

Systolic 算法和结构求解代数 Riccati 方程

慕德俊 戴冠中

(西北工业大学自动控制系·西安, 710072)

摘要: 本文基于矩阵的符号函数法, 提出了一种 U-D 分解算法和脉动(Systolic) 结构有效地求解代数 Riccati 方程以及用固定大小的方形阵列解决大型问题的方法。

关键词: Systolic 阵列; U-D 分解; 符号函数法

1 引言

求解代数 Riccati 方程, 在线性二次型最优控制系统和最优滤波系统的分析和设计中起着关键作用, 求解这类方程的方法有多种, 其中包括解析法、符号函数法以及分解约简法等^[1], 由于串行求解方程的计算量很大, 对于大型 Riccati 方程, 有必要寻求一种并行的计算方法. 本文基于矩阵符号函数法提出了 U-D 分解方法和三种阵列结构(梯形、三角形、方形) 并行求解 Riccati 方程, 阵列结构是一种细粒度并行结构^[2], 它使得计算的效率及并行性较高, 这种算法和结构很适合于进行象符号函数法这样的迭代计算, 并且很容易推广到求解一般的 Riccati 方程和 Lyapunov 方程的计算. 为了满足不同大小方程的需要, 本文给出了在固定大小阵列上求解大型方程的方法, 对迭代周期及处理器的利用率作了简单的分析.

2 利用 U-D 分解法求解代数 Riccati 方程

对于一个 $n \times n$ 的矩阵 H , 其基本的符号函数算法^[1,3] 如下:

$$Z_0 = H, \quad Z_{k+1} = \frac{1}{2C}(Z_k + C^2 Z_k^{-1}).$$

其中

$$C = |\det Z_k|^{1/n}.$$

如果 H 在虚轴上没有特征值, 那么 Z_k 将收敛于 $\text{sgn}(H)$, 即 $Z_k \rightarrow \text{sgn}(H)$.

对于一个矩阵束为 $H = \lambda E_1 E_2$ 的符号函数算法^[4,5] 如下:

$$Y_0 = E_1^{-1} H E_2^{-1}, \quad Y_{k+1} = \frac{1}{2C}(Y_k + C^2 Y_k^{-1}).$$

其中

$$C = |\det E_1^{-1} H E_2^{-1}|^{1/n}.$$

当 E_1 和 E_2 是非奇异的, 并且 $E_1^{-1} H E_2^{-1}$ 在虚轴上没有特征值, 则 $Y_k \rightarrow \text{sgn}(E_1^{-1} H E_2^{-1})$, 如果定义 $Z_k = E_1 Y_k E_2$, 那么 $Z_0 = E_1 Y_0 E_2 = H$,

$$Z_{k+1} = E_1 Y_{k+1} E_2 = E_1 (Y_k + C^2 Y_k^{-1}) E_2 / (2C).$$

整理得

$$Z_{k+1} = \frac{1}{2C}(Z_k + C^2 M Z_k^{-1} M). \quad (1)$$

其中 $M = E_1 E_2$, 加速收敛的系数 $C = |\det(E_1^{-1} Z_k E_2^{-1})|^{1/n}$.

最终

$$Z_k \rightarrow E_1 [\operatorname{sgn}(E_1^{-1} H E_2^{-1})] E_2.$$

若连续的代数 Riccati 方程有如下形式:

$$A^T X E + E^T X A - E^T X B R^{-1} B^T X E + Q = 0, \quad (2)$$

式中 A, E, Q 都是实 $n \times n$ 阵, X 是 $n \times n$ 的未知矩阵, B 是实 $n \times m$ 阵, R 是 $m \times m$ 的非负定对称可逆阵, Q 是对称阵, E 是可逆阵. 若 Q 是非负定的, (A, B) 可隐, (Q, A) 可观, 则存在唯一的对称非负定稳定解. 若 Hamiltonian 束 $H - \lambda M = H - \lambda \begin{bmatrix} E & 0 \\ 0 & E^T \end{bmatrix}$ 在虚轴上没有特征值, 并且 (A, B) 可控, (2) 式存在唯一的非负定对称定解 X . 矩阵的符号函数法是求解 (2) 式的一种计算方法, 对于 (2) 式的 Hamiltonian 矩阵可表示为:

$$H_1 = \begin{bmatrix} A E^{-1} & -B R^{-1} B^T \\ -E^T Q E^{-1} & -E^{-T} A^T \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & E^{-T} \end{bmatrix} \begin{bmatrix} A & -B R^{-1} B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} E^{-1} & 0 \\ 0 & I \end{bmatrix} = E_1^{-1} H E_2^{-1}.$$

其中

$$H = \begin{bmatrix} A & -B R^{-1} B^T \\ -Q & -A^T \end{bmatrix}, E_1 = \begin{bmatrix} I & 0 \\ 0 & E^{-T} \end{bmatrix}, E_2 = \begin{bmatrix} E & 0 \\ 0 & I \end{bmatrix}, M = E_1 E_2 = \begin{bmatrix} E & 0 \\ 0 & E^T \end{bmatrix}.$$

利用符号函数法求 (2) 式的非负定解为:

$$\begin{bmatrix} S_{12} \\ S_{22} + E^T \end{bmatrix} X E = - \begin{bmatrix} S_{11} + E \\ S_{21} \end{bmatrix}. \quad (3)$$

式中

$$S = E_1 [\operatorname{sgn}(E_1^{-1} H E_2^{-1})] E_2 = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}.$$

S 为 (1) 式中 Z_k 的收敛值. 通过 (3) 式可求出 X , 但在最优控制中, 反馈矩阵 $K = -R^{-1} B^T X E$, 因此通常只须求出 $X E$ 即可.

设

$$J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}, \quad Z_0 = J H_1,$$

$$\text{则} \quad Z_0 = \begin{bmatrix} -E^T Q E^{-1} & -E^{-T} A^T \\ -A E^{-1} & B R^{-1} B^T \end{bmatrix}; \quad J M = - (J M)^T = \begin{bmatrix} 0 & E^T \\ -E & 0 \end{bmatrix};$$

$$Z_{k+1} = \frac{1}{2} [Z_k - (J M)^T Z_k^{-1} (J M)]. \quad (4)$$

Z_k 为对称矩阵且收敛于 $J S$, 可应用 U-D 方法将 Z_k 分解成 $Z_k = U_k D_k U_k^T$, 式中 U_k^T, D_k 分别是主对角线为 1 的上三角阵和对角矩阵, 若 Z_k 的所有主子式不为零时, D_k 为对角阵, 这时对 (4) 式进行 U-D 分解得

$$\begin{bmatrix} U_{k+1} & 0 \end{bmatrix} \begin{bmatrix} D_{k+1} & \\ * & \end{bmatrix} \begin{bmatrix} U_{k+1}^T \\ 0 \end{bmatrix} = \begin{bmatrix} U_k & (J M)^T U_k^{-T} \end{bmatrix} \begin{bmatrix} \frac{1}{2} D_k & \\ & -(2 D_k)^{-1} \end{bmatrix} \begin{bmatrix} U_k^T \\ U_k^{-1} J M \end{bmatrix}. \quad (5)$$

1 期

* 代表无关紧要项。

3 U-D 分解的 Systolic 算法及阵列实现

由以上分析可知,通过对(1)式的迭代计算,可求出(3)式中 Z_k 的收敛值 S ,在迭代的过程中可不考虑 C 的影响即 $C=1$,这样可用一个 $n \times (2n+1)$ 的三角阵列和 $2n \times 2n$ 的正方形阵列组成的梯形阵列进行 Faddeev 运算^[2],得到的 Z_{k+1} 再返回到阵列的输入端进行下次迭代计算. 经过 N 次迭代后得到 Z_k 的收敛值 S . 每次迭代需 $6n+1$ 个单位时间,即迭代周期为 $6n+1$ 个单位时间,每个单元进行 $4n$ 次操作运算,处理器的利用率约 70%.

通过图 1 所示的 $2n \times 2n$ 三角阵列可完成(5)式的 U-D 分解计算. 将对角阵 $(2D_k)^{-1}$ 和方阵 $U_k^{-1}JM$ 与阵列里的上三角阵 U_k^T 和对角阵列里的 $D_k/2$ 进行修正的 Givens 旋转计算,在阵列中产生 D_{k+1}, U_{k+1} .

为了进行下次迭代计算,还需求出 $D_{k+1}/2, -(2D_{k+1})^{-1}, U_{k+1}^{-1}(JM)$,其求解方法是将标量 2 和 $(JM)^T$ 阵输入阵列(如图 1 所示),分别与阵列里的 D_{k+1}, U_{k+1}^T 进行运算^[6],得到的 $-(2D_{k+1})^{-1}, U_{k+1}^{-1}JM$ 从阵列的左右两边输出, $D_{k+1}/2$ 保留在对角单元内, U_{k+1}^T 在阵列里保持不变. 各阶段中单元的功能见图 1.

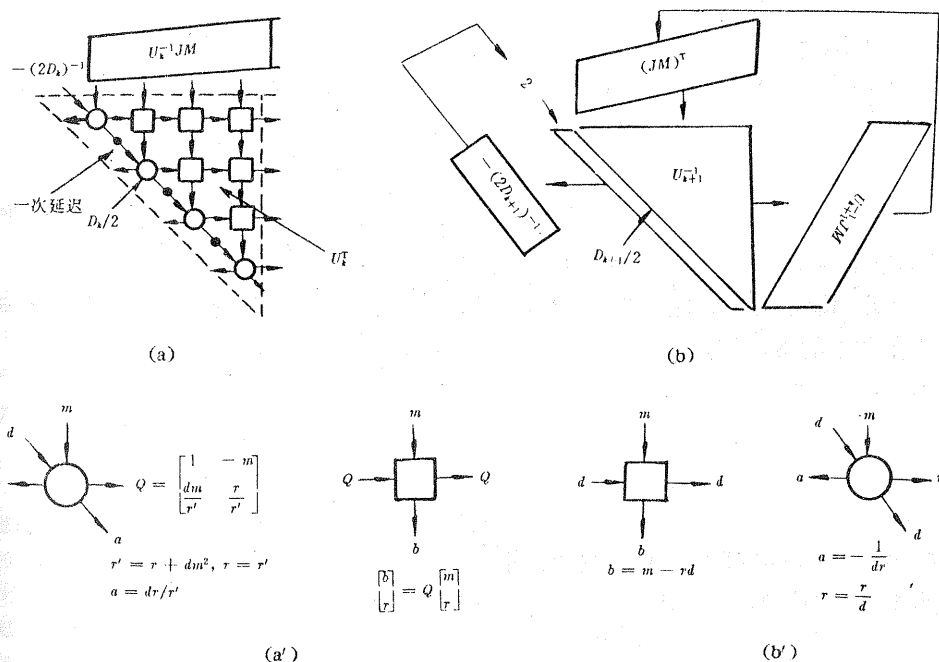


图 1 三角阵列实现 U-D 算法

由以上分析容易看出,每次迭代需 $4n$ 个单位时间,处理器的利用率接近 100%,因此 U-D 分解法可使数据流快速平稳地流动,处理单元的利用率高. 修正的 Givens 旋转法也避免平方计算.

若 Z_k 为一般对称阵时,则(5)式中的 D_k 为 $|x|$ 或 2×2 的块对角阵,这时实现(5)式 U-D 分解的阵列结构有待进一步研究.

4 固定大小阵列上求解方程

在实际计算中所遇到矩阵方程的大小往往是变化的,因此在固定大小阵列上求解大维数的 Riccati 方程是有意义的. 求解这种问题的方法较多,通常有模拟法和分解法,本文采用模拟方法,为了说明方便,将上述的梯形阵列分解成 7 块 n 维子阵列,用一个 n 维的实方阵列(见图 2)上逐个进行运算. 记图 2 主对角线右上的三角阵列为模块 A,整个方阵列为模块 B,线性阵列为模块 C;QR 分解的旋转参数阵及 Gauss 消去的乘数阵分别为 Q, T . 每次 Z_{k+1} 的迭代求解可分 5 步进行,方法如下:

将第 K 次的 Z_k 分块成 $Z_k = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$ 并令

$$F_k = F_{4 \times 4} = \begin{bmatrix} Z_k & M \\ M & Z_k \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} & E & 0 \\ Z_{21} & Z_{22} & 0 & E^T \\ E & 0 & Z_{11} & Z_{12} \\ 0 & E^T & Z_{21} & Z_{22} \end{bmatrix}$$

第 1 步(模块 A) QR 分解 $F_{11}(Z_{11})$, $F_{21}(Z_{21})$; Gauss 消去 $F_{31}(E)$; 分别存旋转参数, Gauss 消去的乘数在右边的缓冲器里.

第 2 步(模块 B) 计算

$$Q^{(1)} \begin{bmatrix} Z_{12} & E & 0 \\ Z_{12} & 0 & E^T \end{bmatrix} \rightarrow \begin{bmatrix} F_{12}^{(1)} & F_{13}^{(1)} & F_{14}^{(1)} \\ F_{22}^{(1)} & F_{23}^{(1)} & F_{24}^{(1)} \end{bmatrix},$$

$$T^{(1)} \begin{bmatrix} F_{12}^{(1)} & F_{13}^{(1)} & F_{14}^{(1)} \\ 0 & Z_{11} & Z_{12} \end{bmatrix} \rightarrow \begin{bmatrix} F_{12}^{(1)} & F_{13}^{(1)} & F_{14}^{(1)} \\ F_{32}^{(1)} & F_{33}^{(1)} & F_{34}^{(1)} \end{bmatrix}.$$

将 $F_{22}^{(1)}, F_{23}^{(1)}, F_{24}^{(1)}, F_{32}^{(1)}, F_{33}^{(1)}, F_{34}^{(1)}$ 存在下面的缓冲器里.

第 3 步(模块 A) QR 分解 $F_{22}^{(1)}$; Gauss 消去 $F_{32}^{(1)}, F_{42}(E^T)$; 分别存旋转参数, Gauss 消去的乘数在右边的缓冲器里.

第 4 步(模块 B) 计算

$$Q^{(2)} [F_{23}^{(1)} \quad F_{24}^{(1)}] \rightarrow [F_{23}^{(2)} \quad F_{24}^{(2)}], \quad T^{(2)} \begin{bmatrix} F_{23}^{(2)} & F_{24}^{(2)} \\ F_{33}^{(1)} & Z_{34}^{(1)} \\ Z_{21} & Z_{22} \end{bmatrix} \rightarrow \begin{bmatrix} F_{23}^{(2)} & F_{24}^{(2)} \\ F_{33}^{(2)} & F_{34}^{(2)} \\ F_{43}^{(1)} & F_{44}^{(1)} \end{bmatrix}.$$

将 $F_{33}^{(2)}, F_{34}^{(2)}, F_{43}^{(2)}, F_{44}^{(2)}$ 存在下面的缓冲器里.

第 5 步(模块 C) 计算

$$\frac{1}{2} \begin{bmatrix} F_{33}^{(2)} & F_{34}^{(2)} \\ F_{43}^{(2)} & F_{44}^{(2)} \end{bmatrix} \rightarrow \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} = Z_{k+1},$$

经 N 次迭代计算后得到 S , 这时将 E 输入线性阵列与 S_{11} 相加, 输出 $S_{11} + E$, 再将 $S_{12}, I_{n \times n} - (S_{11} + E), 0_{n \times n}$ 依次输入阵列, 进行 QR 分解和 Gaussian 消去得到 XE 在方阵列底部输出. 通过分析, 每次迭代的周期为 $21n$ 个单位时间, 处理器的利用率为 85% 以上, 此方法可推广到用任意大小阵列去求解大型 Riccati 方程的计算.

5 结 论

本文所提出的 U-D 分解算法和 Systolic 结构是有效地快速实现求解代数 Riccati 方程

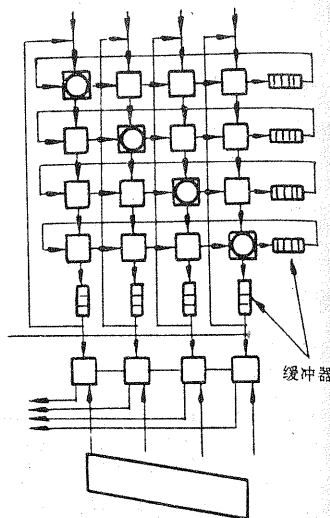


图 2 方阵阵列结构实现 Riccati 方程计算

的方法,这种并行方法很容易推广到求解离散情形及一般情形的 Riccati 方程和 Lyapunov 方程,本文所使用的算法不必进行特征值和特征向量的计算,且算法简单易于并行计算.应用 $n \times n$ 实方阵列实现方程计算的原理,可在任意大小的方阵列上求解大型 Riccati 及 Lyapunov 方程的计算.

参 考 文 献

- [1] 毛剑琴,姜长生等.控制系统的计算机辅助设计.北京:北京航空航天大学出版社,1988
- [2] 陈国良,陈凌.VLSI 计算理论与并行算法.安徽:中国科技大学出版社,1991
- [3] Roberts, J. D.. Linear Model Reduction and Solution of Algebraic Riccati Equation by Use of the Sign Function. Int. J, Control, 1980, 32(4):667—687
- [4] Bierman, G. J.. Computational Aspects of the Matrix Sign Function Solution to the ARE, Proc. of Conf. Decision and Control, 1984, 514—519
- [5] Gardiner, J. D.. A Generalization of the Matrix-Sign Function Solution for Algebraic Riccati Equations. Int. J, Control, 1986, 44(3):823—832
- [6] Mcwhirter, J. G. Algorithmic Engineering in Adaptive Signal Processor. IEE Pt F, 1992, 139(3):226—232

Systolic Algorithms and Architectures for Algebraic Riccati Equation

MU Dejun and DAI Guanzhong

(Department of Automatic Control, Northwestern Polytechnical University • Xi'an, 710072, PRC)

Abstract: This paper proposes an efficient U-D decomposed algorithm based on sign function and two architectures to solve the generalized algebraic Riccati equation. The method of solving large-scale problems in a size-independent square array is shown.

Key words: U-D decomposition; systolic array; Riccati equation

本文作者简介

慕德俊 1963 年生. 1983 年获机械工程学院导弹专业学士学位,此后在新疆某部从事导弹业务. 1990 和 1994 年分别获西北工业大学惯导专业硕士学位和自动控制理论及应用专业博士学位. 现在南京航空航天大学自动控制研究所做博士后研究. 主要研究领域是控制系统中的并行算法,低空突防中的地形跟随与地形回避等.

戴冠中 1937 年生. 教授. 自动控制理论及应用学科博士生导师. 现任西北工业大学校长. 目前的主要研究领域是大系统估计与控制理论,智能控制,控制系统中的并行处理理论,算法与并行仿真计算机,控制理论在石油和天然气勘探中的应用等等.