

动态大系统递阶实时控制的新算法及其实现

陈志新* 李人厚

(西安交通大学系统工程研究所)

摘要 本文采用了一种新的适用于实时在线控制的递阶算法,并在分布式计算机控制系统上予以实现。应用这种算法,可减少两级之间通讯的信息量,使控制级任务减轻,协调级的计算简单,收敛速度快,局部任务的选择自由度大。本文所设计的分布式计算机控制系统具有通讯可靠,易于扩充、维护等优点。实验证明,用该系统实现实时动态大系统的递阶控制有较理想的结果。

关键词: 大系统, 递阶控制, 算法, 分解, 协调, 局域网络

1 引言

动态递阶控制系统基本上采用目标协调和关联预测二种方法,它们都基于拉格朗日对偶原理,把问题等效为对偶函数对拉格朗日乘子向量求极大问题,协调量一般是取拉格朗日乘子向量。这两种方法在求解对偶函数的极值时都要遇到解两点边值问题,即要在第一级(控制级)解Riccati方程。解题(包括协调)时间较长,不适合在线控制。因此,寻求一种适于实时控制的算法是进行动态大系统控制的关键问题之一。

动态大系统递阶控制在工程上的实现已愈来愈受到人们的重视,并且已有了很多成功的实例。如: Tamura 延时算法应用于交通控制,关联预测法在河流污染控制的应用。这些研究的最大不足之处是仅基于数学模型的仿真,与实际在线控制还有一段距离。近些年来,一些研究者们开始用多微机系统来实现大系统的递阶控制,Markos Papagorgion 和 Guenther Schmidt 采用多微机把关联预测法应用于高速公路交通控制问题^[5]。Titli 等人用多微机解决高峰交通控制问题也采用了递阶优化方法。国内^[2,1]也对这方面的课题进行了深入的研究,并得到了较好的结果,但由于条件所限,仍存在一些问题,如:通讯的速度、可靠性、通用性等。

本文采用了一种新的适于实时控制的递阶算法^[4],并用分布式计算机控制系统实现这种动态递阶控制新算法。本文第二部分介绍实

时递阶控制算法,第三、四部分分别介绍实现此算法的硬件与软件,最后给出实验结果及结论。

2 适于实时控制的递阶算法

对于如下二次目标函数

$$\min J = \frac{1}{2} \sum_{k=0}^K x_{k+1}^T Q x_{k+1} + u_k^T R u_k$$

$$\text{s.t. } x_{k+1} = f(x_k, u_k)$$

的最优控制问题。

我们首先把系统分解为 N 个子系统。设系统是线性的,且具有关联作用。对于第 i 个子系统

$$x_{i,k+1} = A_{ii} x_{i,k} + B_{ii} u_{i,k} + \sum_{j \neq i}^N A_{ij} x_{j,k}$$

$x_{i,k}$, $u_{i,k}$ 分别是该子系统第 k 时刻的状态和控制向量。我们对目标函数进行如下分解:

$$J_{i,k} = \frac{1}{2} (x_{i,k+1}^T Q_i x_{i,k+1} + u_{i,k}^T R_i u_{i,k})$$

$$\rho_{i,k} = \bar{J}_{i,k} = J - J_{i,k} = \frac{1}{2} \sum_{j=1}^N \left(\sum_{l=0}^K x_{j,l+1}^T Q_j x_{j,l+1} + u_{j,l}^T R_j u_{j,l} \right) - J_{i,k}$$

由求解最优条件 $\frac{\partial \bar{J}}{\partial u_{i,k}} = 0$

$$\text{即 } \frac{\partial \bar{J}_{i,k}}{\partial u_{i,k}} + \frac{\partial J_{i,k}}{\partial u_{i,k}} = 0 \quad \text{可以得到}$$

* 现在太原重机学院任教

收本到文的时间是1988年4月25日。

$$u_{ik} = -R_i^{-1} \left[\frac{1}{2} B_i^T Q_i x_{ik} + \rho_{ik} \right] = \Lambda_i x_{ik} + \theta_{ik}$$

这里 u_{ik} 由两部分组成, 前项仅与该子系统有关, 是局部反馈控制; 后项是由协调器给出的开环控制量, 它计入了其它子系统对第 i 子系统的贡献, 与系统的所有状态有关。协调向量由以下式子计算:

$$\rho_{ik} = \frac{\partial \bar{J}_{ik}}{\partial u_{ik}} = B_i^T W_k - \frac{1}{2} B_i^T Q_i x_{ik}$$

$$W_k = Q x_{k+1} + \tilde{\Lambda}^T R u_{k+1} \tilde{\mathcal{A}}_i^T W_{k+1}; \quad W_K = 0$$

$$\tilde{\mathcal{A}}_{ii} = A_{ii} + B_{ii} \Lambda_i \quad \tilde{\mathcal{A}}_{ij} = A_{ij}, \quad i \neq j$$

$$\tilde{\Lambda} = \text{diag}(\Lambda_i); \quad B_i^T = [0, 0, \dots, B_i^T, \dots, 0]$$

计算步骤

初始条件 $\rho_{ik}^0 = 0; \quad x_{i0}^0 = x_{i0} \quad k = 1, 2, \dots, K, \quad i = 1, 2, \dots, N$

第一步 计算 $u_{ik}^0 = U(x_{ik}^0, \rho_{ik}^0)$

第二步 由 $x_{i,k+1}^0 = f(x_{ik}^0, u_{ik}^0)$ 计算 $x_{i,k+1}^0; \quad i = 1, 2, \dots, N, \quad k = 1, 2, \dots, K$, 重复第一、二步骤, 得到 x_{ik}^0, u_{ik}^0 , 即得到 x_i^0 和 u_i^0 的轨迹。

第三步 计算 $W_{k-1}^0 = W(W_k^0, x_{ik}^0, u_{ik}^0); \quad W_K = 0; \quad k = K, K-1, \dots, 1$ 。

第四步 计算 $\rho_{ik}^1 = \rho(W_k^0, x_{ik}^0) \quad i = 1, 2, \dots, N \quad k = 1, \dots, K$

第五步 修正 $\rho_{ik}^{j+1} = (I - Q) \rho_{ik}^j + Q \rho_{ik}^1$, 若 $\|\rho_{ik}^{j+1} - \rho_{ik}^j\| \geq ER$ 返回第一步, 否则将 ρ_{ik}^{j+1} 送到各局部控制器。

由上得出递阶算法结构如图 1 所示。协调

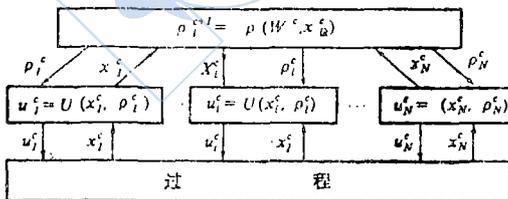


图 1 新的递阶算法结构

量 ρ_i 是协调中起关键作用的变量, 当系统受到扰动时, 状态要偏离预定的值, 这时协调量须

以该时刻的状态为初值, 重新修正协调向量 ρ_{ik} 。系统通过各各子系统的局部反馈和重新计算出来的 ρ_{ik} 所产生的局部开环控制使系统回到预定的状态。在协调器计算 ρ_{ik} 时, 局部控制器是按局部反馈控制规律来对系统进行控制的。因此, 对协调量 ρ_{ik} 的计算有两点要求:

(1) ρ_i 的收敛速度愈快愈好。这是因为 ρ_i 是一个与状态初值有关的开环量, 即: $\rho_i = \rho_i(x_0)$, 如果计算出 ρ_i 的时间 ($\Delta t = t_1 - t_0$) 愈短, 则把 ρ_i 施加于系统时的状态 x_1 与状态 x_0 就愈接近, $\rho_i(x_1)$ 与 $\rho_i(x_0)$ 也就愈接近, 这时 $\rho_i(x_1)$ 可以近似等于 $\rho_i(x_0)$ 。施加 ρ_i 也就愈“实时”。

(2) 计算出的 ρ_i 愈小愈好。 ρ_i 愈小, 说明其它子系统对 i 子系统的影响愈小, 当开环量施加于系统时, 状态变化才愈平稳。

提高计算 ρ_i 的速度, 应从两方面着手。

(1) 适当选取矩阵 Q , 会使 ρ_i 收敛较快。

(2) 适当选取采样周期。周期愈小, 则在动态过程中计算 ρ_i 的点数愈多, 速度则慢。当然, 采样周期也不能太大。

减小 ρ_i 应从分解子系统入手, 应尽量使分解后的子系统关联作用小。即各子系统之间为弱耦合。这样, 当某个子系统受到扰动时, 对其它子系统的影响小。

3 动态递阶控制系统的硬件结构与实现

用多微机实现动态递阶优化控制, 在文(2)中已实现。但由于条件所限, 还存在一些问题, 如: 通讯速率、可靠性、通用性等。此外, 在算法上, 很多计算是离线进行的。计算机局域网的出现, 为分布计算机控制系统实现动态大系统递阶控制提供了有力工具。

在局域网络上建立一个递阶控制系统, 硬件上是很方便的。选用三台微型机联入网中就可以成为如图 2 所示的结构。

各部件的配置和功能:

协调器 由装有 8087 协处理器的 IBM-PC 微机组成。完成对系统模型的分解、离散化、计算、对控制器的管理与协调。

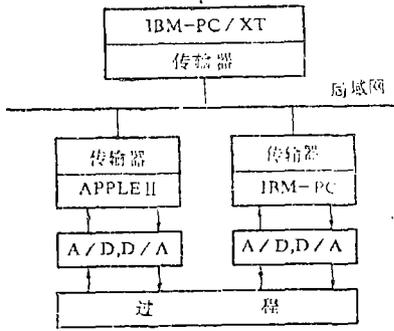


图 2 系统结构图

局部控制器 由一台 Apple II 和 IBM-PC 微机组成。主要功能是定时采集系统的状态信息，向协调器报告，同时还要接收协调器发来的协调参数，计算控制向量，对系统进行在线控制。

传输器 这是微机在通讯网中能够进行通讯的硬件核心。它执行 ISO 的 OSI 从物理层到传输层的协议。其作用是相当于网络中的一台前端处理机，因而可大大减轻主机的负担。

A/D, D/A 转换器 这是控制与被控对象信息交换的重要通道。A/D 转换器由控制器定时采集子系统的状态。控制器运算得到的控制向量通过 D/A 转换器施加给被控对象。

选择和设计这种结构具有一定的典型性、通用性。局域网是用 OMNINET 网改造的，运用点对点通讯传输率为 1Mbit/s。

4 系统软件的设计与实现

为了使系统管理方便、操作简单、实时性强，本系统软件用高级程序语言进行管理、运算。通讯处理、A/D、D/A 转换采用汇编语言。系统运行表明，高级语言与汇编语言混合编程效果较好。

本系统软件为协调器系统软件、控制器系统软件以及这两者之间通讯的基本通讯软件。

4.1 协调器的系统软件

协调器的系统软件主要由以下几部分组成。

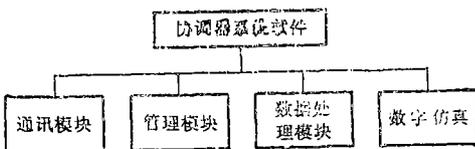


图 3 协调器系统软件组成框图

通讯模块 分接收和发送两部分。发送时，将发送的数据按信包格式装入发送缓存区，向控制器发送。接收时，它实际上是一个中断接收处理程序，接收控制器随时发来的信息包，进行分析、处理。

管理模块 完成对系统的初始化、数据管理、存储、打印、显示，为控制器分配任务，以及使它们同步工作。

数据处理模块 对用户输入的源数据进行分析处理，使连续的微分方程离散化，建立各子系统的差分方程，计算、修正协调量等。

数字仿真模块 这是一个独立模块。执行该模块，可以对系统进行数字仿真研究。

4.2 控制器系统软件

与协调器相对应，控制器的系统软件由以下几部分组成。



图 4 控制器系统软件组成框图

数据采集与输出控制模块 这部分是在定时器控制下进行工作的。每当采样周期到时，定时器产生一个中断信号，启动该模块运行。控制器是按通道的顺序对状态采样、计算、然后输出进行控制的。

通讯模块 与协调器发送接收对应，所不同的是 Apple II 机采用查询式接收。

运算模块 由于对控制器的实时性要求较高，运算量应尽量减少，在每个采集周期内，控制器要根据采集到的状态，重新计算控制 u 。

定时 定时的目的是使各控制器能周期地对其所对应的子系统采样。本系统的定时，在 IBM-PC 机上是重新对时钟中断 (08H 中断) 进行设置。在 Apple II 机上采用了一个时钟卡定时。用户可以从键盘输入采样周期对定时器进行预置。定时器开始工作是由协调器来启动的，以保证各定时器能同时工作。

4.3 基本通讯软件的设计与实现

针对 Apple 和 PC 机的传输器，我们分别设

计了与其相对应的基本通讯软件，以实现系统中各机之间的通信。

在设计基本通讯软件时，为保证系统通讯的实时性，采用了如下措施：（1）协调器与控制器采用点对点直接通讯方式。（2）协调器采用广播式发送命令，使控制器同步工作，同时对各自控制的子系统进行控制。（3）设置了各控制器的优先级别，任务重的优先级别高。（4）基本通讯程序采用汇编语言编制，缩短了系统的响应时间。

5 实验结果

例 考虑以下的互连系统

$$\dot{x} = \begin{bmatrix} -1. & 0.5 & 0.1 & 0.5 & 0. \\ 0.1 & -2. & -0.5 & 0.2 & -0.1 \\ 0.2 & 0. & -5. & 0.5 & -1. \\ 0.1 & -0.2 & 0. & -2. & 0. \\ 0.4 & 0.1 & -0.5 & 0. & -4. \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} u$$

特性指标函数为：

$$J = \sum_{i=1}^2 \frac{1}{2} [x_i^T(2)Q_i x_i(2)] + \int_0^2 [x_i^T(t)Q_i x_i(t) + u_i^T(t)R_i u_i(t)] dt$$

$$Q = \text{diag}(2, 1, 1, 1, 1) \quad R = I_2$$

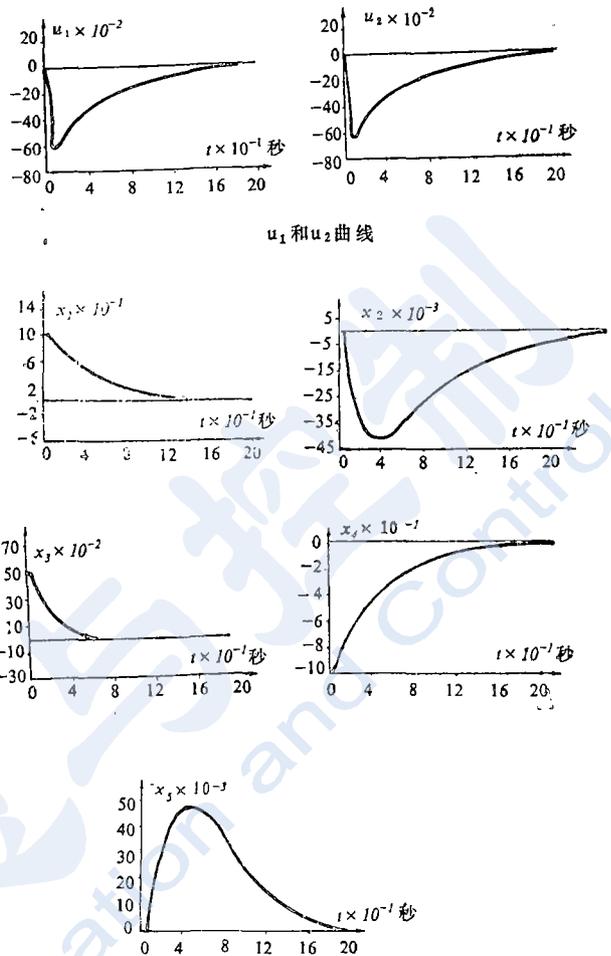
初始条件 $x^T(0) = (1, 0, 0.5, -1, 0)$

求解最优控制。

表 1 实验结果

| Ω | J | 迭代次数 | Δt (秒) |
|----------------|----------|------|----------------|
| diag(1, 1) | .4210572 | 23 | .014 |
| diag(0.9, 0.9) | .4206811 | 6 | .014 |
| diag(0.5, 0.5) | .4197927 | 3 | .014 |
| diag(0.3, 0.3) | .4205976 | 4 | .014 |
| diag(0.1, 0.1) | .4702208 | 1 | .014 |

由表可看出， Ω 的选取对收敛速度有影响。一般来讲， Ω 越大，收敛慢，但 J 精度高； Ω 越小，收敛快，J 精度差。该例的控制曲线如图 5 所示。



状态 $x_1 - x_5$ 曲线

图 5 模拟实验控制与状态曲线

6 结论

通过以上设计与实验，可以得出如下结论：

（1）所采用的算法使得两级之间的通讯量减少，控制级任务减轻，协调级运算简单，收敛快。算法对稳态大系统也可适用。

（2）在分布计算机控制系统上设计递阶控制系统，具有连接简单，通讯方便可靠，对机型选择自由度大，组件配置方便，通用性好，易维护管理，对用户透明性好等优点。

（3）实验虽然以模拟装置为对象，但反映了实际对象的特征。结果表明，用分布式计算机控制系统实现动态大系统递阶控制是可行的，在工业控制系统中具有一定的应用价值。

参 考 文 献

- 1 李人厚, 邵福庆. 大系统的递阶与分散控制. 西安, 西安交通大学出版社, 1986
- 2 贺显耀, 李人厚. 多微机系统实现动态大系统递阶优化控制. 信息与控制, 1985, 14(5): 1—7
- 3 张公忠等. 微型计算机局域网络 OMNINET 文集. 北京, 清华大学出版社, 1987
- 4 Drouin M, Bertrand P. A New Coordination Structure for on-line Control of Complex Processes. Large Scale Systems, 1982, 147—157
- 6 Markos Papageorgiou, Guenther Schmidt. Implementation of a Hierarchical Optimization Algorithm on Multimicrocomputer System. IEEE Trans, Syst, Man, Cybern, 1983, smc-13(1), 11—18
- 6 Drouin M, Abon-Kandil H, Bertrand P. On a Practical Approach to Real-time Control. In, IFAC Large Scale Systems, Warsaw Poland, 1983, 87—91
- 7 Mesarovic M D, Macko D, Takahara Y. Theory of Hierarchical Multilevel Systems. New York, Academic Press, 1970
- 8 Singh M G. Dynamical Hierarchical Control. North-Holland, 1980
- 9 Singh M G, Titli A. Systems, Decomposition, Optimization and Control. Oxford, Pergamon Press, 1978
- 10 Jamshidi M. Large Scale Systems, Modeling and Control. North-Holland, 1983

ALGORITHM AND IMPLEMENTATION OF HIERARCHICAL CONTROL OF LARGE SCALE DYNAMIC SYSTEMS

CHEN Zhixin LI Renhou

(Systems Engng Institute, Xi'an Jiaotong Univ)

ABSTRACT

A new decomposition—coordination algorithm, which is very efficient for on-line hierarchical control of large scale dynamic system, is implemented on a distributed computer control system. In this implementation the amount of information exchange between the coordinator and the local agents is reduced. The computation in each local agent is also simplified. The convergence of the coordination strategy is very fast and the flexibility of allocating the computation tasks among local agents in the lower level is greatly increased. Through using local area network, the connection among microcomputers is more efficient and reliable. The advantages of the system are generality, extensibility and maintainability. Carefully designed experiments show that distributed computer control systems is a desirable implementation of the hierarchical control for large scale dynamic systems.

Keywords: large scale systems, hierarchical control, algorithm, decomposition coordination, local area network

欢迎订阅《自动化博览》

《自动化》杂志从1989年第一期起更名为《自动化博览》。

《自动化博览》是中国自动化学会主办的国内第一份自动化专业的科普期刊, 文章图文并茂, 深入浅出地向广大读者介绍自动化科学技术领域的基础知识和实用技术, 广泛报道国内外自动化科技领域的新闻、动态, 及时宣传自动化科技在各行各业中的应用及取得的新成就, 为我国的四个现代化建设服务。

《自动化博览》为16开本, 彩色封面, 32页, 双月15日出版, 每期定价 0.50元, 全国邮局发行, 刊号: 18—79。

临时增订或补订, 请直接与本社联系, 地址: 北京2728信箱《自动化博览》杂志社。