetores x_i ,
 $i = 1, 2, \dots, n$, da matriz de malha fechada.

Sabe-se, no entanto, que no caso multivariável, devido à não unicidade da solução, existem várias matrizes de autovetores

para um mesmo conjunto de autovalores. Utilizando-se estes graus de liberdade, os autovetores x_i , $i = 1, 2, \dots, n$, são escolhidos, dentre os possíveis, de tal forma que os ângulos entre eles sejam os mais próximos de 90° quanto possível. Assim, os parâmetros $|s_i^{-1}|$ apresentados em (2.10), que indicam a sensibilidade dos autovalores da matriz de malha fechada, são minimizados. A matriz K é obtida como solução da equação:

$$BK = X \wedge X^{-1} - A \quad (2.28)$$

Utilizando-se a decomposição QR em B , obtém-se:

$$B = (U_0 \ U_1) \begin{vmatrix} Z \\ 0 \end{vmatrix} \quad (2.29)$$

onde: $U = (U_0 \ U_1)$ é uma matriz ortogonal;

Z é uma matriz não singular, triangular inferior.

Assim, a solução de (2.28) é dada por:

$$K = Z^{-1} U_0^T (X \wedge X^{-1} - A) \quad (2.30)$$

Este algoritmo consiste, então, das três etapas seguintes:

1) **etapa A:** efetua-se a decomposição QR de B , obtendo-se U_0 , U_1 e Z e constroem-se bases ortogonais para todos os possíveis autovetores à direita, correspondentes aos autovalores desejados;

2) etapa X: seleciona-se, utilizando-se um dos critérios de otimalidade dados em (Kautsky e outros, 1982), um conjunto de autovetores a direita x_i , com $\|x_i\|_2 = 1$ e $i = 1, 2, \dots, n$, destas bases, tal que $X = (x_1 \ x_2 \ \dots \ x_n)$ é a melhor condicionada quanto possível;

3) etapa K: encontra-se a matriz $M = A + BK$, resolvendo a equação $MX = X\Lambda$ e então calcula-se K através de (2.30).

Neste algoritmo, além da redução na sensibilidade da matriz de malha fechada com a escolha adequada dos autovetores, atende-se também a outros critérios, tais como:

- redução do limite superior da norma da matriz de ganho K, podendo-se esperar também uma redução do esforço dispendido na ação de controle ;
- redução dos limites superior e inferior da resposta transitória do sistema em malha fechada;
- aumento da matriz perturbação Δ , tal que a matriz de malha fechada perturbada $(A + BK + \Delta)$ permanece estável.

d) Algoritmo de Petkov:

No algoritmo apresentado por Petkov (Petkov e outros, 1984), o par (A, B) é reduzido, inicialmente, para (A_M, B_M) na forma superior de Hessenberg (2.19), onde as submatrizes $A_i, i-1$, $i = 2, 3, \dots, k$ têm posto n_i , através de transformações

ortogonais. Como no caso monovariável, a idéia básica é a seguinte:

- desde que a matriz de malha fechada $(A_n - B_n \tilde{K})$ também se encontra na FBH, é possível achar um autovetor x_i desta matriz conhecendo somente a matriz A_n e o autovalor desejado λ_i .

Na alocação de um autovalor, os elementos do autovetor são sucessivamente anulados por meio de rotações de plano para se obter uma equação para uma coluna de K . A sequência de rotações de plano é executada de forma que a estrutura de A_n é preservada, garantindo-se assim, a possibilidade de prosseguir da mesma forma na alocação do próximo autovalor, com um subsistema de ordem menor. No final do algoritmo, obtém-se uma matriz de realimentação de estados \tilde{K} , tal que:

$$Q^T(A_n + B_n \tilde{K})Q = S \quad (2.31)$$

onde: Q é matriz que representa todas as rotações efetuadas;

S é a matriz final, na FRS.

e) Algoritmo de Miminis:

O algoritmo proposto por Miminis (Miminis, 1984) baseia-se também no método QR para cálculo de autovalores, e o deslocamento, como no caso monovariável, também é efetuado sobre a matriz de malha fechada. Inicialmente o algoritmo reduz o par

(A, B) para (A_{n_i}, B_{n_i}) na forma bloco superior irredutível de Hessenberg, onde as submatrizes $A_{n_i, i-1}$, com $i = 1, 2, \dots, k$ estão na forma $(0 \ R_i)$, sendo R_i uma matriz não-singular triangular inferior. Utiliza-se no processo de alocação dos autovalores dois procedimentos:

- o primeiro acontece na alocação dos primeiros $(n_1 - n_2)$ autovalores, quando $n_1 > n_2$, e na alocação de um autovalor λ_i , quando o número de elementos da menor subdiagonal inferior não nula da matriz $\tilde{A}_c^{(i)}$, que representa a submatriz de \tilde{A}_c atualizada após alocação de $(i - 1)$ autovalores, for igual ao número de linhas desconhecidas da referida submatriz. Neste caso, a alocação é direta, sem necessidade de se efetuar transformações, utilizando-se os graus de liberdade de tal forma que o autovetor associado ao autovalor alocado seja ortogonal ao espaço gerado pelos demais autovetores.

- o segundo, que acontece nos demais casos, é baseado no método QR, onde se determina uma matriz Q_i que representa todas as rotações de plano necessárias para alocação do autovalor λ_i . A MRE \tilde{K} , referida à base do par (A_{n_i}, B_{n_i}) , é determinada como segue:

$$\tilde{K} = \bar{K}^{(n)} Q_n^T \dots Q_1^T \quad (2.32)$$

onde: $\bar{K}^{(n)}$ corresponde a solução do caso monovariável, obtido após alocação de $p-1$ autovalores, assumindo que $\text{posto}(B) = p$;

$$Q_i = \begin{vmatrix} I_{i-1} & 0 \\ 0 & \tilde{Q}_i \end{vmatrix}, \quad i = 1, 2, \dots, n, \quad \text{sendo } \tilde{Q}_i \text{ a matriz que}$$

representa todas as rotações executadas na alocação de λ_i .

2.6.2.1- Considerações Sobre os Algoritmos:

Algumas considerações sobre os algoritmos apresentados neste item merecem destaque:

1) no algoritmo de Petkov os graus de liberdade, representados pela possibilidade de arbítrio de elementos dos autovetores, são utilizados arbitrariamente sem compromisso com a robustez da solução;

2) o algoritmo de Miminis, apesar de apresentar uma solução robusta, é conceitualmente complexo e apresenta dificuldades na implementação;

3) o algoritmo de Kautsky depende do condicionamento da matriz X , composta pelos autovetores da matriz de malha fechada. Apesar do algoritmo utilizar os graus de liberdade para minimizar o condicionamento da referida matriz, esta pode ainda ser mal-condicionada, tendo em vista a natureza do problema;

4) o algoritmo de Cavin-Bhattacharyya, como no caso anterior, depende do condicionamento da matriz T . Neste algoritmo, além das considerações citadas anteriormente, deve-se ressaltar que a

equação (2.7) é mal-condicionada se os autovalores de F são próximos aos da matriz A (Golub, 1979). Ainda neste algoritmo, um outro problema é a necessidade de se ter algum conhecimento sobre os autovalores de A , de forma a garantir que $\sigma(A) \cap \sigma(F) = \emptyset$;

5) o primeiro algoritmo apresentado por Patel concentra o esforço de controle entre as primeiras entradas. Com isto, apesar do algoritmo ser numericamente estável, a solução encontrada não atende aos critérios 2 e 3 apresentados no item (2.4.2);

6) o segundo algoritmo apresentado por Patel, ao contrário do anterior, distribue o esforço de controle entre todas as entradas, de forma que a norma da MRE encontrada é reduzida. Consequentemente reduz-se também o esforço dispendido na ação de controle. Apesar da solução encontrada não ser a ótima, dentro dos critérios citados no sub-item (2.4.2), esta não depende do condicionamento do sistema realimentado, além do que este algoritmo é conceitualmente simples e de fácil implementação. Deve-se ressaltar, no entanto, que a pré-realimentação de estado, utilizada neste algoritmo, modifica o condicionamento do problema, o que pode comprometer, para sistemas mal-condicionados, a estabilidade numérica do algoritmo.

2.7- Conclusão:

Apresentou-se neste capítulo a formulação do problema de posicionamento de pólos por realimentação de estados de sistemas lineares multivariáveis invariantes no tempo, seus aspectos teóricos e numéricos e uma breve discussão sobre os principais algoritmos numericamente estáveis existentes para a solução do caso mono e multivariável. Também, devido à sua importância no contexto do problema, abordou-se a questão da verificação computacional da controlabilidade.

Efetuando-se uma análise do que foi apresentado e discutido neste capítulo, e considerando-se a natureza mal-condicionada do problema e a extrema sensibilidade dos autovalores da matriz de malha fechada à perturbações nos parâmetros da MRE, conclui-se que o problema de PPRE ainda é uma questão não totalmente solucionada, porque não existe um algoritmo numericamente estável, que permita resolver o problema integralmente com todas as características citadas.

Capítulo 3

ALGORITMO DE LUENBERGER

3.1- Introdução:

Descreve-se neste capítulo o algoritmo de Luenberger (Anderson e Luenberger, 1967), apresentado como Método II em (Chen, 1984). Este consiste em reduzir o sistema original para uma forma canônica controlável multivariável (FCCM) e, então computar a MRE para o sistema efetivamente desacoplado. A FCCM é encontrada baseando-se nos índices de Controlabilidade (IC) e no reagrupamento das variáveis de estado. Embora para sua obtenção seja obrigatório utilizar procedimentos numericamente instáveis, sua descrição neste trabalho deve-se ao fato de conter a base conceitual para o desenvolvimento dos algoritmos descritos nos dois próximos capítulos.

Para obtenção da FCCM, utiliza-se um algoritmo que reduz inicialmente a representação do sistema a FBH, através de transformações ortogonais de Householder e determina-se, também, os IC. A seguir, efetua-se uma redução intermediária para a Forma Bloco de Frobenius (FBF), que corresponde ao passo numericamente instável do algoritmo, e finalmente, através de permutações de linhas e de colunas, obtém-se a forma desejada (Chen, 1984).

3.2- Redução a FBH: cálculo dos Índices de Controlabilidade:

A forma bloco de Hessenberg (FBH), mencionada em (2.19) é obtida, recursivamente, através de condensação por linhas das submatrizes de A, determinadas pelo particionamento desta com base no posto da submatriz anterior já condensada, e utilizando-se transformações ortogonais de Householder. Na condensação, a forma final das submatrizes $A_{i,i-1}$, $i = 2, 3, \dots, k$ é a "escalonada superior" (Strang, 1980), necessária para a obtenção da FCCM desejada. Esta forma facilita, também, o cálculo dos índices de Kronecker e dos índices de Controlabilidade.

Os índices de Controlabilidade (IC), que indicam o número de variáveis de estado controladas por cada entrada, são calculados analisando a dependência linear entre as colunas das submatrizes $A_{i,i-1}$ em (2.19), o que é equivalente à análise da dependência linear entre as colunas da matriz de controlabilidade U e utilizando-se os índices de Kronecker.

Com o objetivo de ilustrar e facilitar o entendimento deste algoritmo, e também dos apresentados nos capítulos seguintes, tome-se como exemplo o par (A,B), controlável, na FBH e B de posto completo, representado pelas seguintes matrizes:

$$A = \begin{array}{c|ccc|ccc|cc} \hline x & x & x & x & x & x & x & x \\ \hline x & x & x & x & x & x & x & x \\ \hline x & x & x & x & x & x & x & x \\ \hline \underline{x} & x & x & x & x & x & x & x \\ \hline o & \underline{x} & x & x & x & x & x & x \\ \hline o & o & \underline{x} & x & x & x & x & x \\ \hline o & o & o & o & \underline{x} & x & x & x \\ \hline o & o & o & o & o & \underline{x} & x & x \\ \hline \end{array} ; \quad B = \begin{array}{c|ccc} \hline \underline{x} & x & x \\ \hline o & \underline{x} & x \\ \hline o & o & \underline{x} \\ \hline o & o & o \\ \hline \end{array} \quad (3.1)$$

onde : $n_1 = 3, n_2 = 3, n_3 = 2$ e $k = 3$.

A tabela 3.1 abaixo ilustra, para o exemplo, o mecanismo de obtenção dos IC, onde os campos preenchidos com "x" representam as colunas linearmente independentes na matriz de controlabilidade, identificadas pelo processo de análise das submatrizes $A_{i,i}$ mencionado anteriormente.

	b_1	b_2	b_3	
I	x	x	x	$n_1=3$
A	x	x	x	$n_2=3$
A^2	o	x	x	$n_3=2$
	$\mu_1=2$	$\mu_2=3$	$\mu_3=3$	

Tabela 3.1. Cálculo dos Índices de Controlabilidade

Na redução do par (A,B) para a FBH, (A_μ, B_μ) , obtém-se uma matriz ortogonal T , tal que:

$$A_\mu = TAT^T \quad e \quad B_\mu = TB \quad (3.2)$$

3.3- Redução a Forma Bloco de Frobenius:

A redução à FBF (A_f, B_f) , como proposta em (Chen, 1984), é realizada nos seguintes passos:

1) transforma-se os elementos x_i do par (A,B) , $i = 1, 2, \dots, n_1$ em B e $i = n_1+1, n_1+2, \dots, n$ em A , para elementos unitários, através de operações elementares que podem ser representados pela matriz:

Realizando-se as operações descritas anteriormente, obtém-se:

$$A_f = \begin{array}{|ccc|ccc|cc|} \hline x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hline \end{array} ; B_f = \begin{array}{|ccc|} \hline 1 & x & x \\ 0 & 1 & x \\ 0 & 0 & 1 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 0 & 0 \\ \hline \end{array} \quad (3.5)$$

3.4- Redução à Forma Canônica Controlável Multivariável:

A FCCM é obtida a partir da matriz reduzida à FBF, usando uma transformação ortogonal, pela permutação de linhas e colunas. O objetivo das permutações é reordenar as variáveis de estado, de forma que cada entrada controle a parcela do vetor de estado correspondente ao seu índice de controlabilidade, sem conflito com as demais. As permutações correspondem às trocas de posição entre as colunas $A^j b_i$, $j = 0 \dots u; -1$ e $i = 1 \dots p$, na matriz U_c formada pelas n primeiras colunas linearmente independentes da matriz de controlabilidade U e a matriz M_c , construída da seguinte forma:

$$M_c = (b_1 \dots A^{u_1-1} b_1 \dots b_p \dots A^{u_p-1} b_p) \quad (3.6)$$

Para o exemplo obtém-se:

$$U_c = (b_1 \ b_2 \ b_3 \ Ab_1 \ Ab_2 \ Ab_3 \ A^2 b_2 \ A^1 b_3) \quad (3.7)$$

$$M = (b_1 \ Ab_1 \ b_2 \ Ab_2 \ A^2 b_1 \ b_3 \ Ab_3 \ A^2 b_3) \quad (3.8)$$

A matriz que efetua as permutações é, então:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.9)$$

Assim, a partir de (3.5) e utilizando-se a matriz de permutação P, obtém-se o par (A,B) na FCCM, como segue:

$$\bar{A} = \begin{pmatrix} x & x & w & w & w & w & w & w \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w & w & x & x & x & w & w & w \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ w & w & w & w & w & x & x & x \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}; \quad \bar{B} = \begin{pmatrix} 1 & x & x \\ 0 & 0 & 0 \\ 0 & 1 & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (3.10)$$

onde: os elementos "w" representam os elementos que devem ser anulados para se efetivar o desacoplamento.

Verifica-se, facilmente que as matrizes \bar{A} e \bar{B} têm a forma desejada, estando as submatrizes \bar{A}_{ii} , da diagonal numa forma canônica controlável monovariável. Uma análise da estrutura de desacoplamento mostra também que, apesar dos elementos $x \neq 0$ em \bar{B} , é possível anular os elementos w de \bar{A} , sem conflito com o posicionamento de pólos desejado.

3.5- Cálculo da MRE:

No cálculo da matriz de realimentação de estados para o par (A,B), realizado algebricamente, dois objetivos são alcançados: a efetivação do desacoplamento, transformando \bar{A} para a Forma

Bloco Diagonal, onde se anula os elementos w em (3.10), e o posicionamento dos pólos desejados, utilizando-se os coeficientes dos polinômios característicos de cada subsistema monovariável obtido com o desacoplamento e os coeficientes dos polinômios desejados, a partir do p -ésimo subsistema. A matriz $\hat{A} = (\bar{A} + \bar{B}\bar{K})$ correspondente é:

$$\hat{A} = \begin{array}{c|ccc|ccc} c & c & o & o & o & o & o & o \\ 1 & o & o & o & o & o & o & o \\ \hline o & o & c & c & c & o & o & o \\ o & o & 1 & o & o & o & o & o \\ o & o & o & 1 & o & o & o & o \\ \hline o & o & o & o & o & c & c & c \\ o & o & o & o & o & 1 & o & o \\ o & o & o & o & o & o & 1 & o \end{array} \quad (3.11)$$

onde: os elementos "c" representam os coeficientes dos polinômios característicos desejados.

A MRE em relação à base original é calculada levando-se em conta todas as transformações envolvidas para a obtenção da FCCM, como segue:

$$K = \bar{K}TRP \quad (3.12)$$

3.6- Conclusão:

Descreveu-se neste capítulo o algoritmo de Luenberger, destacando-se os aspectos conceituais. Este algoritmo, apesar de numericamente instável é conceitualmente simples, de fácil implementação e se constitui na base para o desenvolvimento do

Algoritmo de Patel (a.2) e do Algoritmo Proposto, apresentados nos próximos capítulos.

Uma análise na estrutura de desacoplamento da matriz de malha fechada, mostra que, ao invés de reduzi-la a uma forma diagonal, pode-se optar por uma forma triangular, superior ou inferior, dependendo dos valores dos IC. Isto permite reduzir a norma da MRE e conseqüentemente diminuir o esforço dispendido na ação de controle. Uma outra medida, com o objetivo de minimizar a instabilidade do algoritmo, é evitar o procedimento de obtenção dos elementos unitários, que é a principal fonte de instabilidade.

Capítulo 4

ALGORITMO DE PATEL

4.1- Introdução:

O algoritmo apresentado neste capítulo (Patel, 1985) é conceitualmente semelhante ao anterior, distribuindo o esforço de controle, no desacoplamento, entre todas as entradas. O desacoplamento é obtido reduzindo a representação do sistema original a uma Forma Bloco Triangular Inferior (FBTI). Inicialmente reduz-se o par (A,B) à Forma Bloco Superior Irreduzível de Hessenberg (FBSIH) e, a seguir, efetua-se uma redução à uma forma intermediária, onde todas as submatrizes têm a forma triangular/trapezoidal inferior, usando-se, adequada e iterativamente, pré-realimentações de estado e rotações de plano. Finalmente, através de permutações de linhas e colunas obtém-se a forma desejada. Uma das características principais deste algoritmo é que todos seus passos são executados através de procedimentos numericamente estáveis.

4.2- Redução a FBSIH e Cálculo dos Índices de Controlabilidade:

A redução à FBSIH é realizada em duas etapas. Na primeira é feita a condensação por linhas, reduzindo-se o par (A,B) à FBH mencionada em (2.19), através de uma matriz ortogonal, também denominada T . Os inteiros n_i (Índices de Kronecker), como no algoritmo anterior, são a base para o cálculo dos índices de

Controlabilidade. Na segunda etapa realiza-se a condensação por colunas, através de uma matriz ortogonal V , que tem a seguinte estrutura:

$$V = \text{diag} (Q_1 \cdot \cdot \cdot Q_{k-1} Q_k)_{n \times n} \quad (4.1)$$

onde: Q_i , com $i = 1, 2, \dots, k-1$, são matrizes ortogonais $n_i \times n_i$, tais que $Q_i A_{i,i-1} Q_{i-1}^T = (R_i \quad 0)$, $i = 2, 3, \dots, k$ e $Q_k = I_k$; k é o índice de controlabilidade do sistema.

A matriz que reduz o par (A, B) para a FBSIH, (A_{HI}, B_{HI}) , é dada, então por:

$$Z = VT \quad (4.2)$$

Após a redução, a matriz A tem a seguinte estrutura:

$$A_{HI} = Z A_H Z^T = \begin{vmatrix} A_{HI1,1} & A_{HI1,2} & \dots & A_{HI1,k-1} & A_{HI1,k} \\ A_{HI2,1} & A_{HI2,2} & \dots & A_{HI2,k-1} & A_{HI2,k} \\ 0 & 0 & \dots & A_{HI3,k-1} & A_{HI3,k} \\ \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & & \cdot & \cdot \\ 0 & 0 & \dots & A_{HIk-1,k-1} & A_{HIk-1,k} \\ 0 & 0 & \dots & A_{HIk,k-1} & A_{HIk,k} \end{vmatrix} \quad (4.3)$$

onde: $A_{HI,i,j} \in R^{n_i \times n_j}$;

os inteiros n_1, n_2, \dots, n_k , chamados índices de Kronecker, são definidos por $n_i = \text{posto} (A_{HI,i,i-1})$, com $i = 1, 2, \dots, k$;

as submatrizes $A_{n_i, i-1} = Q_i A_{i, i-1} Q_{i-1}^T$, com $i = 2 \dots k$, têm a forma $(R_{n_i} : 0)$ e $R_{n_i} \in R^{n_i \times n_i}$ são matrizes triangulares inferiores.

A matriz $B_{n_i} = ZB_n$, condensada somente por linhas, é da seguinte forma:

$$B_{n_i} = \begin{vmatrix} Q_i A_{1,0} \\ 0 \\ \vdots \\ 0 \end{vmatrix} \quad (4.4)$$

onde: $Q_i A_{1,0}$ é uma matriz sem qualquer estrutura definida.

Devido à necessidade que todas as submatrizes do par (A_{n_i}, B_{n_i}) tenham a forma triangular/trapezoidal inferior, determina-se, então, uma matriz ortogonal $Q_0 \in R^{l \times l}$, tal que:

$$Q_i A_{1,0} Q_0^T = (R_i) \quad (4.5)$$

onde: R_i é uma matriz triangular inferior.

Com isto o par (A_{n_i}, B_{n_i}) é reduzido para (A_{n_i}, B_0) onde $B_0 = B_{n_i} Q_0^T$. Resulta, então, que a MRE (\hat{K}) referida à base do par (A_{n_i}, B_0) deve ser multiplicada por Q_0^T , uma vez que os espectros $\sigma(A_{n_i} + B_0 \hat{K})$ e $\sigma(A_{n_i} + B_{n_i} K^*)$ devem ser iguais ao desejado, onde \hat{K} e K^* são as MRE referidas à base do par (A_{n_i}, B_0) e (A_{n_i}, B_{n_i}) , respectivamente. Tem-se então que:

$$B_0 \hat{K} = B_{n_i} K^* \quad (4.6)$$

Conseqüentemente:

$$B_{M1} Q_0^T \hat{K} = B_{M1} K^* \quad (4.7)$$

e

$$K^* = Q_0^T \hat{K} \quad (4.8)$$

Semelhantemente ao algoritmo anterior, o cálculo dos índices de controlabilidade é realizado utilizando-se os índices de Kronecker e observando-se a dependência linear entre as colunas das submatrizes $A_{M1}^i, i=1, \dots$. No entanto, com a nova definição das variáveis de estado, os IC são facilmente calculados e verifica-se que sempre se encontram em ordem decrescente de magnitude, pelo fato que as colunas linearmente dependentes das referidas submatrizes são sempre as últimas.

Considerando-se o exemplo anterior, a FBSIH é como segue:

$$A_{M1} = \begin{vmatrix} x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & 0 & 0 & x & x & x & x & x \\ x & x & 0 & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ 0 & 0 & 0 & x & 0 & 0 & x & x \\ 0 & 0 & 0 & x & x & 0 & x & x \end{vmatrix}; \quad B_0 = \begin{vmatrix} x & 0 & 0 \\ x & x & 0 \\ x & x & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \quad (4.9)$$

A tabela 4.1 a seguir, semelhante à apresentada anteriormente, ilustra o mecanismo de obtenção dos IC (μ_i) para o exemplo.

	b_{01}	b_{02}	b_{03}	
I	x	x	x	$n_1 = 3$
A_{M1}	x	x	x	$\hat{n}_1 = 3$
A_{M1}^2	x	x	0	$n_1 = 2$
	$\mu_1 = 3$	$\mu_2 = 3$	$\mu_3 = 2$	

Tabela 4.1. Cálculo dos índices de controlabilidade.

Como se observa, os IC são os mesmos do algoritmo anterior, uma vez que os IC do par (A,B) são invariantes sob qualquer transformação de similaridade e qualquer ordenamento das colunas de B (Chen,1984, teorema 5.8).

4.3- Redução à Forma Intermediária (\tilde{A},\tilde{B}) :

Esta redução visa tornar as submatrizes $A_{nr\ i,j}$, com $i = 1,2\dots k$ e $j = i,i+1\dots k$, triangular/trapezoidais inferiores, através de um processo iterativo realizado k vezes na seguinte sequência:

1) **pré-realimentação de estado:** que reduz a cada passo todas as submatrizes $A_{nr}^{(t)}\ i,j$, para $j = 1,2\dots k$, onde t representa o t -ésimo passo da iteração, à forma bloco triangular/trapezoidal inferior, sem alterar a estrutura de qualquer das outras submatrizes;

2) **rotação de plano:** que até o final do processo iterativo efetivará a redução das demais submatrizes.

A matriz de realimentação de estados \hat{K} calculada para o par (A_{nr}, B_o) é composta, então, de duas parcelas:

- \hat{K}_1 : parcela referente à pré-realimentação de estado, obtida na redução do par (A_{nr}, B_o) para a forma intermediária desejada;

- \hat{K}_2 : parcela correspondente ao posicionamento de pólos do sistema desacoplado, referida à base do mesmo par.

4.3.1- Pré-realimentação de Estado:

A matriz $\Theta^{(t)} \in R^{p \times n}$ de pré-realimentação de estado é determinada tal que:

$$\tilde{A}^{(t)} = A_{nr}^{(t)} - B_0^{(t)} \Theta^{(t)} \quad (4.10)$$

onde: $\tilde{A}_{i,j}^{(t)}$, com $j = 1, 2 \dots k$, são matrizes triangular/trapezoidais inferiores.

Da expressão (4.10) obtém-se o seguinte sistema de equações lineares:

$$B_0^{(t)} \Theta^{(t)} = A_{nr}^{(t)} - \tilde{A}^{(t)} \quad (4.11)$$

A solução deste sistema, de n^2 equações e $n \times p$ incógnitas, com $n > p$, sempre existe, uma vez que as n_1 primeiras equações são linearmente independentes e as demais equações não nulas são consistentes. Isto se deve ao fato que os elementos (i,j) , $i > n_1$ e $j > n_1$, de $B_0^{(t)} \cdot \Theta^{(t)}$, diferentes de zero, não alteram a estrutura das submatrizes $A_{nr}^{(t)}{}_{i,j}$, $i = 2, 3 \dots k$ e $j = 1, 2 \dots k$, devido à sequência apropriada de rotações efetuadas. Por outro lado, somente as primeiras $(n_1 - 1)$ linhas da matriz $\Theta^{(t)}$ são necessárias para reduzir as submatrizes $A_{nr}^{(t)}{}_{i,j}$, $j = 1, 2 \dots k$, para a forma desejada. Como consequência, contrariamente ao

É importante ressaltar que, embora a solução descrita acima seja obtida por procedimentos numericamente estáveis, o condicionamento do problema é modificado neste passo, uma vez que a pré-realimentação de estado corresponde, de fato, a um pré-posicionamento de pólos realizado sem compromisso com a sensibilidade final do problema.

4.3.2- Redução por Rotação de Plano:

As rotações de plano são efetuadas numa sequência bem definida, com sua operação inversa aplicada consecutivamente de modo a não alterar a forma já alcançada das submatrizes $A_{i,i-1}^{(t)}$ e reduzir à forma triangular/trapezoidal as submatrizes $A_{i,j}^{(t)}$, $i = 2, 3 \dots k$ e $j = i, i+1 \dots k$. Esta sequência é como segue:

1) para $t = 1, 2 \dots k-1$ e $j = 1, 2 \dots n-1$ anula-se os (j, n_t) -ésimos elementos das submatrizes $A_{l,q}^{(t)}$, $l = \alpha+1, \dots, 2$, $q = \alpha+1, \dots, 1$, por meio de rotação de plano, com os (j, j) -ésimos elementos das submatrizes triangular inferior $A_{l,l-1}^{(t)}$, sendo o maior inteiro tal que $n_\alpha = n_\alpha - 1 \dots = n_t$;

2) no final, quando $t = k$, e para $l = 3 \dots k$, $j = 1 \dots n_{l-1}$ e $q = 1 \dots k-1$, anula-se os (j, j) -ésimos elementos das submatrizes $A_{l,q}^{(k)}$, por meio de rotação de plano, com os (j, j) -ésimos elementos das submatrizes $A_{l+1,q}^{(k)}$.

Assim no final do processo iterativo, descritos anteriormenete, obtém-se:

$$\tilde{A} = RA_{MI}R^T - RB_o\hat{K}_1R^T \quad (4.14)$$

$$\tilde{B} = RB_o \quad (4.15)$$

onde: $\tilde{A} = A_{MI}^{(k+1)} = R_k A^{(k)} R_k^T$;

$$R = R_k R_{k-1} \dots R_1$$

$$\hat{K}_1 = \Theta_1 + \Theta_2 R_1 + \Theta_3 R_2 R_1 + \dots + \Theta_k R_{k-1} \dots R_1$$

Θ_t e R_t , com $t = 1 \dots k$, representam, para cada $t = 1 \dots k$ a pré-realimentação de estado e o total de rotações efetuadas em cada passo, respectivamente.

A expressão (4.14) mostra que os autovalores da matriz \tilde{A} são os mesmos da matriz $(A_{MI} - B_o\hat{K}_1)$.

Para o exemplo, no primeiro passo, obtem-se:

$$A_{MI}^{(1)} = \begin{array}{cccc|cccc} x & \otimes & \otimes & x & \otimes & \otimes & x & \otimes \\ x & x & \otimes & x & x & \otimes & x & x \\ x & x & x & x & x & x & x & x \\ \hline x & o & o & x & x & x & x & x \\ x & x & o & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ \hline o & o & o & x & o & o & x & x \\ o & o & o & x & x & o & x & x \end{array}; B_o^{(1)} = \begin{array}{ccc} x & o & o \\ x & x & o \\ x & x & x \\ \hline o & o & o \\ o & o & o \\ o & o & o \\ o & o & o \end{array} \quad (4.16)$$

onde: 1 e 2 representam rotações de plano que são efetuadas através de matrizes ortogonais R^1 e R^2 , respectivamente;

1' e 2' representam rotações de plano realizadas pelas transpostas de R^1 e R^2 , respectivamente;

⊗ representam, neste caso, os elementos que devem ser anulados por pré-realimentação de estado.

Neste passo, a matriz de pré-realimentação de estado é:

$$\Theta^{(2)} = \begin{vmatrix} 0 & x & x & 0 & x & x & 0 & x \\ 0 & x & x & 0 & x & x & 0 & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \quad (4.17)$$

No final obtém-se:

$$A_{HI}^{(2)} = \begin{vmatrix} x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & 0 & 0 & x & 0 & 0 & x & x \\ x & x & 0 & x & x & 0 & x & x \\ x & x & x & x & x & x & x & x \\ 0 & 0 & 0 & x & 0 & 0 & x & x \\ 0 & 0 & 0 & x & x & 0 & x & x \end{vmatrix}; B^{(2)} = \begin{vmatrix} x & 0 & 0 \\ x & x & 0 \\ x & x & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ x & x & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} \quad (4.18)$$

$$\text{onde: } A_{HI}^{(2)} = R_1^T A^{(1)} R_1;$$

$$B_o^{(2)} = R_1 B_o^{(1)};$$

$$R_1 = R_1 R_2.$$

O último passo do processo iterativo resulta em:

$$A_{HI}^{(3)} = \begin{vmatrix} x & x & 0 & x & x & 0 & x & x \\ x & x & 0 & x & x & 0 & x & x \\ x & x & x & x & x & x & x & x \\ x & 0 & 0 & x & 0 & 0 & x & 0 \\ x & x & 0 & x & x & 0 & x & x \\ x & x & x & x & x & x & x & x \\ 0 & 0 & 0 & x & 0 & 0 & x & 0 \\ x & 0 & 0 & x & x & 0 & x & x \end{vmatrix}; B_o^{(3)} = \begin{vmatrix} x & 0 & 0 \\ x & x & 0 \\ x & x & x \\ 0 & 0 & 0 \\ x & 0 & 0 \\ x & x & 0 \\ 0 & 0 & 0 \\ x & 0 & 0 \end{vmatrix} \quad (4.19)$$

A matriz de pré-realimentação de estado é:

$$\Theta^{(3)} = \begin{vmatrix} 0 & x & 0 & 0 & x & 0 & 0 & x \\ 0 & x & 0 & 0 & x & 0 & 0 & x \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \quad (4.20)$$

A forma intermediária (\tilde{A}, \tilde{B}) no final da redução é:

$$\tilde{A} = \begin{array}{|ccc|ccc|cc|} \hline x & o & o & x & o & o & x & o \\ x & x & o & x & x & o & x & x \\ x & x & x & x & x & x & x & x \\ \hline x & o & o & x & o & o & x & o \\ \hline x & x & o & x & x & o & x & x \\ x & x & x & x & x & x & x & x \\ \hline o & o & o & x & o & o & x & o \\ \hline x & o & o & x & x & o & x & x \\ \hline \end{array} ; \quad \tilde{B} = \begin{array}{|ccc|} \hline x & o & o \\ x & x & o \\ x & x & x \\ \hline o & o & o \\ \hline x & o & o \\ x & x & o \\ \hline o & o & o \\ \hline x & o & o \\ \hline \end{array} \quad (4.21)$$

$$\text{onde: } \tilde{A} = RA_{nr}R^T - RB_o\hat{K}_1R^T;$$

$$\tilde{B} = RB_o;$$

A matriz R que reduz o par (A_{nr}, B_o) à forma intermediária (\tilde{A}, \tilde{B}) e a parcela \hat{K}_1 , referente à pré-realimentação de estados, são dadas por:

$$R = R_1 R_2 R_3 \quad (4.22)$$

$$\hat{K}_1 = \theta_1 + \theta_2 R_1 + \theta_3 R_2 R_1 \quad (4.23)$$

4.4- Redução a FBTI:

A FBTI desejada é obtida pela permutação de linhas e de colunas na matriz A e de linhas na matriz B , através de uma transformação ortogonal. Este procedimento é conceitualmente semelhante ao apresentado no capítulo anterior e, como tal, visa reagrupar as variáveis de estado de acordo com os IC. Assim, após as permutações necessárias, a representação final do sistema desacoplado tem a seguinte estrutura:

$$\bar{A} = PAP^T = \begin{array}{|cccc|} \hline \bar{A}_{11} & 0 & \dots & 0 \\ \bar{A}_{21} & \bar{A}_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{A}_{p-1} & \bar{A}_{p-2} & \dots & \bar{A}_{p-1} \\ \hline \end{array} ; \quad \bar{B} = PB = \begin{array}{|cccc|} \hline \bar{b}_{11} & 0 & \dots & 0 \\ \bar{b}_{21} & \bar{b}_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \bar{b}_{p-1} & \bar{b}_{p-2} & \dots & \bar{b}_{p-1} \\ \hline \end{array} \quad (4.24)$$

onde: $A_{i,j}$ são matrizes de dimensão $\mu_i \times \mu_j$;

$b_{i,j}$ são vetores de dimensão $\mu_i \times 1$;

$A_{i,i}$ são matrizes na forma superior de Hessenberg;

$b_{i,i}$ são vetores de dimensão $\mu_i \times 1$,

com a seguinte estrutura:

$$b_{i,i} = (x \ 0 \ 0 \ \dots \ 0)^T.$$

As submatrizes $A_{i,j}$ e $b_{i,j}$, com $j = 1 \dots n_i - 1$ e $i = j + 1 \dots n_i$, não têm estrutura específica e não contribuem na alocação dos autovalores desejados.

Então, semelhantemente ao algoritmo anterior, para o exemplo, tem-se

$$U_p = (b_{01} \ b_{02} \ b_{03} \ A_{H1} b_{01} \ A_{H1} b_{02} \ A_{H1} b_{03} \ A_{H1}^2 b_{01} \ A_{H1}^2 b_{02}) \quad (4.25)$$

$$M_p = (b_{01} \ A_{H1} b_{01} \ A_{H1}^2 b_{01} \ b_{02} \ A_{H1} b_{02} \ A_{H1}^2 b_{02} \ b_{03} \ A_{H1} b_{03}) \quad (4.26)$$

A matriz P de permutações é:

$$P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (4.27)$$

Assim, a partir de (4.21) e utilizando-se a matriz P, obtém-se o par (\bar{A}, \bar{B}) na FBTI, como segue:

$$\bar{A} = \begin{array}{|cccc|cccc|} \hline x & x & x & o & o & o & o & o \\ x & x & x & o & o & o & o & o \\ o & x & x & o & o & o & o & o \\ \hline x & x & x & x & x & x & o & o \\ x & x & x & x & x & x & o & o \\ x & x & x & o & x & x & o & o \\ \hline x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ \hline \end{array}; \quad \bar{B} = \begin{array}{|ccc|} \hline x & o & o \\ o & o & o \\ o & o & o \\ \hline x & x & o \\ x & o & o \\ x & o & o \\ \hline x & x & x \\ x & x & o \\ \hline \end{array} \quad (4.28)$$

4.5- Cálculo da MRE (\bar{K}):

A estrutura genérica de \bar{K} visando manter o desacoplamento, diminuir sua magnitude e, conseqüentemente, melhorar o condicionamento do sistema resultante e o comportamento transitório das variáveis de estado, é:

$$\bar{K} = \begin{array}{|cccc|cccc|} \hline \bar{k}_1 & o & . & . & . & o \\ o & \bar{k}_2 & . & . & . & o \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ o & o & . & . & . & \bar{k}_p \\ \hline \end{array} \quad (4.29)$$

onde: $\bar{k}_i^T \in \mathbb{R}^{n_i}$, com $i = 1, 2, \dots, p$, são vetores obtidos nas soluções dos casos monovariáveis, representado pelos pares $(A_{i,i}, b_{i,i})$;

o são vetores nulos de dimensões apropriadas.

Assim, para o exemplo, tem-se:

$$\bar{K} = \begin{array}{|cccc|cccc|} \hline x & x & x & o & o & o & o & o \\ o & o & o & x & x & x & o & o \\ o & o & o & o & o & o & x & x \\ \hline \end{array} \quad (4.30)$$

4.6- Cálculo da MRE (K) em Relação a Base Original:

A partir de (4.140), (4.15) e (4.24), obtém-se:

$$\bar{A} = PRA_{M_I}R^T P^T - PRB_o K_i R^T P^T \quad (4.31)$$

$$\bar{B} = PRB_o \quad (4.320)$$

O cálculo da MRE \bar{K} é realizado de forma que o espectro da matriz de malha fechada $(\bar{A} - \bar{B}\bar{K})$ seja o desejado. Então, a partir de (.31) e (.32), obtém-se:

$$\bar{A} - \bar{B}\bar{K} = PRA_{M_I}R^T P^T - PRB_o (\hat{K}_i R^T P^T + \bar{K}) \quad (4.33)$$

ou ainda:

$$R^T P^T (\bar{A} - \bar{B}\bar{K}) PR = A_{M_I} - B_o (\hat{K}_i + \bar{K}PR) \quad (4.34)$$

Esta expressão mostra que os autovalores da matriz $(\bar{A} - \bar{B}\bar{K})$ são os mesmos da matriz $(A_{M_I} - B_o (\hat{K}_i + \bar{K}PR))$. Portanto, a matriz \hat{K} , citada em (4.3), é:

$$\hat{K} = \hat{K}_i + \bar{K}PR \quad (4.35)$$

onde: $\bar{K}PR = \hat{K}_i$.

Tendo em vista a equação (4.2) e (4.8), obtém-se:

$$K = Q_o^T (\hat{K}_i + \bar{K}PR) Z \quad (4.36)$$

Esta equação permite calcular a MRE para o par (A,B) , onde são consideradas todas as transformações efetuadas e a parcela referente às pré-realimentações de estado.

4.7- Conclusão:

O algoritmo apresentado neste capítulo (Patel,1985) utiliza a abordagem do desacoplamento com distribuição do esforço de controle entre todas as entradas como estratégia de solução, reduzindo a representação original do sistema à forma bloco

triangular inferior, utilizando unicamente procedimentos numericamente estáveis, tais como: transformações ortogonais de Householder, rotações de plano, pré-realimentação de estado e permutação de linhas e colunas. Esta estratégia é particularmente vantajosa para sistemas com um grande número de entradas, pois as ordens dos subsistemas monovariáveis obtidos com o desacoplamento, que correspondem aos índices de Controlabilidade do sistema, são consideravelmente menores

No entanto deve-se ressaltar que, embora o algoritmo seja conceitualmente simples como o descrito no capítulo anterior e de fácil implementação, a pré-realimentação de estado utilizada neste algoritmo, apesar de ser efetuada por procedimentos numericamente estáveis, modifica o condicionamento do problema. Isto porque esta corresponde, como já mencionado anteriormente, a um pré-posicionamento de pólos, neste caso realizado sem compromisso com a sensibilidade final do problema.

Apresentou-se, também, um procedimento para o cálculo da pré-realimentação de estado válido para todos os casos, enquanto que o apresentado por Patel em (Patel,1985) é válido quando $n_1 > n_2$.

Os exemplos apresentados no capítulo de "Implementação e Análise de Resultados" comprovam a influência da pré-realimentação de estado no comportamento numérico do algoritmo em termos de posicionamento efetivo dos autovalores desejados e da sensibilidade da matriz de malha fechada do sistema.

Capítulo 5

"ALGORITMO PROPOSTO"

5.1- Introdução:

O algoritmo aqui proposto é conceitualmente semelhante ao apresentado no capítulo 3, diferindo somente na etapa de redução à forma de Frobenius. As diferenças foram introduzidas com o objetivo de minimizar os passos numericamente instáveis do algoritmo de Luenberger e, conseqüentemente, obter um comportamento numérico satisfatório. Nesta proposta, a partir do par (A_n, B_n) na FBH, efetua-se uma redução à uma forma intermediária (FI) onde as submatrizes da subdiagonal principal têm a forma escalonada inferior e os elementos " x_i ;" não são transformados em elementos unitários, o que constitui a principal fonte de instabilidade numérica do algoritmo de Luenberger. A FI é obtida utilizando-se dois procedimentos: rotação de plano e eliminação gaussiana, a qual além de ser utilizada em um número reduzido de casos, é efetuada tomando-se como referência (pivot) o elemento de maior valor à esquerda do elemento a ser eliminado. Deste modo, este procedimento, apesar de não serem efetuadas permutações de linhas e colunas, corresponde a um pivoteamento parcial, garantindo-se um melhor comportamento numérico. Como nos casos anteriores, através de permutações de linhas e colunas, obtém-se uma forma controlável multivariável (FCM) desejada a partir da FI. A MRE é obtida algebricamente de forma que a matriz de malha fechada resultante, após o desacoplamento efetivo, tem uma forma bloco triangular inferior (FBTI).

5.2- Redução à Forma Intermediária (FI):

Com o objetivo de facilitar a descrição do procedimento de obtenção da FI, duas definições importantes, com base na matriz A_{μ} , são apresentadas à seguir:

Definição 1: chamam-se "Elementos de Fronteira Principal" os elementos $a(i,1)$, com $i = 1 \dots n$, pertencentes a submatriz $A_{:,1}$;

Definição 2: chamam-se "Elementos de Fronteira Secundária" os elementos $a(p,1)$, com $p = 1 \dots n$, pertencentes as submatrizes $A_{i,j}$, com $i = 2 \dots k$ e $j = 3 \dots k$, quando $i = 2$ ou $j = i \dots k$ nos demais casos.

Considerando-se então, o exemplo, tem-se:

$$A_{\mu} = \begin{array}{cccc|cccc} x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & \boxed{x} & x & x & \triangle & x \\ o & x & x & \boxed{x} & x & x & \triangle & x \\ o & o & x & \boxed{x} & x & x & \triangle & x \\ o & o & o & o & x & x & \triangle & x \\ o & o & o & o & o & x & \triangle & x \end{array} \quad (5.1)$$

onde: \boxed{x} são os elementos de fronteira principal;

$\triangle x$ são os elementos de fronteira secundária;

A forma intermediária desejada, obtida a partir da FBH em (2.19), tem a seguinte estrutura:

$$A_I = \begin{vmatrix} A_{I11} & A_{I12} & \dots & A_{I1, k-1} & A_{I1, k} \\ A_{I21} & 0 & \dots & 0 & 0 \\ 0 & A_{I22} & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & A_{Ik, k-1} & 0 \end{vmatrix} ; B_I = \begin{vmatrix} A_{I1\phi} \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{vmatrix} \quad (5.2)$$

onde: as submatrizes $A_{i, i-1}$, com $i = 1, \dots, k$, têm a forma escalonada inferior e os elementos correspondentes aos \underline{x}_i do par (A_M, B_M) são não-nulos e não necessariamente unitários.

Esta é obtida em duas etapas, como segue:

1) **Primeira etapa:** anulam-se as submatrizes $A_{i, j}$, para $j = k, k-1, \dots, 2$ e $i = j, j-1, \dots, 2$, segundo o seguinte procedimento:

- para $p = n_i, n_i-1, \dots, 1$ e $q = n_j, n_j-1, \dots, 1$ anula-se o (p, q) -ésimo elemento da submatriz $A_{i, j}$, por meio de rotação de plano com o $(p, q-1)$ -ésimo elemento, caso não seja de fronteira principal ou secundária. E anula-se estes por meio de eliminação gaussiana com o elemento a sua esquerda de maior valor \underline{x}_o , onde "o" corresponde à linha de A a qual pertence a p-ésima linha da referida submatriz;

2) **Segunda etapa:** reduzem-se as submatrizes $A_{i, i-1}$ à forma escalonada inferior, para $i = k, k-1, \dots, 2$, conforme o procedimento a seguir:

- para $p = 1, 2, \dots, n_i$; e $q = 1, 2, \dots, s-1$, onde s corresponde a coluna ocupada por \underline{x}_o , anula-se o (p, q) -ésimo elemento da submatriz

$A_{i,i-1}$ por meio de rotação de plano com o $(p,q-1)$ -ésimo elemento desta submatriz;

Após o processo de redução de cada submatriz A efetua-se a atualização do par (A_H, B_H) em transformação e no final obtém-se uma matriz Q , tal que:

$$A_I = QA_H Q^{-1} \quad ; \quad B_o = QB_H = \begin{vmatrix} B \\ \hline 0 \end{vmatrix} \quad (5.3)$$

onde: $B \in \mathbb{R}^{n_1 \times n_1}$ não tem estrutura definida.

Tendo em vista a necessidade de reduzir B à forma triangular superior, determina-se uma matriz ortogonal Q_o , tal que:

$$BQ_o = A_{to} \quad (5.4)$$

onde: A é uma matriz triangular superior.

Assim, para o exemplo, tem-se:

$$A_H = \begin{array}{cccccc|cccc} \text{x} & \text{x} & \text{x} & & \text{x} & \text{x} & \text{x} & & \text{x} & \text{x} \\ \text{x} & \text{x} & \text{x} & & \text{x} & \text{x} & \text{x} & & \text{x} & \text{x} \\ \text{x} & \text{x} & \text{x} & & \text{x} & \text{x} & \text{x} & & \text{x} & \text{x} \\ \hline \text{x} & & & & \text{x} & \text{x} & \text{x} & & \text{x} & \text{x} \\ & \text{x} & & & \text{x} & \text{x} & \text{x} & & \text{x} & \text{x} \\ & \text{o} & \text{x} & & \text{x} & \text{x} & \text{x} & & \text{x} & \text{x} \\ & \text{o} & \text{o} & \text{x} & & \text{x} & \text{x} & & \text{x} & \text{x} \\ \hline & \text{o} & \text{o} & \text{o} & \text{o} & \text{x} & \text{x} & & \text{x} & \text{x} \\ & \text{o} & \text{o} & \text{o} & \text{o} & \text{o} & \text{x} & & \text{x} & \text{x} \end{array} \quad (5.5)$$

onde: $\overset{i}{\curvearrowright}$ representa a i-ésima rotação;

$\underset{j}{\curvearrowleft}$ representa a j-ésima eliminação;

Obs.: supõe-se, neste caso, que \underline{x}_i seja o elemento de maior valor absoluto à esquerda do elemento a ser eliminado.

As matrizes A_r e B_r , após a redução resultam em:

$$A_r = \begin{array}{c} \left| \begin{array}{cccccccc} x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ x & x & x & x & x & x & x & x \\ \underline{x} & o & o & o & o & o & o & o \\ x & \underline{x} & o & o & o & o & o & o \\ x & x & \underline{x} & o & o & o & o & o \\ o & o & o & x & \underline{x} & o & o & o \\ o & o & o & x & x & \underline{x} & o & o \end{array} \right| \end{array} ; B_r = \begin{array}{c} \left| \begin{array}{ccc} \underline{x} & x & x \\ o & \underline{x} & x \\ o & o & \underline{x} \\ o & o & o \end{array} \right| \end{array} \quad (5.6)$$

Com relação à redução à FI, as seguintes considerações são importantes:

1) a anulação dos elementos de fronteira principal e de fronteira secundária por eliminação gaussiana, e não por rotação de plano, com qualquer um dos elementos a sua esquerda, visa manter o desacoplamento, uma vez que:

- no primeiro caso altera a estrutura de B_n , não permanecendo condensada por linhas;

- no segundo altera a estrutura das submatrizes $A_{i,j}$, $i = 3 \dots k$, $j = 1 \dots k-2$ e $i > j+1$.

2) o número de eliminações é minimizado nesta proposta e pode ser determinado por:

$$N = \sum_{i=1}^{k-1} (k - i)n_i + 1 \quad (5.7)$$

onde: N representa o número de eliminações efetuadas.

3) quanto menor k, menor o número de eliminações a serem efetuadas. Para o exemplo, tem-se 8 eliminações num total de 23 anulações, o que representa aproximadamente 33%. Para um sistema com a mesma ordem, no entanto com quatro entradas, $p = 4$ e $k = 2$, tem-se 4 eliminações num total de 22 anulações, representando 18%. Portanto, o desempenho do algoritmo aqui apresentado melhora a medida que aumenta o número de entradas. O mesmo não acontece com o algoritmo de Luenberger.

5.3- Redução à FCM:

A forma canônica multivariável desejada é obtida de modo idêntico ao do algoritmo apresentado no capítulo 3, através de uma matriz ortogonal de permutações P, e tem a seguinte estrutura:

$$\bar{A} = \begin{bmatrix} \bar{A}_{11} & \bar{Q}^* & \bar{Q}^* & \dots & \bar{Q}^* \\ \bar{A}_{21} & \bar{A}_{22} & \bar{Q}^* & \dots & \bar{Q}^* \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \bar{A}_{p1} & \bar{A}_{p2} & \bar{A}_{p3} & \dots & \bar{A}_{pp} \end{bmatrix}; \bar{B} = \begin{bmatrix} \bar{b}_{11} & \bar{b}_{12} & \dots & \bar{b}_{1p} \\ 0 & \bar{b}_{22} & \dots & \bar{b}_{2p} \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ 0 & 0 & \dots & \bar{b}_{pp} \end{bmatrix} \quad (5.8)$$

onde: as submatrizes $\bar{A}_{i,j}$, $\bar{Q}^*_{i,j}$, com $i = 1, 2, \dots, p$, $j = i+1, \dots, p$, têm as seguintes estruturas:

$$\bar{A}_{i,i} = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_{\mu_i-1} & \alpha_{\mu_i} \\ a_1 & 0 & \dots & 0 & 0 \\ 0 & a_2 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & a_{\mu_i-1} & 0 \end{pmatrix}_{\mu_i \times \mu_i} \quad (5.9)$$

$$O^{s_{i,j}} = \begin{pmatrix} x & x & \dots & x \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix}_{\mu_i \times \mu_j} \quad (5.10)$$

as submatrizes $\bar{A}_{i,j}$, com $i = 2 \dots p$, $j = 1 \dots p-1$ não têm estrutura definida;

os vetores $\bar{b}_{i,j}$, com $i = 1 \dots p$, $j = i \dots p$, têm a seguinte estrutura:

$$\bar{b}_{i,j} = (x \ 0 \ \dots \ 0)^T_{1 \times \mu_j} \quad (5.11)$$

Para o exemplo, tem-se:

$$\bar{A} = \begin{pmatrix} x & x & w & w & w & w & w & w \\ \underline{x} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ x & x & x & x & x & w & w & w \\ x & 0 & \underline{x} & 0 & 0 & 0 & 0 & 0 \\ 0 & \underline{x} & 0 & \underline{x} & 0 & 0 & 0 & 0 \\ x & x & x & x & x & x & x & x \\ x & 0 & x & 0 & 0 & \underline{x} & 0 & 0 \\ 0 & x & 0 & x & 0 & 0 & \underline{x} & 0 \end{pmatrix}; \quad \bar{B} = \begin{pmatrix} \underline{x} & x & x \\ 0 & 0 & 0 \\ 0 & \underline{x} & x \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \underline{x} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (5.12)$$

O polinômio característico $\Delta(\lambda)$ de $A_{i,i}$ em 5.9 é dado por:

$$\Delta(\lambda) = \lambda^{\mu_i} - \alpha_1 \lambda^{\mu_i-1} - a_1 \alpha_2 \lambda^{\mu_i-2} - \dots - a_1 a_2 \dots a_{\mu_i-1} \alpha_{\mu_i} \quad (5.13)$$

Isto é facilmente verificado, uma vez que $\bar{A}_{i,i}$ pode ser transformada através das operações elementares citadas em (3.3), em:

$$\bar{A}_{i,i}^* = \begin{pmatrix} \alpha_1 & a_1 \alpha_1 & a_1 a_2 \alpha_1 & \dots & a_1 a_2 \dots a_{p_i-2} \alpha_{p_i-1} & a_1 a_2 \dots a_{p_i-1} \alpha_{p_i} \\ 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \end{pmatrix}_{p_i \times p_i} \quad (5.14)$$

cujo polinômio característico é o apresentado acima (Chen,1984).

5.4- Cálculo da MRE:

Também neste algoritmo, o cálculo da matriz de realimentação de estado (\bar{K}) para o par (\bar{A}, \bar{B}) é realizado algebricamente e tem dois objetivos:

- a efetivação do desacoplamento, transformando A para a forma triangular inferior anulando-se os elementos w;
- o posicionamento dos pólos desejados, utilizando-se também os coeficientes dos polinômios característicos de cada subsistema monovariável obtido e os coeficientes dos polinômios desejados, a partir do p-ésimo subsistema.

Assim, a estrutura genérica de K é:

$$\bar{K} = \begin{pmatrix} \bar{k}_{11} & \bar{k}_{12} & \dots & \bar{k}_{1p} \\ 0 & \bar{k}_{22} & \dots & \bar{k}_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & \bar{k}_{pp} \end{pmatrix} \quad (5.15)$$

onde: $\bar{k}_{i,j} \in \mathbb{R}^{m_i}$, com $i = 1 \dots p$, são vetores obtidos na solução dos casos monovariáveis;

$\bar{k}_{i,j} \in \mathbb{R}^{n_j}$, com $i = 1 \dots p$ e $j = i+1 \dots p$, são vetores obtidos na anulação dos elementos w da correspondente submatriz $Q^*_{i,j}$ de \bar{A} , utilizando-se $b_{i,p}$;

Q representa um vetor nulo de dimensão apropriada.

Para o exemplo, tem-se:

$$\bar{K} = \left| \begin{array}{cc|ccc|ccc} x & x & x & x & x & x & x & x \\ o & o & x & x & x & x & x & x \\ o & o & o & o & o & x & x & x \end{array} \right| \quad (5.16)$$

Para um subsistema genérico $(A_{i,i}, b_{i,i})$, o vetor de realimentação de estado \bar{k} é dado por:

$$\bar{k}_j = - \frac{\bar{\alpha}_j + a_1 \cdot a_2 \dots a_{j-1} \cdot \alpha_j}{a_1 \cdot a_2 \dots a_{j-1} \cdot b_{o,i}} \quad (5.17)$$

onde: \bar{k}_j , com $j = 1 \dots \mu_i$, representa o j -ésimo elemento do vetor $\bar{k}_{i,i}$;

$\bar{\alpha}_j$ representa o j -ésimo coeficiente do polinômio característico desejado;

$b_{o,i}$ corresponde ao elemento "x" diferente de zero de $b_{i,i}$.

Deve-se ressaltar que os a_i , com $i = 1 \dots \mu_i$, são obrigatoriamente diferentes de zero, uma vez que o par (A,B) é controlável, e seus valores tanto podem ser maiores ou menores do que a unidade. Isto garante que, mesmo para μ_i elevado, o denominador de (5.17) não tende a zero, principalmente quando a distância ao mais próximo sistema não controlável é grande.

A MRE calculada para o par (A,B) é dada por:

$$K = Q_0 \bar{K} P Q T \quad (5.18)$$

onde: T é a matriz que reduz o par (A,B) à FBH.

5.5- Conclusão:

Apresentou-se neste capítulo um algoritmo para posicionamento de pólos por realimentação de estados em sistemas lineares multivariáveis, invariantes no tempo, que utiliza a abordagem do desacoplamento com distribuição do esforço de controle entre todas as entradas como estratégia de solução. Esta estratégia, além de despertar grande interesse teórico/didático, é particularmente vantajosa, como já mencionado, para sistemas com um grande número de entradas, pois as ordens dos subsistemas monovariáveis obtidos com o desacoplamento são consideravelmente menores.

Neste algoritmo, minimiza-se a utilização de procedimentos que alteram o condicionamento do problema, de modo que o seu comportamento numérico é satisfatório. Deve-se ressaltar que, apesar da existência de eliminação gaussiana no algoritmo, esta é realizada com pivoteamento e em um número reduzido de casos, o que garante o comportamento numérico desejado.

Os exemplos apresentados no próximo capítulo comprovam que, contrariamente ao algoritmo de Patel, o algoritmo proposto

apresenta um comportamento numérico satisfatório em termos de posicionamento efetivo de pólos, norma da MRE, sensibilidade da matriz de malha fechada e transitório das variáveis de estado. No entanto, deve-se ressaltar que, a norma da MRE apresentada por este algoritmo está intimamente ligada à propriedade de controlabilidade do sistema, isto porque os valores de \underline{x} , que influenciam no cálculo da MRE, estão relacionados com a controlabilidade do sistema.

Capítulo 6

IMPLEMENTAÇÃO E ANÁLISE DE RESULTADOS

6.1- Introdução:

Apresenta-se neste capítulo detalhes sobre a implementação computacional dos algoritmos, para o caso multivariável, descritos neste trabalho, e avalia-se, através de resultados, a robustez numérica das soluções encontradas. Esta avaliação é feita utilizando-se como critérios o número de condicionamento definido em (1), a norma de Frobenius da MRE na base original e o desvio entre os autovalores efetivamente posicionados e os desejados.

Os algoritmos foram implementados em microcomputador compatível IBM-PC, em simples precisão, usando-se linguagem Fortran-77 e levando-se em consideração a precisão finita da máquina. Adotou-se um conjunto de regras básicas para a codificação estruturada, foi pré-definido um plano de testes para a integração e validação de cada um dos algoritmos e utilizaram-se, na maior parte dos casos, procedimentos numericamente confiáveis e computacionalmente eficientes. Devido às características do problema tratado, dentre estes procedimentos, destacam-se:

1- os relacionados à operações gerais com matrizes: operações algébricas (soma, subtração, multiplicação, transposição e inversão); tratamento com submatrizes (criação e devolução à

matriz original); cálculo de norma; criação de matrizes identidade e nula; rotação de plano; etc.

2- os relacionados diretamente à **autoestrutura de matrizes**: neste caso utilizou-se rotinas de pacotes comprovadamente eficientes e confiáveis do ponto de vista numérico; destacam-se os pacotes EISPACK (Smith,1977) e LINPACK (Dongarra,1979), que se adaptam facilmente ao problema e são de domínio público.

Os resultados obtidos nos vários casos testados permitiram não só verificar a estabilidade numérica de cada um dos algoritmos, mas também, como mostrado neste capítulo, fazer uma análise comparativa detalhada entre eles.

6.2- Aspectos de Implementação:

A seguir descrevem-se os principais procedimentos de implementação da verificação computacional da controlabilidade e dos algoritmos descritos neste trabalho.

6.2.1- Verificação da controlabilidade:

Para a verificação computacional da controlabilidade, utilizou-se o método apresentado no ítem 2.5, por ser o numericamente mais eficiente e, também, por se constituir no primeiro passo nas transformações necessárias para os três

algoritmos descritos. Além disto, este efetua a decomposição canônica de Kalman.

Na sua implementação, a determinação de cada submatriz $A_{i, i-1}$ em (2.19), $i = 2, 3 \dots k$, é obtida pelo cálculo do posto da submatriz $A_{i-1, i-2}$, já condensada. A condensação por linhas é efetuada através de transformações ortogonais de Householder, utilizando-se uma versão modificada da rotina SQRDC do LINPACK (Dongarra, 1979). A necessidade desta adaptação, cujo pseudocódigo é mostrado a seguir, deve-se à exigência da obtenção destas submatrizes condensadas na forma escalonada superior. Isto facilita o cálculo do posto e permite, para o cálculo dos IC no algoritmo de Luenberger, identificar quais colunas são linearmente independentes (LI) nas referidas submatrizes.

Rotina SQRDCM: pseudocódigo

$j = 1$

para $i = 1$ até número de colunas faça

se (norma da coluna $i = 0$) então

- calcular a transformação de Householder para a coluna i , a partir da linha j ;
- aplicar a transformação nas colunas restantes;
- atualizar a matriz de transformação;
- $j = j + 1$

fim se

fim para

O cálculo dos n_i é efetuado a partir da forma escalonada pela análise de elementos que obrigatoriamente devem ser considerados nulos. Devido ao erro de arredondamento, deve-se considerar um desvio nestes elementos, levando-se em conta os seguintes parâmetros:

- número total de operações elementares realizadas;
- precisão da máquina;
- norma Euclidiana ou de Frobenius da matriz.

Este desvio pode ser calculado pela seguinte relação:

$$\mathcal{E} = \alpha \phi(n) \cdot 10^{-t} \cdot \|A\|_F \quad (6.1)$$

onde: α é uma constante = 1;

$\phi(n)$ é um polinômio em n e de baixo grau (utiliza-se grau 2, neste trabalho);

t é a precisão do computador em ponto flutuante.

A atualização do par (A,B) e da matriz de transformação no decorrer deste processo é facilmente realizada utilizando-se adequadamente os procedimentos de criação e devolução de submatrizes, e os de multiplicação de matrizes. Para facilitar a implementação e reduzir a ocupação de memória, utilizou-se, neste caso e no decorrer das outras implementações, duas rotinas distintas de multiplicação de matrizes, como segue:

Pré-multiplica: que realiza o produto AxB , entre duas matrizes genéricas A e B , retornando o resultado na segunda;

Pós-multiplica: que realiza o produto AxB , retornando o resultado na primeira.

6.2.2- Algoritmo de Luenberger:

Na implementação deste algoritmo, dois procedimentos destacam-se:

1) o cálculo dos IC (μ_i , $i = 1, 2, \dots, p$), onde p é o número de entradas efetivas, que é realizado através da análise da dependência linear entre as colunas de cada submatriz $A_{i, i-1}$ em (2.19), baseado na forma escalonada, durante o processo de obtenção da FBH. A identificação das colunas LD com as anteriores indica quais deixam de contribuir no cálculo, na próxima submatriz. Tal procedimento é representado pelo seguinte pseudocódigo:

```

m = 1
enquanto aux(m) = 0 faça
    m = m + 1
fim enquanto
l = 1
para j = 1 até b faça
    se (l < n1) então
        se ( $A_{i, i-1}(l, j) \neq 0$ ) então
            enquanto (aux(m) = 0) faça
                m = m + 1
            fim enquanto
    fim enquanto

```

IC(m) = IC(m) + 1

l = l + 1

m = m + 1

senão

aux(m) = 0

m = m + 1

fim se

senão

para m = 1 até nc faça

aux(m+1) = 0

fim para

fim se

fim para

onde: nc e nl representam o número de colunas e linhas da submatriz $A_{i,i-1}$, respectivamente;

aux(m) é um vetor auxiliar inicializado com elementos unitários;

IC(m) é o vetor que conterà os índices de controlabilidade.

2) a obtenção da FCCM, através da permutação de linhas e de colunas na matriz reduzida à FBF. A informação necessária para realização das permutações é obtida pelo seguinte procedimento:

$\beta(1) = 0$

para i = 1 até (k+1) faça

$\beta(i) = \beta(i-1) + n_i - 1$

fim para

j = 1

```

para i = 1 até p faça
  para m = 1 até i faça
    nca = 0
    para l = 1 até (i-1) faça
      se (  $\mu(l) < m$  ) então
        nca = nca + 1
      fim se
    fim para
    V(j) = i + (m) - nca
    j = j+1
  fim para
fim para

```

onde: n_{ca} corresponde ao número de colunas anteriores de U, LD com $A^{m-1}b$;

V(j) indica a linha e a coluna que deverá ser permutada com a linha e a coluna j, respectivamente;

$\beta(i)$ contém os IK.

6.3- Algoritmo de Patel:

Os principais procedimentos da implementação do algoritmo de Patel, são apresentados a seguir:

1) obtenção da FBSIH a partir da FBH, onde efetua-se a condensação por colunas das submatrizes $A_{i,i-1}$, em (2.19), para a forma triangular/trapezoidal inferior. Isto é realizado efetuando-se a condensação por linhas das transpostas das

submatrizes $A_{i,i-1}$, $i = k, k-1, \dots, 2$, onde k é o índice de controlabilidade do sistema, utilizando-se a rotina SQRDC (Dongarra, 1979) ou a modificação proposta. Considerando-se (2.19), a FBH condensada por linhas e colunas tem a seguinte estrutura:

$$A_{M,I} = \begin{vmatrix} Q_1 A_{1,1} Q_1 & Q_1 A_{1,2} Q_2 & \dots & Q_1 A_{1,k-1} Q_{k-1} & Q_1 A_{1,k} \\ Q_2 A_{2,1} Q_1 & Q_2 A_{2,2} Q_2 & \dots & Q_2 A_{2,k-1} Q_{k-1} & Q_2 A_{2,k} \\ 0 & Q_3 A_{3,2} Q_2 & \dots & Q_3 A_{3,k-1} Q_{k-1} & Q_3 A_{3,k} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \vdots & \dots & A_{k,k-1} Q_{k-1} & A_{k,k} \end{vmatrix} ;$$

$$B_o = B_{M,I} Q_o = \begin{vmatrix} Q_1 A_{1,o} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{vmatrix} \quad Q_o = \begin{vmatrix} R_{n,1} \\ 0 \\ \vdots \\ \vdots \\ 0 \end{vmatrix} \quad (6.2)$$

onde: $Q_i \in R^{n_i \times n_i}$ são matrizes ortogonais;

$$Q_i A_{i,i-1} Q_{i-1}^T = (R_{n_i} \ 0), \quad i = 1 \dots k, \quad Q_k = I_{n_k} \quad e$$

$R_{n_i} \in R^{n_i \times n_i}$ é triangular inferior.

O pseudocódigo deste procedimento é:

para $i = k$ até 2 **faça**

- condensa-se a submatriz $A_{i,i-1}$
- atualiza-se a matriz $A_{i,i}$
- atualiza-se a matriz P de transformação

fim para

2) **obtenção dos IC.** Semelhantemente ao algoritmo de Luenberger, o cálculo dos IC é realizado utilizando-se os IK e observando-se a dependência linear entre as colunas das submatrizes

condensadas de A_{HI} . Com a nova definição das variáveis de estado, os IC são facilmente calculados como segue :

```

para j = 1 até n faça
     $\mu(j) = 0$ 
fim para
para i = 1 até k faça
    para j = 1 até n faça
         $\mu(j) = \mu(j) + 1$ 
    fim para
fim para

```

3) obtenção da Forma Intermediária. Para obtenção da Forma Intermediária (FI), dada em (4.12) e (4.13), reduz-se as submatrizes $A_{HI\ i,j}$, com $i = 1, 2 \dots k$ e $j = i, i+1, \dots k$, à forma triangular/trapezoidal inferior através de um processo iterativo realizado k vezes, utilizando-se pré-realimentação de estados e rotação de plano. O pseudocódigo a seguir ilustra este procedimento:

```

para t = 1 até k faça
    - reduz-se as submatrizes  $A_{HI}^{(t)}\ i,j$ , com  $j = 1, 2 \dots k$ , à
      forma triangular/trapezoidal inferior por meio de
      pré-realimentação de estados;
    - reduz-se as submatrizes  $A_{HI}^{(t)}\ i,j$ , com  $i = 2 \dots k$ ,
       $j = 2 \dots k$  à forma triangular/trapezoidal inferior por
      meio de rotações de plano;
fim para

```

Também neste caso, com o mesmo objetivo citado em relação à multiplicação de matrizes, foram utilizadas duas rotinas distintas para realizar a rotação de plano, denominadas: pré-rotaciona e pós-rotaciona, que efetuam operações elementares somente sobre as linhas ou colunas envolvidas, respectivamente.

4) obtenção da FBTI, como já mencionado, através de permutações de linhas e colunas sobre a matriz na Forma Intermediária. Embora conceitualmente semelhante ao algoritmo de Luenberger, esta é de implementação mais simples, uma vez que as colunas ID nas submatrizes $A_{i,j-1}$ em $A_{n,r}$ são as últimas. Com isto, o pseudocódigo é idêntico ao apresentado para o caso anterior, exceto que $n_{i,j} = 0$.

6.2.4- Algoritmo Proposto:

Devido a semelhança do algoritmo proposto com o algoritmo de Luenberger, para a sua implementação foi necessário somente desenvolver o procedimento de obtenção da Forma Intermediária, cujo pseudocódigo é apresentado a seguir:

$c = \text{ordem}$

$n = 0$

para $i = k$ até 2 faça

$r = 0$

 para $j = i$ até 2 faça

$R = I_n$

$n_i = n_j$

$m = 0$

enquanto ($n_l = 0$) faça

$l = \text{ordem} - n - m - r$

$n_c = n;$

$c = \text{ordem} - n$

$t =$ coluna ocupada pelo elemento de maior valor da linha l à esquerda do elemento a ser eliminado;

enquanto ($n_c = 1$) faça

se ($c > t$) faça

- rotacionar o $(1,c)$ -ésimo elemento de A com o $(1,c-1)$ -ésimo elemento;

- atualizar as matrizes de transformação global, sua inversa e a matriz de transformação desta redução;

fim se

$n_c = n_c - 1$

$c = c - 1$

fim enquanto

se ($c > t$) então

- eliminar o $(1,c)$ -ésimo elemento de A com o $(1,t)$ ésimo elemento;

- atualizar as matrizes de transformação global, sua inversa e a matriz de transformação desta redução;

fim se

$m = m + 1$

```

         $n_c = n_c - 1$ 
    fim enquanto
    - atualizar a matriz A
     $r = r + n_j$ 
fim para
 $n = n + n_j$ 

fim para
 $n = 0$ 
para  $i = k$  até 2 faça
     $n = n + n_j$ 
     $l_w = \text{ordem} - n_j - n + 1$ 
     $R = I_n$ 
    para  $l_r = (\text{ordem} - n + 1)$  até  $(\text{ordem} - n + n_j)$  faça
         $n_c = n_j - 1$ 
         $c = \text{ordem} - n$ 
         $t = \text{coluna ocupada pelo elemento } x_{ij}, \text{ com } i = l_w;$ 
        enquanto  $(n_c = 1)$  faça
            se  $(c_r > t)$  então
                - rotacionar o  $(l_r, c_r)$ -ésimo elemento de
                  A com o  $(l_r, c_r - 1)$ -ésimo elemento;
                - atualizar as matrizes de transformação
                  global, sua inversa e a matriz de
                  transformação desta redução;
            fim se
             $n_c = n_c - 1$ 
             $c_r = c_r - 1$ 
        fim enquanto
         $l_w = l_w + 1$ 
    fim para

```

- atualizar a matriz A

fim para

onde: l e c representam a linha e coluna dos elementos das submatrizes $A_{i,j}$, $j = k, k-1 \dots 2$ e $i = j \dots 2$, os quais são anulados por rotação ou por eliminação;

l_r e c_r representam a linha e coluna dos elementos das submatrizes $A_{i,i-1}$, $i = k \dots 2$, os quais são anulados somente por rotação de plano;

l_n representa a linha na matriz A do elemento a ser anulado;

n_l e n_c representam o número de linhas e de colunas de cada submatriz $A_{i,j}$ que está sendo reduzida à forma desejada;

r , m e n são variáveis de controle que são utilizadas para indicar a correspondência entre as linhas e colunas de cada submatriz e da matriz A_n .

6.3- Análise de Resultados:

Os exemplos apresentados a seguir foram escolhidos entre os vários testados, visando avaliar a estabilidade numérica dos algoritmos descritos neste trabalho, e também dos algoritmos de Petkov e de Kautsky, em termos de posicionamento de pólos e dentro dos critérios mencionados em 2.3.2.

Exemplo 1. Coluna de destilação (Kautsky, 1982):

$$n = 5, p = 2$$

$$A = \begin{vmatrix} -0,1094 & 0,0628 & 0 & 0 & 0 \\ 1,3060 & -2,1320 & 0,9807 & 0 & 0 \\ 0 & 1,5950 & -3,1490 & 1,5470 & 0 \\ 0 & 0,0355 & 2,6320 & -4,2570 & 1,8550 \\ 0 & 0,00227 & 0 & 0,1636 & -0,1625 \end{vmatrix}$$

$$B^T = \begin{vmatrix} 0 & 0,0638 & 0,0838 & 0,1004 & 0,0063 \\ 0 & 0 & -0,1396 & -0,2060 & -0,0128 \end{vmatrix}$$

Autovalores desejados: $(-0,2; -0,5; -1,0; -1,0 \text{ j}1,0)$

Deve-se observar que este exemplo é apresentado em (Kautsky e outros, 1982; Petkov e outros, 1986) e portanto, serve como referência para verificação da estabilidade numérica dos algoritmos, uma vez que este é mal-condicionado e representa uma situação real. Os resultados são apresentados em forma de tabela, onde constam os principais parâmetros de avaliação, e para o algoritmo proposto, apresenta-se os pares (A_μ, B_μ) , (A_r, B_r) e (A, B) e a MRE (K) em relação à base original.

$$A_\mu = \begin{vmatrix} -1,0971 & 0,8847 & 1,7360 & 0,8880 & 0,7024 \\ 0,3002 & -2,6280 & -0,7901 & -1,0180 & 0,9899 \\ \underline{0,7817} & \underline{0,2165} & -5,7337 & 0,1299 & -0,0310 \\ 0 & \underline{-0,2380} & 0,9399 & -0,2437 & -0,00075 \\ 0 & 0 & \underline{0,3808} & -0,02908 & -0,1074 \end{vmatrix}$$

$$B_\mu^T = \begin{vmatrix} -0,1456 & 0 & 0 & 0 & 0 \\ 0,2230 & -0,1114 & 0 & 0 & 0 \end{vmatrix}$$

$$A_r = \begin{vmatrix} -6,7856 & 0,4476 & -1,1876 & -0,0226 & -0,0193 \\ 1,8687 & -3,0243 & -1,6448 & -0,8242 & -0,1134 \\ \underline{3,4986} & 0 & 0 & 0 & 0 \\ 1,5828 & \underline{0,2858} & 0 & 0 & 0 \\ 0 & 0 & \underline{1,2539} & 0 & 0 \end{vmatrix}$$

$$B_1^T = \begin{vmatrix} 0,0923 & 0 & 0 & 0 & 0 \\ -0,2094 & 0,1758 & 0 & 0 & 0 \end{vmatrix}$$

$$\bar{A} = \begin{vmatrix} -6,7856 & -1,1876 & -0,0193 & 0,4476 & -0,0226 \\ 3,4986 & 0 & 0 & 0 & 0 \\ 0 & 1,2539 & 0 & 0 & 0 \\ 1,8687 & -1,6448 & -0,1134 & -3,0243 & -0,8242 \\ 1,5828 & 0 & 0 & 0,2858 & 0 \end{vmatrix}$$

$$\bar{B}^T = \begin{vmatrix} 0,0923 & 0 & 0 & 0 & 0 \\ -0,2094 & 0 & 0 & 0,1758 & 0 \end{vmatrix}$$

$$K = \begin{vmatrix} -16,5050 & 140,2371 & -376,5744 & 296,1104 & 10,2348 \\ 0,2938 & 56,6945 & -167,0607 & 115,7138 & 43,6623 \end{vmatrix}$$

Na tabela 6.1, abaixo, são mostrados os autovalores efetivamente posicionados (λ_i) com seus desvios absoluto (τ_i) e relativo (e_i ;%) para os algoritmos de Luenberger, Patel e Proposto.

Algor	λ_i desejados	-0,2	-0,5	-1,0	-1,0 j1,0
Proposto	λ_i	-0,2	-0,5	-1,0001	-1,000 j1,00
	$\tau_i \times 10^5$	0,07	3,00	5,00	0,50
	$e_i \times 10^3$	0,35	6,00	5,00	0,35
Patel	λ_i	-0,2	-0,4998	-1,0003	-0,9999 j1,00
	$\tau_i \times 10^5$	1,00	20,00	30,00	10,00
	$e_i \times 10^3$	5,00	40,00	30,00	7,10
Luenberger	λ_i	-0,0294	0,8748	-1,7089	-1,000 j1,00
	$\tau_i \times 10^3$	*	*	70.000	0,01
	$e_i \times 10^3$	*	*	70.000	0,0071

Tabela 6.1. Pólos efetivamente posicionados e seus desvios absoluto e relativo. Exemplo 1.

Obs.: * representa um autovalor posicionado com um desvio superior a 100% do autovalor desejado.

Na tabela 6.2, abaixo, são mostrados o condicionamento da matriz de malha fechada e a norma de Frobenius da MRE também, para os algoritmos de Luenberger, Patel e Proposto.

Algoritmo	$k(A + BK)$	$\ K\ _F$
Proposto	1.402	542
Patel	1.529	141
Luenberger	4.364	480

Tabela 6.2. Condicionamento da matriz de malha fechada e norma de Frobenius da MRE

Os resultados apresentados na tabela 6.1 mostram a instabilidade numérica do algoritmo de Luenberger e a superioridade em termos de posicionamento efetivo de pólos do algoritmo proposto com relação ao algoritmo de Patel. A tabela 6.2 mostra que a sensibilidade da matriz de malha fechada para o algoritmo proposto é inferior à apresentada pelo algoritmo de Patel, conseqüentemente, as variáveis de estado, para o sistema realimentado com a MRE obtida por este algoritmo apresentará um melhor comportamento transitório (Kautsky, 1985). Embora a norma de Frobenius da MRE, para o algoritmo proposto, seja superior as apresentadas pelos demais algoritmos, deve-se ressaltar que a sensibilidade da matriz de malha fechada, que

indica a robustez da solução é um critério mais importante, podendo-se então concluir, para este exemplo, que o algoritmo proposto apresentou resultados qualitativamente superiores aos algoritmos de Patel e Luenberger, os quais utilizam a mesma abordagem de solução (Kautsky, 1985).

Exemplo 2. Sistema Máquina-barras infinita (Jorge Dantas, 1983)

$$n = 6, p = 2$$

$$A = \begin{pmatrix} 0 & 377 & 0 & 0 & 0 & 0 \\ -0,0970 & 0 & -0,1155 & 0 & 0,1000 & 0 \\ -0,2849 & 0 & -0,5518 & 0,1695 & 0 & 0 \\ 0 & 0 & 0 & -25 & 0 & 0 \\ 0 & 0 & 0 & 0 & -3,3333 & 3,3333 \\ 0 & 0 & 0 & 0 & 0 & -12,5 \end{pmatrix}$$

$$B^T = \begin{pmatrix} 0,9704 & 0 & 1,1551 & 0 & 0 & 0 \\ -0,0380 & 0 & 0,4814 & 0 & 0 & 0 \end{pmatrix}$$

Autovalores desejados: (-11,1367 j1,2350;
-9,0000 j9,0000 e
-8,8633 j1,0319)

Na tabela 6.3, abaixo, são mostrados os autovalores efetivamente posicionados (λ_i) para os algoritmos de Luenberger, Patel, Proposto, Petkov e Kautsky.

Algor	-9,0000 j9,0000	-11,1367 j1,2350	-8,8633 j1,0319
Luenberger	-8,9998 j8,9997	-33,8690* -12,5300*	-4,1911 j2,5097
Patel	-9,0000 j9,0000	-10,3965 j2,5152	-3,6272* -12,246*
Proposto	-8,9997 j7,2010	-11,1678 j1,2558	-8,8325 j1,0613
Petkov	-9,0194 j8,9548	-11,0753 j1,1967	-8,9046 j1,1302
Kautsky	-9,0000 j9,0000	-11,1367 j1,2350	-8,8633 j1,0319

6.3 - Autovalores efetivamente posicionados. Exemplo 2.

Obs.: * representa um autovalor posicionado com um desvio superior a 100% do autovalor desejado.

Na tabela 6.4, abaixo, são mostrados o condicionamento da matriz de malha fechada e a norma de Frobenius da MRE para os algoritmos de Kautsky, Petkov e Proposto. Nesta não são apresentados os resultados obtidos com o algoritmo de Patel e Luenberger, tendo em vista que estes apresentaram instabilidade numérica.

Algoritmo	$k(A + BK)$	$\ K\ _F$
Proposto	29.905	3.560
Petkov	13.497.000	84.253
Kautsky	17.227	2.523

Tabela 6.4 - Condicionamento da matriz de malha fechada e norma de Frobenius da MRE

Analisando-se os resultados apresentados nas tabelas 6.3 e 6.4, conclui-se que o algoritmo proposto é numericamente eficiente e o seu comportamento mostrou-se superior, não só ao algoritmo de Patel, como também ao de Petkov. Neste exemplo, a influência da pré-realimentação de estado no comportamento numérico do algoritmo de Patel ficou evidenciada pela instabilidade numérica apresentada.

6.4- Conclusão:

Apresentou-se, neste capítulo, detalhes sobre a implementação computacional dos algoritmos de Luenberger, Patel e Proposto, que utilizam a abordagem de desacoplamento como solução do problema de posicionamento de pólos por realimentação de estados em sistemas lineares multivariáveis, invariantes no tempo.

Resultados obtidos numa extensa bateria de testes permitem concluir que o algoritmo proposto é numericamente eficiente, apresentando sempre uma MRE que efetivamente posiciona os pólos do sistema onde desejado e por utilizar a abordagem de desacoplamento, a norma da MRE e a sensibilidade da matriz de malha fechada atendem os critérios citados em 2.4.2.

Capítulo 7

CONCLUSÃO E PERSPECTIVAS

O objetivo deste trabalho foi o de apresentar um estudo dos aspectos teóricos e numéricos do problema de posicionamento de pólos por realimentação de estados em sistemas lineares multivariáveis, invariantes no tempo.

Primeiramente foram mostrados os aspectos teóricos do problema, destacando-se sua formulação e as formas de abordagem, e os aspectos numéricos básicos. Destes, foram dados os conceitos de:

- condicionamento de um problema;
- estabilidade numérica de um algoritmo;

e apresentadas as transformações ortogonais utilizadas:

- transformação de Householder;
- rotação de plano.

Propuseram-se, então, critérios para avaliação das soluções encontradas, e verificou-se que, dentre, o de estabilidade numérica do algoritmo é o mais representativo, uma vez que este indica o posicionamento efetivo dos pólos. No entanto, o critério da robustez numérica permite validar a eficiência da lei de controle. Como a adoção destes critérios reduz o número de algoritmos numericamente eficientes, apresentou-se, sucintamente, os principais algoritmos numericamente estáveis para os casos monovariável e multivariável.

Pela importância da abordagem de desacoplamento do sistema multivariável para solução, apresentaram-se detalhadamente os aspectos conceituais, numéricos e de implementação de dois algoritmos baseados nesta abordagem:

- Algoritmo de Luenberger;
- Algoritmo de Patel.

O utilizado no primeiro, para obter a Forma Canônica Controlável utiliza procedimentos numericamente instáveis, apresentou, e conseqüentemente, apresenta um comportamento numérico insatisfatório. O algoritmo de Patel, utiliza somente procedimentos numericamente estáveis, mas, no entanto, como verificado através de resultados, tem uma robustez numérica comprometida, devido à utilização de pré-realimentação de estado.

Propôs-se, então, um algoritmo também baseado na abordagem de desacoplamento com distribuição dos esforços de controle que, embora utilize eliminação gaussiana com pivoteamento parcial em um número reduzido de casos, apresentou desempenho numérico superior aos dois algoritmos citados anteriormente. Este, consiste em desacoplar o sistema pela redução à uma forma canônica multivariável particular, tal que a matriz de malha fechada, na base final, tem uma forma triangular inferior.

Deve-se ressaltar, no entanto que, considerando-se o que foi discutido neste trabalho, a natureza mal-condicionada do problema e a extrema sensibilidade dos autovalores da matriz de malha fechada à perturbações nos parâmetros da MRE, conclui-se

que o problema de posicionamento de pólos por realimentação de estados ainda é uma questão não totalmente resolvida. Isto porque não existe um algoritmo que, embora numericamente estável, resolva o problema integralmente, com todas suas características analisadas neste trabalho.

Como perspectivas de continuidade de estudos e pesquisas nesta área, propõe-se:

- utilização da idéia básica do algoritmo de Patel para o caso monovariável, que trabalha na alocação dos autovalores com a matriz de malha aberta, para a solução do caso multivariável, em conjunto com a idéia central do algoritmo de Miminis para o caso multivariável;
- redução, no algoritmo de Patel, do número de elementos eliminados por pré-realimentação de estado, por exemplo, utilizando-se somente pré-realimentação para redução da submatriz $A_{1,1}$. As demais submatrizes $A_{i,j}$ $j = 2,3\dots k$, seriam reduzidas por rotação de plano.
- especificação e desenvolvimento de um pacote de Projeto Assistido por Computador para Sistemas de Controle, direcionado ao PPRE. Este deverá conter algoritmos numericamente estáveis, para os casos monovariável e multivariável, um conjunto de ferramentas auxiliares para avaliação das soluções (cálculo de normas, de autovalores e autovetores, simulação temporal, número de condicionamento, saídas gráficas, etc.) e auxílios ao usuário, do ponto de vista conceitual e numérico.

REFERÊNCIAS

- Caso Monovariável:

Mimimis, G.S. e Paige, C.C.. "An Algorithm for Pole Assignment of Time Invariant Linear Systems", Int. J. Control, Vol. 35, n 2, 1982: 341-354.

Mimimis, G.S. "Numerical Algorithms for Controllability and Eigenvalue Allocation", M. Sc. Thesis, McGill University, Canada, 1981.

Petkov, P.H.; Christov, N. e Konstantinov, M.M.. "A Computational Algorithm for Pole Assignment of Linear Single Input Systems", IEEE TAC, Vol. AC 29, n 11, 1984: 1045-1048.

- Caso Multivariável:

Anderson, B.D.O. et Luenberger, D.G. "Design of Multivariable Feedback Systems", Proc. IEEE, Vol. 114, n 3, 1967.

Cavin, R.K.III et Bhattacharyya. "Robust and Well Conditioned Eigenstructured Assignment via Sylvester's Equations", Proc. American Control Conference, Arlington, Va 1982.

Kautsky, J.; Nichols, N.K. e Van Dooren, P.. "Robust Pole Assignment in Linear State Feedback", Int.J. Control, Vol.41, n 5, 1985: 1129-1155.

Miminis, G.S. "Numerical Algorithms for the Pole Placement Problem", Ph.D. Thesis, McGill University, Canada, 1984.

Miminis, G.S. e Paige, C.C.. "An Algorithm for Pole Assignment of Time Invariant Multi-Input Linear Systems", Proc. 21st CDC, Vol. 1, 1982: 62-67.

Patel, R.V.. "Algorithms for Eigenvalue Assignment in Multivariable Systems", em (Jamishidi, 1985): 315-345.

Patel, R.V. "Computational Algorithms for Pole Assignment in Linear Multivariable Systems", IFAC Symposium on Computer Aided Design of Multivariable Technological Systems, Purdue University, USA.

Patel, R.V. e Misra, P.. "Numerical Algorithms for Eigenvalue Assignment by State Feedback", Proc. IEEE, Vol. 72, n 12, 1984: 1755-1763.

Petkov, P.H.; Christov, N. e Konstantinov, M.M.. "A Computational Algorithm for Pole Assignment of Linear Multi-Input Systems", IEEE TAC, Vol. AC 31, n 11, 1986: 1044-1047.

- Artigos Teóricos-Básicos:

Eising, R. "Pole Assignment, a New Proof and Algorithms",
Eindhoven University of Technology, Netherlands, 1981.

Paige, C.C. "Properties of Numerical Algorithms Related to
Computing Controllability", IEEE TAC, Vol. AC 26, n 1,
1981.

Van Dooren, P.M. "The Generalized Eigenstructure Problem in Linear
System Theory", IEEE TAC, Vol. AC 26, n 1, 1981.

- Livros:

Chen, C.T. "Linear System Theory and Design", Holt, Rinehart e
Winston, 1984: Caps. 5 e 7, 1984.

Dongarra, J.J.; Moler, C.B.; Bunch, J.R. e Stewart, G.W. "LINPACK
User's Guide", SIAM, Philadelphia, 1979.

Jamishidi, M. e Herget, C.J., Computer-Aided Control Systems
Engineering, North-Holland, 1985

Smith, B.T. et Alli. "Matrix Eigensystem Routines - EISPACK Guide"
and "EISPACK Guide Extension", Lectures Notes in Computer,
New York, 1976 e 1977.

Stewart,G.W.."Introduction to Matrix Computations", Academic Press, 1973.

Strang,Gilbert."Linear Algebra and It's Applications", Academic Press - 1980.

Wilkinson,J.H.."The Algebraic Eigenvalue Problem", Oxford Academic Press,1965.

Wonham,W.M."Linear Multivariable Control: A Geometric Aproach", New York: Springer - 1979