UM MÉTODO DE IDENTIFICAÇÃO RECURSIVA PARALELA

José Tarcisio Costa Filho, Celso Pascoli Bottura e Gilmar Barreto
DMCSI - FEE - UNICAMP
C.P. 6101 - CEP - 13.081-970 - Campinas - SP - Brasil
FAX: (0192) 391395 - E-mail: dmcsi@ccvax.unicamp.br

Resumo

Para as aplicações em sistemas de tempo real onde o controle e a identificação são separados em tarefas independentes e cooperantes e que podem ser executadas sobre computadores paralelos é possível aumentarmos o desempenho computacional das tarefas de identificação e controle, possibilitando a utilização eficiente de recursos de *hardware* e *software*. Neste trabalho, empregamos esquemas de fatorização matricial na estrutura algébrica de métodos de estimacão recursiva de parâmetros, visando a geração de algoritmos paralelos e vetoriais de alto nível. A idéia básica é a obtenção de decomposições matriciais baseadas nas transformações ortonormais de Householder para paralelizar algoritmos de identificação.

Abstract

For real time systems applications where control and identification are separated into independent and cooperating tasks, executable on parallel computers, we may improve the computacional performace of the identification and control tasks, enabling the efficient utilization of hardware and software resources. In this work, aiming to generate high level parallel and vetorial algorithms we use matrix factorization schemes on the algebraic structure of recursive parameters estimation. The basic idea is to obtain matrix decompositions based on Householder orthonormal transformations for identification algorithms parallelizations.

1 Introdução

O emprego de arquiteturas de processamento paralelo e vetorial para o controle em tempo real de sistemas dinâmicos estocásticos ainda apresenta muitas dificuldades de implantação prática, devido a variedade das aplicações e de situações de contexto. A estimação de parâmetros em tempo real é particularmente difícil de resolver uma vez que envolve grande quantidade s de dados e equacões mal condicionadas. Para as aplicações onde o controle e a identificação são separados em tarefas independentes e cooperantes e que podem ser executadas sobre computadores paralelos é possível aumentarmos o desempenho computacional das tarefas de identificação e controle, possibilitando a utilização eficiente de recursos de *hardware* e *software*.

Neste trabalho, empregamos esquemas de fatorização matricial na estrutura algébrica de métodos de estimacão recursiva de parâmetros, visando a geração de algoritmos paralelos e vetoriais de alto nível. A idéia básica é a obtencão de decomposicões matriciais baseadas nas transformacões ortonormais de Householder para paralelizar algoritmos de identificacão.

2 Metodologia de Identificação Recursiva

Os algoritmos de identificacão são bastante conhecidos na literatura [1, 2]. Basicamente, as equações recursivas dos algoritmos de identificação apresentam a mesma estrutura algébrica. Apresentamos, neste trabalho, a forma mais simples destes algoritmos, uma vez que as estratégias de paralelizacão propostas podem ser aplicadas sobre os vários algoritmos de identificacão.

A forma mais simples de problemas de míminos quadrados é baseada na minimizacão de funcões da forma :

$$S(\Theta) = \sum_{k=1}^{N} (y_k - x'_k\Theta)^2 \tag{1}$$

onde y_k é a observação escalar no tempo k e $\Theta \in R^p$ é o vetor de parâmetros a ser estimado. O algortimo de mínimos quadrados que estima Θ, baseado nas observações de $y_1, y_2, \ldots, y_N$, é dado por :

$$\Theta_N = (X'_N X_N)^{-1} X'_N Y \tag{2}$$

onde: $X_N = [x'_1, x'_2, \ldots, x'_N]$; $Y = [y_1, y_2, \ldots, y_N]$ Uma versão recursiva clássica do algoritmo de mínimos quadrados [1, 2, 3] pode ser calculada pelas seguintes equacões:

$$\hat{\Theta}_{N+1} = \hat{\Theta}_N + K_{N+1}(y_{N+1} - x'_{N+1}\hat{\Theta}_N) \tag{3}$$

onde K_N é uma matriz de ganho variante satisfazendo:

$$K_{N+1} = P_N x_{N+1}(1 + x'_{N+1} P_N x_{N+1}) \tag{4}$$

$$P_{N+1} = P_N - K_{N+1} x_{N+1} P_N \tag{5}$$

3 Identificação Paralela Via Fatorização Triangular

Um melhor desempenho computacional pode ser obtido se o programa organiza seus dados de forma a tornar eficiente o uso de memória virtual, memória principal, memória *cache* [4, 6] e registradores. Uma forma de se conseguir maior velocidade de processamento sobre arquiteturas paralelas e vetoriais é através do emprego de algoritmos-bloco [4, 5, 6]. Estes algoritmos realizam as operações matriciais na forma particionada, utilizando submatrizes em vez de escalares. Há, portanto, um aumento na localidade das operações (explorando-se a capacidade vetorial de cada processador) e uma redução da quantidade de dados transferidos da memória principal para a *cache* e da *cache* para os registradores (gerenciando-se eficientemente o compartilhamento da *cache* e dos registradores vetoriais). Na maioria das arquiteturas paralelas e vetoriais existentes, há um conjunto de subrotinas de álgebra linear básica que implementam eficientemente operações bloco-matriciais.

Para obtermos uma formulação bloco para processamento paralelo e vetorial do algoritmo de identificação recursiva apresentado na seção 2, empregamos uma sequência de transformações de Householder [1, 2]:

$$H = I - 2uu' \,,\ \| u \|_2 = 1 \tag{6}$$

onde u é um vetor de comprimento unitario, para formar a seguinte equação:

$$\hat{\Theta}_{N+1} = R_{N+1}^{-1}\eta_{N+1} \tag{7}$$

onde R_{N+1} é uma matriz triagular superior $p \times p$ e η_{N+1} é um vetor de dimensão p. A sequência de transformações de Householder deve ser escolhida de forma conveniente para que R_{N+1} e η_{N+1} possam ser computados como:

$$\begin{pmatrix} R_{N+1} \\ O \end{pmatrix} = \Psi \begin{pmatrix} R_N \\ x'_{N+1} \end{pmatrix} \tag{8}$$

onde Ψ_N é uma matriz ortonormal ($\Psi'\Psi = I$) dada pelo produto de transformações de Householder:

$$\Psi_N = H_p H_{p-1} \ldots H_1 \tag{9}$$

onde:

$$H_i = \begin{bmatrix} I_{i-1} & & 0 & & \\ & c_i & 0 & s_i \\ 0 & 0 & I_{p-i} & 0 \\ & s_i & 0 & c_i \end{bmatrix} ,\ i = 1, \ldots, p \tag{10}$$

onde: $c_i = (R_N)_{ii}/\lambda_i$, $s_i = (x_{N+1})_i/\lambda_i$, $\lambda_i = [(R_N)_{ii}^2 + (x_{N+1})_i]^{1/2}$, e

$$\begin{pmatrix} \eta_{N+1} \\ v_{N+1} \end{pmatrix} = \Psi \begin{pmatrix} \eta_N \\ y_{N+1} \end{pmatrix} \tag{11}$$

A aplicação de H à matriz X envolve somente simples operações matriciais do tipo matriz-vetor: $z \leftarrow X'u$ e a atualização: $X \leftarrow X - 2uz'$. Cada uma destas operações requer $O(n^2)$ operações de ponto de flutuante em $O(n^2)$ dados. Normalmente, as dimensões matriciais são muito maiores do que o número de processadores disponíveis, implicando na necessidade de se realizar multiplicação e adição na forma particionada. Uma análise do procedimento de paralelização e vetorização dessas operações sobre blocos matriciais é dada em [6].

Já a equação (7) pode ser resolvida em paralelo sem dificuldades, uma vez que R_{N+1} é uma matriz triangular superior e $\hat{\Theta}_{N+1}$ pode ser obtido por substituição *backward*:

$$\begin{bmatrix} R_{11} & \cdots & R_{1p} \\ & & \\ & & R_{pp} \end{bmatrix}_{N+1} \hat{\Theta}_{N+1} = \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_n \end{bmatrix}_{N+1} \tag{12}$$

O algoritmo de substituição *back* mais comum é (por simplicidade de notação, consideremos $\hat{\Theta}_{N+1} = [\theta_1, \ldots, \theta_p]'$:

$$\hat{\theta}_j = (\eta_j - R_{jj+1}\hat{\theta}_{j+1} - \cdots - R_{jp}\hat{\theta}_p)/R_{jj},\ j = p, \ldots, 1. \tag{13}$$

Se R_{N+1} é armazenado por linha, então (12) pode ser resolvida pelo produto interno com vetores de comprimento variando de 1 a $p-1$, juntamente com n divisões escalares. Ignorando as divisões, o grau médio de vetorização é proporcional a $p/2$.

4 Comentário Final

Neste trabalho, exploramos o paralelismo intrínsico das estruturas de equações algébricas de um algoritmo de identificação recursiva, através da obtenção de decomposições matriciais, utilizando, convenientemente, sequências de transformaccões de Householder. A estratégia de paralelismo e vetorização investigada pode ser facilmente estendida para outros processos de identificação recursiva. Também, estamos investigando um esquema de fatorização paralela para a forma matricial de Hessenberg [4].

References

[1] Goodwin, G. C. and Payne, R. L., Dynamic System Identification, Academic Press, 1977.

[2] Golub, G., Numerical Methods for Solving Linear Least Squares Problems, Numerische Mathematik 7, 206-216, 1965.

[3] Bertsekas, D. and Tsitsiklis, J., Parallel and Distributed Computations, Prentice-Hall, 1989.

[4] Ortega, J.M., Introduction to Parallel and Vector Solution of Linear Systems, Plenum Press, 1988.

[5] Costa Filho, J. Tarcisio e Celso P. Bottura, Parallel and Distributed Dynamic Games Computation on a Network of Multiple Workstations, 12th IFAC Workshop on Distributed Computer Control Systems, Beijing, China, 23 a 25/08/1992.

[6] Costa Filho, J. Tarcisio, Proposta para Computação Paralela e Distribuída Assíncrona de Estruturas Especiais de Jogos Dinâmicos, Tese de Doutorado, FEE-UNICAMP, 1992.