COBEM 99
XV Congresso Brasileiro de Engenharia Mecânica
15th Brazilian Congress of Mechanical Engineering
22 - 26 de Novembro de 1999 / November 22 - 26, 1999 Águas de Lindóia, São Paulo,

FORMAS SEQUENCIAL E PARALELA PARA SOLUÇÃO DA EQUAÇÃO ALGÉBRICA DE RICCATI POR UM ALGORITMO DE SCHUR-MODIFICADO

Annabell Del Real Tamariz
Celso P. Bottura
João V. Fonseca Neto
Gilmar Barreto
Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas
Cx. P. 6122 - 13083-970 - Campinas, SP, Brasil

Resumo. *Neste trabalho uma proposta de solução numérica sequencial e paralela da Equação Algébrica de Riccati é apresentada. Este algoritmo apresenta uma paralelização para uma implementação especial do Método de Schur (Laub,1979), onde a matriz simplética é modificada através de Transformações de Similaridade Elementares Estabilizadas. No primeiro módulo do algoritmo a matriz simplética é transformada para a Forma Superior de Hessenberg. No segundo módulo a Forma Superior de Hessenberg é transformada à Forma Real de Schur pelo método QR-Francis, em forma paralela. Nosso objetivo é obter uma única solução simétrica definida não negativada equação algébrica de Riccati.*

Palavras-chave: *Equação Algébrica de Riccati, Sistemas de Controle, Computação Paralela e Distribuída, Identificação de Modelos.*

1. INTRODUÇÃO

Neste trabalho propomos um algoritmo para solução da *Equação Algébrica de Riccati* que modifica a matriz simplética, bem como sugerimos uma paralelização do novo algoritmo "Método de Schur-Modificado utilizando Transformações de Similaridade Elementares Estabilizadas".

O trabalho está dividido em duas partes; na primeira parte a matriz simplética é transformada à *Forma Superior de Hessenberg (FSH)* através de uma Transformação de Similaridade Estável utilizando matrizes elementares estabilizadas. Desta forma garante-se estabilidade numérica e ganha-se em eficiência sobre as outras formas de realizar a redução (Transformação de Givens ou de Householder) em relação ao número de operações realizadas (Jennings & McKeown,1993).

A partir da matriz em *Forma Superior de Hessenberg* obtida calculamos uma matriz em *Forma Real de Schur*; este processo é realizado em forma paralela para aumentar a velocidade do processamento, pois só precisamos informação local sobre a matriz atual transformada, o que admite uma implementação paralela mais eficiente.

Uma das aplicações do algoritmo implementado é viabilizar a identificação de modelos no espaço de estado com múltiplas entradas-múltiplas saídas ($MIMO$), linear e invariante no tempo a partir de medidas de entrada e saída. A identificação de modelos no espaço de estados é hoje um problema de central importância em análise de sistemas, projeto e controle e processamento de sinais. A realização eficiente do método aqui proposto através de processamento paralelo e distribuído viabiliza aplicações em identificação de sistemas que necessitam processamento em tempo real, e esta é uma motivação importante deste trabalho.

2. UM MÉTODO DE SCHUR MODIFICADO

Esta seção descreve o *Método de Schur Modificado* proposto para a *Equação Algébrica de Riccati Discreta (EARD)*:

$$A^T X A - X - A^T X G_1 (G_2 + G_1^T X G_1)^{-1} G_1^T X A + Q = 0 \tag{1}$$

para a qual existe, sob certas condições, uma solução simétrica e definida não negativa $X \in R^{n \times n}$, (Laub, 1979).

Para a EARD associamos a matriz simplética:

$$H_{symp} = \begin{bmatrix} A + G A^{-T} Q & -G A^{-T} \\ -A^{-T} Q & A^{-T} \end{bmatrix} \tag{2}$$

onde $G = G_1 G_2^{-1} G_1^T$.

Quatro passos são considerados no algoritmo:

1. A matriz simplética do sistema H_{symp} é formada;
2. H_{symp} é transformada numa matriz em forma superior de Hessenberg H_{Hes};
3. H_{Hes} é transformada numa matriz em forma real de Schur H_{schur};
4. Finalmente, a partir da matriz de transformação Schur obtem-se a solução da $EARD$.

A redução de H_{symp} à forma superior de Hessenberg H_{Hes}, é feita com transformações de similaridade elementares estabilizadas (*TSEE*), Martin & Wilkinson (1968). O método transforma a matriz simplética em uma matriz em forma superior de Hessenberg *FSH*, com os mesmos autovalores originais, Eq. (3).

$$H_{Hes} = Q H_{symp} Q^{-1} \tag{3}$$

onde Q é uma matriz não singular e não ortogonal da mesma ordem que H_{Hes}.

Obtida a matriz em forma superior de Hessenberg H_{Hes} e a correspondente matriz de transformação Q, o seguinte passo é obter a forma real de Schur (*FRS*) H_{schur} pelo método QR-Francis, (Golub,1989). Acha-se uma matriz ortogonal V_s que transforma a matriz H_{Hes} em *FRS* H_{schur}:

$$H_{schur} = V_s^{-1} H_{Hes} V_s = \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \tag{4}$$

De forma geral o algoritmo QR-Francis devolve os autovalores da matriz de *Hessenberg*, uma matriz H_{schur} em *FRS* e uma matriz V_s com os correspondentes autovetores Schur.

Tendo obtido as duas expressões Eq. (3) e Eq. (4), pode-se fazer uma substituição de Eq. (3) em Eq. (4) e obter a Eq. (5), que permite achar a matriz em *FRS* diretamente da matriz simplética do sistema.

$$H_{schur} = U^{-1} H_{symp} U \tag{5}$$

onde

$$U = Q^{-1} V_s = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \tag{6}$$

Posteriormente é necessário reordenar os autovalores na matriz H_{schur}, Eq. (5), pois o algoritmo não garante um ordenamento em ordem descendente dos autovalores que estão dentro do círculo unitário. Após o reordenamento tem-se que T_{11} têm os autovalores λ em ordem descendente e T_{22} têm os autovalores $1/\lambda$.

Finalmente, a solução da *EARD*, Eq. (7), pode ser obtida da matriz de autovetores Schur (Vaughan, 1970), (Laub, 1979).

$$X = U_{21} U_{11}^{-1} \tag{7}$$

3. ALGORITMO

O algoritmo para achar a solução da equação algébrica de Riccati pode ser esquematizado em cinco passos fundamentais, que tem início com a formação da matriz simplética e termina com a solução de Riccati através da transformada de Schur ordenada. Este algoritmo garante uma solução simétrica, única e definida não negativa para *ERAD*.

No passo 1 do algoritmo a matriz simplética é montada a partir do sistema em variáveis de estado discretas. O segundo passo do algoritmo realiza a transformação da matriz simplética H_{symp} a forma superior de Hessenberg H_{Hes} utilizando transformações de similaridade elementares estabilizadas (*TSEE*) e com isto o número de operações de multiplicação efetivadas é reduzido em $\frac{15n^3}{6}$ e $\frac{5n^3}{6}$ operações em relação às transformações ortogonais de Givens e Householder.

Simultaneamente ao processo de redução para Hessenberg, as matrizes de transformação não ortogonais são montadas para serem utilizadas para obter a partir da matriz

simplética a matriz de transformação da forma real de Schur ordenada.

1. Montar a matriz Simplética H_{symp}, Eq (2)
2. $H_{Hes} \leftarrow H_{symp}$.
   Fazer $k \leftarrow 1, N-2$
   Fazer $i \leftarrow k+1, N$
   $\mathrm{k}^{'} = max|H_{symp}{}^{(k)}_{(i,k)}|$
   $Linha\ \mathrm{k}^{'} \leftarrow$ Linha i
   $Linha$ i $\leftarrow$ Linha $k^{'}$
   Fazer l $\leftarrow k+2, N$
   $\mathrm{n}_{l,k+1} = H_{symp}{}^{(k)}_{(l,k)} / H_{symp}{}^{(k)}_{(k+1,k)}$
   Fazer j $\leftarrow 1, N$
   Se $\mathrm{n}_{l,k+1} \neq 0$ **Fazer**
   $\mathrm{H}_{Hes(i,j)} = H_{Hes(i,j)} - n_{i,k+1} * H_{symp(k+1,j)}$
   $\mathrm{H}_{Hes(j,k+1)} = H_{Hes(j,k+1)} + n_{i,k+1} * H_{symp(j,i)}$
   Fim Se
   Fim Fazer
   Fim Fazer
   Montar as matrizes de transformação Q_{Hes} e Q^{-1}_{Hes}
   Fim Fazer
   Fim Fazer
3. Forma Real de Schur $Schur_1 \leftarrow V_s^{-1} H_{Hes} V_s$, Eq (4).
4. Ordenar a forma real de Schur descrita acima e obter $H_{schur} \leftarrow U^{-1} H_{symp} U$.
5. Calcular a solução da equação a partir da expressão $X \leftarrow U_{21} U_{11}^{-1}$.

No passo 3 obtém-se a forma real de Schur utilizando o método *QR-Francis*; com isto obtém-se uma matriz de transformação ortogonal que junto com a matriz não ortogonal obtida da transformação de Hessenberg é utilizada para obter a transformação do passo 4 que leva a solução da *EARD* no passo 5.

4. IMPLEMENTAÇÃO

Para fazer o algoritmo paralelo, utilizamos uma biblioteca pública de rotinas paralelas que ajudam o cálculo da solução de *EARD*. A versão paralela apresentada neste trabalho baseia-se numa arquitetura de computador paralelo virtual *SIMD (única instrução - múltiplos dados)*, onde todos os processadores participantes do processamento executam a mesma instrução simultaneamente sobre vários dados.

De forma semelhante, apresenta-se a seguir o algoritmo paralelo implementado para obter a solução da *EARD*.

1. Incializar o ambiente de programação paralela (MPI):
   - Pode-se trabalhar num intervalo de processadores de 1 até 4 aproximadamente.
   - MPI_INIT;
   - MPI_COMM_RANK;
   - MPI_COMM_SIZE;
2. Se é processador principal, fazer:
   - Formação da matriz Simplética H_{symp} do sistema a resolver;
   - Transformação paralela de H_{symp} numa matriz em forma superior de Hessenberg H_{Hes} baseada fundamentalmente em operações do tipo matriz-vetor;
3. Inicializar a topologia utilizada pelas subrotinas empregadas para fazer os cálculos;
4. Definir o descritor das matrizes;
   - Definir dimensão do bloco;
   - Ordem das matrizes
   - Identificador onde o primeiro bloco da matriz é atribuído;
5. Calcular a matriz em forma real de Schur H_{schur};
6. Ordenar os autovalores de H_{Hes} na forma real de Schur descrita acima;
7. Se é processador principal:
   - Calcular a solução da *EARD*.

5. RESULTADOS

Nesta seção é mostrado um exemplo de um sistema discreto a ser resolvido. O objetivo do exemplo é mostrar como os resultados obtidos com o algoritmo são idênticos aos resultados obtidos através de outros algoritmos já clássicos na literatura. Para atingir este objetivo, apresentamos o exemplo de Laub (1979), onde:

$$A = \begin{bmatrix} 0.9512 & 0.0 \\ 0.0 & 0.9048 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} 4.877 & 4.877 \\ -1.1895 & 3.569 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 0.33 & 0.0 \\ 0.0 & 3.0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0.005 & 0.0 \\ 0.0 & 0.02 \end{bmatrix}$$

são os dados do problema e devemos resolver a equação algébrica de Riccati:

$$A^T X A - X - A^T X G_1 (G_2 + G_1^T X G_1)^{-1} G_1^T X A + Q = 0$$

A matriz simplética, a matriz em Forma Real de Schur ordenada e a matriz de transformação utilizada para achar a solução da equação são apresentadas a seguir:

$$H_{symp} = \begin{bmatrix} 1.05130 & 0.00000 & 83.68837 & -12.23875 \\ 0.00000 & 1.10522 & -12.86638 & 9.38448 \\ 0.00526 & 0.00000 & 1.36964 & -0.06119 \\ 0.00000 & 0.02210 & -0.25733 & 1.09249 \end{bmatrix}$$

$$H_{schur} = \begin{bmatrix} 1.9696 & 0.2426 & 6.0438 & -85.3252 \\ 0.0000 & 1.4533 & -7.7243 & 5.4132 \\ 0.0000 & 0.0000 & 0.6881 & -0.0848 \\ 0.0000 & 0.0000 & 0.0000 & 0.5077 \end{bmatrix}$$

$$U = \begin{bmatrix} 0.9722 & -0.2339 & -0.0056 & 0.0094 \\ -0.2339 & -0.9718 & -0.0303 & -0.0040 \\ 0.0094 & -0.0056 & 0.2338 & -0.9722 \\ -0.0087 & -0.0497 & 0.9712 & 0.2338 \end{bmatrix}$$

Como solução final, obtém-se uma matriz X simétrica, definida não negativa:

$$X = \begin{bmatrix} 0.01045 & 0.00323 \\ 0.00323 & 0.05041 \end{bmatrix}$$

que coincide com a solução dada por Laub (1979) no seu artigo.

A seguir é apresentado outro exemplo onde a ordem da matriz simplética obtida é (50×50); desta vez, executa-se o algoritmo paralelo; os resultados são apresentados na Tabela 1, na qual é feita uma pequena comparação quanto ao tempo de execução de cada uma das partes do algoritmo, tanto sequencial como paralelo.

Tabela 1: Tempo de execução de uma matriz (50×50)

	SEQUENCIAL	PARALELO	
Actividade	1-proc	2-proc	3-proc
Cálculo da matriz de Hessenberg	59.0045	8.3596	10.3933
Cálculo da matriz em FRS	1.8914	0.9707	1.3932
Tempo de execução global	60.8959	9.3303	11.7865

6. CONCLUSÃO

Pode-se dizer como conclusão deste trabalho que os algoritmos implementados satisfazem os objetivos apresentados, ou seja obter uma solução simétrica, definida não negativa da *EARD*.

Com estes resultados aqui apresentados pode-se concluir que a paralelização surte melhores resultados quando o problema envolve cálculos com matrizes de ordem bem maiores, Tamariz (1999).

Em relação a comparação em tempo de execução dos algoritmos implementados em formas sequencial e paralela, podemos dizer que o processamento paralelo nem sempre foi melhor que o processamento sequencial quanto ao tempo de execução, pois no processamento paralelo existe um *overhead* de comunicação alto entre os processadores, o que implica que a solução seja mais demorada.

Em nossa implementação concluímos que a carga computacional de cada processador envolvido com o algoritmo foi reduzida em relação a utilização de um único processador. O tempo de execução do algoritmo pode ser minimizado a partir da utilização de uma máquina paralela, de forma tal que os tempos de comunicação entre os diferentes processadores participantes na execução da tarefa sejam irrelevantes, diferentemente da utilização de uma rede de estações de trabalho.

REFERÊNCIAS

Laub, A. J., 1979, A Schur Method for Solving Algebraic Riccati Equations, IEEE Transactions on Automatic Control, vol. AC-24, n. 6, December.

Arnold, W. F. & Laub, A. J., 1984, Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations, Proceedings of the IEEE, December, 72(12).

Gardiner, J. D. & Laub, A. J., 1991, Parallel Algorithms for Algebraic Riccati Equations, Int. J. Contr., 54:1317-1333.

Bertsekas,D. P. & Tsitsiklis, J. N., 1989, Parallel and Distributed Computation, Prentice-Hall International Editions.

Golub, G. H. & Van Loan, C. F., 1989, Matrix Computations , second ed., The Johns Hopkins University Press.

Eberlein, P. J., 1987, On the Schur Decomposition of a Matrix for Parallel Computation, IEEE Transactions on Computer, February, C-36(2).

Martin, R. S. & , Wilkinson, J.H., 1968, Similarity Reduction of a General Matrix to Hessenberg Form, Prepublished in Numer. Math. 12, pp. 349-368.

Vaughan, D. R., 1970, A Nonrecursive Algebraic Solution for the Discrete Riccati Equation, IEEE Transactions on Automatic Control, October, vol. AC-15, n. 5, pp. 597-599.

Aoki, M., 1990, State Space Modeling of Time Series, second ed., Springer-Verlag.

Jennings, A. & McKeown, J. J.,1993, Matrix Computation, second ed., John Wiley - Sons Ltd.

Tamariz, A.R., 1999, Uma Nova Proposta para Solução Computacional da Equação Algébrica de Riccati em Formas Sequencial e Paralela, Tese de Mestrado, Universidade Estadual de Campinas, UNICAMP.

SEQUENTIAL AND PARALLEL FORMS FOR THE SOLUTION OF THE ALGEBRAIC RICCATI EQUATION BY A MODIFIED-SCHUR ALGORITHM

Abstract: *This work proposes a new Parallel and Distributed algorithm for the solution of The Algebraic Riccati Equation. This algorithm presents a parallelization for a special implementation of the Schur Method (Laub,1979), where the sympletic matrix is modified through Stabilized Elementary Similarity Transformation. In the first module of the algorithm the sympletic matrix is transformed to the Hessenberg Upper Form (HUF). In the second module the Francis QR algorithm is applied to the HUF to obtain the Real Schur Form in a parallelized way. Our purpose is to obtain the unique symmetric, nonnegative definite solution of the Algebraic Riccati Equation.*

Keywords: *Algebraic Riccati Equations, Control Systems, Parallel and Distributed Computation, Model Identification.*