



并行自适应控制算法及双处理机实现

张志勇 王诗宓 方崇智

(清华大学自动化系 北京 100084)

康景利

(北京理工大学一系 100081)

摘 要

该文提出了一种新的并行处理自适应控制算法。该算法将自适应控制中辨识和控制算法分为运行时间相当的两大部分,在每个控制间隔内,两部分交换信息后分别同时计算,从而在任务级上实现了并行处理。文中讨论了并行算法流程及性能评价指标,并给出了在制导系统中的应用实例。试验表明,通过双机并行处理,自适应控制的处理速度有显著提高。

关键词: 自适应控制,并行处理,实时控制,共享存储,多机系统。

1 引言

随着计算机科学和技术的迅猛发展,并行处理越来越引起广泛的重视,它是提高现有计算机系统性能的有效途径。作为与计算机密切相关的实时控制领域,引入并行处理技术也引起极大的关注^[1]。

在复杂系统的自适应控制中,为了得到较好的控制效果,经常采用计算量比较大的算法。控制间隔是总有限的,这对计算机提出较高的要求。提高单个计算机运算速度将使成本大大增加,有时还是难以实现的,解决这个问题的有效途径是:对自适应控制采用并行处理算法和结构,以硬件上的多重配置换取速度的提高^[1]。

本文提出了自适应控制的一种并行处理新方法,基本思想是将目前应用颇多的基于递推最小二乘估计器的自校正控制分为两大部分在两个处理机上分别同时计算,以满足实时控制对计算速度的要求。与指令级并行结构或阵列式结构^[2]不同,这种并行处理是在任务级上分配和进行,其原理清晰,便于实现。

2 并行处理算法

考虑一般自适应算法的结构,一个自适应系统,主要是由参数估计器和控制器两部分组成。

设被控对象可用下列回归模型描述:

$$y(t) = \varphi(t-1)^T \theta(t-1) + \xi(t), \quad (1)$$

其中 $\theta(t-1)$ 是模型时变参数； $\varphi(t-1)$ 是包括输入 $u(\cdot)$ 和输出 $y(\cdot)$ 的回归向量； $\xi(t)$ 是外界干扰，它可以是白噪声或有界干扰； $y(t) \in R^1$ ， $\varphi(t-1) \in R^{n \times 1}$ ， $\theta(t-1) \in R^{n \times 1}$ 。

一般递推最小二乘参数估计器可写成如下形式^[3]：

$$\hat{\theta}(t) = \hat{\theta}(t-1) + G[P(t-2), \varphi(t-1), \eta(t-1), \beta(t-1)] \cdot \varepsilon(t), \quad (2)$$

其中 $\varepsilon(t)$ 是预测误差

$$\varepsilon(t) = y(t) - \varphi(t-1)^T \hat{\theta}(t-1), \quad (3)$$

$G(\cdot)$ 代表参数更新的增益向量函数， $P(t)$ 是协方差阵，按下式更新：

$$P(t-1) = F[P(t-2), \varphi(t-1), \eta(t), \beta(t)], \quad (4)$$

其中 $\eta(t)$ 和 $\beta(t)$ 是可变遗忘因子

$$\eta(t) = H[\varepsilon(t), P(t-2), \varphi(t-1)], \quad (5)$$

$$\beta(t) = B[\varepsilon(t), P(t-2), \varphi(t-1)]. \quad (6)$$

$\eta(t)$ 和 $\beta(t)$ 有多种形式^[4]，它们保持估计器对时变参数的跟踪能力。

控制器的形式为

$$u(t) = \mathcal{C}[u(t-1), \varphi(t-1), \hat{\theta}(t)], \quad (7)$$

其中 \mathcal{C} 包括了各种控制律，如最小方差、极点配置、广义预测律等。

按照传统的自适应算法，式(2)一(7)是依次执行的。根据任务均分的思想，将它排成图 1 所示的运行时间相当的两部分同时执行。

在图 1 中，在每个控制间隔，程序开始执行时，两部分需交换必要的信息，第 I 部分需要第 II 部分的 $P(t-2)$ ， $\eta(t-1)$ ， $\beta(t-1)$ 数据，第 II 部分需要第 I 部分的 $y(t)$ ， $\varphi(t-1)$ ， $\hat{\theta}(t-1)$ 数据。

将第 I 部分和第 II 部分分别由主处理器 CP1 和从主理器 CP2 处理。CP1 除了运行第 I 部分算法外，还要对整个控制系统进行时钟管理、中断管理、A/D 及 D/A 转换以及其它 I/O 管理。设 A/D、D/A 转换时间为 $t_{A/D}$ 、 $t_{D/A}$ ，其余时间为 t_1 。CP1 和 CP2 交换信息所需的通讯时间为 t_{com} 。

那么，自适应控制的传统串行算法在单处理机上每步运行时间为

$$t_s = t_1 + t_{A/D} + t_{D/A} + t_2 + t_3 + t_4 + t_5 + t_6 + t_7, \quad (8)$$

其中 t_2 — t_7 分别表示运行式(2)一(7)所需时间。

在并行处理情形下，CP1 上运行时间为

$$t_{CP1} = t_1 + t_{A/D} + t_{D/A} + t_3 + t_2 + t_7 + t_{com}, \quad (9)$$

CP2 上运行时间为

$$t_{CP2} = t_3 + t_5 + t_6 + t_4 + t_{com}. \quad (10)$$

一般情况下，对 CP1 和 CP2 上的算法任务合理分配，使之满足

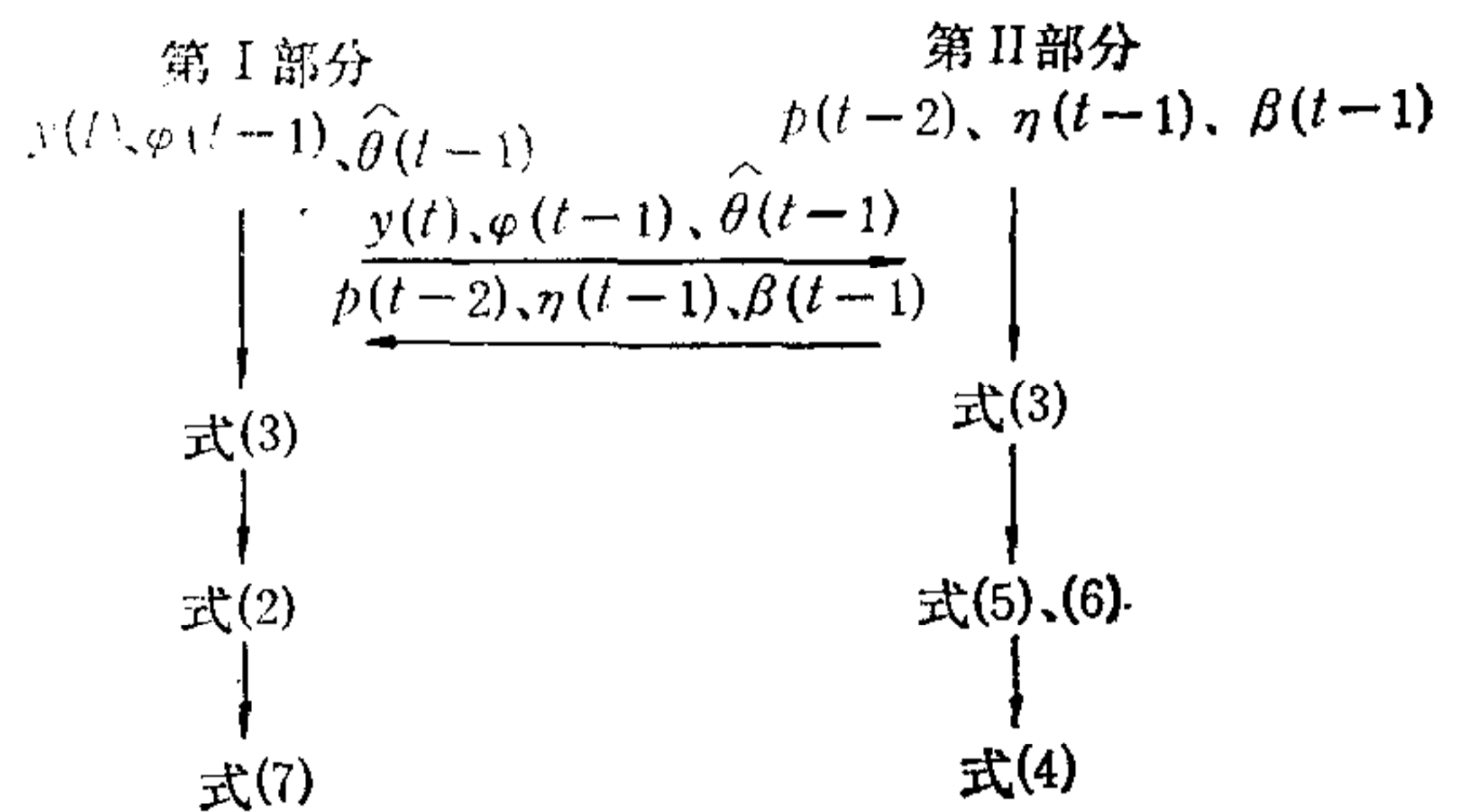


图 1 并行算法流程

$$t_{CP2} \approx t_{CP1}. \quad (11)$$

如果 t_{com} 很小,那么

$$t_{CP1} \approx \frac{1}{2} t_s. \quad (12)$$

从上面可以看出,式(4)的运算比较费时, t_s 内既包含矩阵加减,又包含矩阵乘运算。式(3)的结果 $\varepsilon(t)$ 在 CP1 和 CP2 中都要用到,如果通讯速度很快,也可以在 CP1 或 CP2 中省去一个式(3)的计算,直接通过另一个处理器将 $\varepsilon(t)$ 数据传送过来,将省出的时间用于别的运算。总之,通过逐步调整,达到式(11)或(12)的要求是完全可能的,这样便可将自适应控制的运行速度提高近一倍。

3 算法评价指标

为了评价并行算法的优劣,可以用加速比 S_p 、效率 E_p 、冗余度 R_p 、速效积 F_p 等指标来衡量^[5]。

(1) 加速比 S_p

$$S_p = t_s / t_p, \quad (13)$$

其中 t_s 、 t_p 分别为每个控制间隔内串行算法和并行算法所需的最大运行时间。“加速比”用以度量并行算法对计算时间的改进程度。在理想情况下,一个并行算法使用 p 个处理器应比原来串行算法使用一个处理器快 p 倍,但实际上不太容易做到。一般情况下,有

$$1 < S_p \leq p. \quad (14)$$

对于自适应控制的双机并行算法,则 $1 < S_p \leq 2$ 。

(2) 效率 E_p

$$E_p = (S_p / p) \cdot 100\%. \quad (15)$$

效率可以用来度量处理能力发挥的程度。

(3) 冗余度 R_p

$$R_p = A_p / A_s. \quad (16)$$

其中 A_p 和 A_s 分别为并行算法和串行算法的总运算次数。“冗余度”表示并行算法的运算次数比串行算法的运算次数的增加程度。一般有 $R_p \geq 1$ 。

(4) 速效积 F_p

$$F_p = S_p \cdot E_p. \quad (17)$$

速效积等于加速比与效率的乘积,它综合反映了并行算法的性能。一个好的并行算法应尽可能使 F_p 之值为最大。

4 基于共享存储的 8098 双机系统设计及应用实例

Intel 8098 单片机是 Intel 公司推出的 MCS-96 系列中的成员^[6],其内部采用16位数据总线,232字片的片内寄存器具有累加器功能,运算速度可达每秒100万次。它将四路 A/D 转换、采样/保持器、高速输入/输出口(HSI/HSO)、PWM(可作为 D/A 输

出)等集中于一块芯片,提高了抗干扰能力和可靠性,而且价格低廉,便于实时控制场合应用。

用两个 8098 单片机组成双机并行处理自适应控制系统,关键是要解决两个单片机之间的数据通讯问题。为了提高交换信息速度,这里不采用串行通讯方式,而是基于共享存储技术,让两个单片机都可以向双口 RAM 存取数据,如图 2 所示。

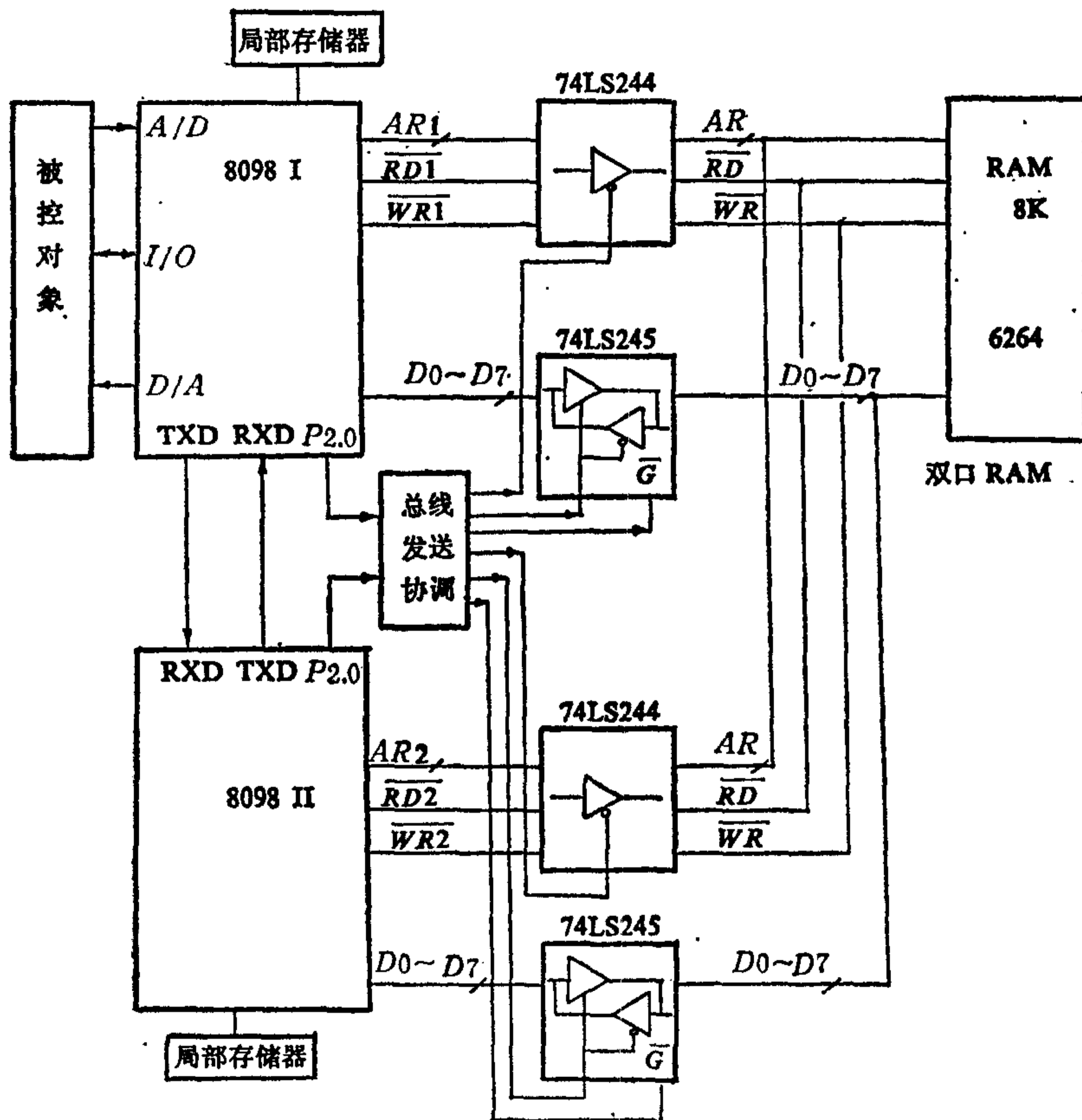


图 2 8098 双机并行处理系统

在图 2 中,地址总线 AR_1 、 AR_2 和读写控制线 $\overline{RD_1}$ 、 $\overline{RD_2}$ 、 $\overline{WR_1}$ 、 $\overline{WR_2}$ 由单向总线驱动器 74LS244 接往双口 RAM,数据总线 D_0-D_7 由双向总线发送器 74LS245 接往双口 RAM。至于何时双口 RAM 何时由主机 8098 I 读写、何时由从机 8098 II 读写,则由总线发送协调器根据自适应算法的进行程度来决定。并行系统的软件采用 PL/M 高级语言开发调试。

飞行器是一复杂被控对象,其制导系统有快速性和精确性要求。针对某战术导弹,其俯仰通道采用如下低阶等效模型描述^[7]:

$$y(t) = a_1(t)y(t-2) + a_2(t)y(t-3) + b_0(t)u(t-2) + b_1(t)u(t-3), \quad (18)$$

其中 $y(t)$ 、 $u(t)$ 分别是输出、输入量, $a_1(t)$ 、 $a_2(t)$ 、 $b_0(t)$ 、 $b_1(t)$ 是时变参数。

采用上述并行自适应算法和 8098 双机系统的实时仿真试验表明, 制导算法的最大运行时间可由串行情形下的 35.08 ms 减小到双机并行情形下的 17.92 ms, 其中双机数据通讯时间 t_{com} 仅占 0.21 ms, 从而满足了控制间隔 20 ms 的要求。加速比 $S_p = 35.08/17.92 = 1.958$, 效率 $E_p = 97.9\%$, 速效积 $F_p = 1.92$ 。整个并行系统提高处理速度近一倍, 有较高的性能价格比。这项研究已通过部级鉴定。

5 结束语

本文提出的并行处理自适应控制算法, 提高了自适应控制的运行速度, 并且在一个基于存储共享的 8098 双机系统上得到了验证。该算法通用性强, 原理清晰明瞭, 具有工程应用价值。

致谢 本项研究得到杨嘉墀、屠善澄、高为炳、李衍达、郑应平、涂序彦等教授的鼓励与支持, 龚云同学参与了实验工作, 在此一并表示感谢。

参 考 文 献

- [1] Irwin G W. Parallel Processing for Real-time Control. Proc. IEE Pt. D, 1991, 138 (4): 177—178.
- [2] Gaston F M F, Irwin G W. VLSI Architectures for Square Root Covariance Kalman Filtering. *J. VLSI Signal Process*, 1990, 2(1): 37—49.
- [3] Goodwin G C, Sin K S. Adaptive Filtering, Prediction and Control, Prentice-Hall. Inc. 1984.
- [4] Zhang Zhiyong, Wang Shifu, Fang Chongzhi, Kang Jingli. Unified Framework of Adaptive Forgetting Identification Algorithms for Time-Varying Systems, Proceedings of 1993 IEEE Region 10 Int. Conference on Computers, Communication and Automation, 4: 424—427.
- [5] Eager D L, Zamorjan J, Lazowska E D. Speed-up versus Efficiency in Parallel Systems. *IEEE Computers*, 1989, 38(3): 408—422.
- [6] Intel Corporation. 16-bit Embedded Microcontroller Handbook. 1989.
- [7] 张志勇, 康景利, 王诗宓, 方崇智. 战术导弹的鲁棒自校正制导及单片机实现. 宇航学报, 1992, (3): 13—18.

A PARALLEL ADAPTIVE CONTROL ALGORITHM AND ITS REALIZATION IN DOUBLE PROCESSORS

ZHANG ZHIYONG WANG SHIFU FANG CHONGZHI

(Department of Automation, Tsinghua University Beijing 100084)

KANG JINGLI

(First Department of Beijing Institute of Technology Beijing 100081)

ABSTRACT

In this paper, a new parallel adaptive control algorithm is presented. In this algorithm, the identification and control algorithms are separated in two parts with almost equal length for execution. In each control period, the two parts separately and concurrently compute after exchanging related information, so that the parallel processing is realized at task level. The flow process of algorithm and index of performance evaluating are also discussed.

An example in guidance systems is given. It is shown that the processing speed of adaptive control have been enhanced through parallel processing in double processors.

Key words: adaptive control, parallel processing, real-time control, common memory, multi-processors systems.