

预测控制滚动优化的时间分解方法¹⁾

杨健 席裕庚 张钟俊

(上海交通大学自动化研究所 200030)

摘 要

基于大系统分解协调思想,针对预测控制系统,提出了一种带有并行结构的时间分解算法,以提高滚动优化在线计算效率.仿真结果表明了该算法的有效性.

关键词: 预测控制,滚动优化,分解协调,并行处理.

1 引言

作为一种有较强工业应用背景的启发式控制方法,预测控制具有易于建模、控制性能较好、鲁棒性强、逻辑结构简单等特点,因而在工业过程中获得了越来越广泛的应用.作为实时控制算法,其在线计算的效率是很重要的,但除了一些简单情况,预测控制的在线计算任务是比较大的,尤其是对于复杂的对象和性能指标以及带有约束条件的情形更是如此.通常情况下,分解协调方法是处理大系统问题的有效方法,它通过对系统进行递阶分解,以达到降低所处理问题规模的目的.对于空间维数较高的系统,文献[1]提出了预测控制递阶方法,对系统进行空间分解.但是当预测控制采用长时段的优化区间时,即使变量数目不多,离散化的优化时间也会导致大维数问题的产生.针对这一情况,本文基于大系统方法中的分解协调思想,在预测控制系统中,将原问题按时间方向分解,提出一种递阶分解协调算法,以提高在线优化的效率,并给出了仿真研究的结果.

2 预测控制滚动优化问题的一般描述

预测控制方法包括模型预测、反馈校正以及一个在线滚动进行的、有限长度区间上的最优控制.在每个采样时刻,它对该时刻起的一个有限区间进行优化;在下一个时刻到来时,将优化区间也向前滚动一个时刻,按照与前面相同的方式进行优化控制.因此预测控制的优化被称为滚动优化,是在线反复进行的.

对于确定性系统,若设当前时刻为 i ,优化区间长度为 p ,状态变量和控制变量分别为 $x(i+k)$, $u(i+k-1)$ ($k=1,2,\dots,p$),则在 k 时刻的优化控制任务为

1) 国家自然科学基金资助课题.
本文于1993年5月24日收到

$$\text{性能指标 } \min J = J_0[\mathbf{u}(i)] + \sum_{k=1}^{p-1} J_k[\mathbf{x}(i+k), \mathbf{u}(i+k)] + J_p[\mathbf{x}(i+p)],$$

$$\text{系统模型 } \mathbf{x}(i+k) = \mathbf{f}[\mathbf{x}(i+k-1), \mathbf{u}(i+k-1)], \quad (k=1, 2, \dots, p),$$

$$\mathbf{x}(i) = \mathbf{x}_{i0},$$

$$\text{约束条件 } \mathbf{g}[\mathbf{x}(i+k), \mathbf{u}(i+k)] \geq 0. \quad (k=0, 1, \dots, p)$$

在某些简化的情况下,上述问题可以获得解析解。而对于一般的情况,它只能作为规划问题来处理,这时系统模型和不等式约束一起作为约束条件。处理方法包括一些常用的规划方法,如罚函数法和共轭梯度法等。对于预测控制系统来说,即使系统的维数不高,但由于优化长度 p 一般不能太小,所涉及的状态变量和控制变量数目较多,因而优化变量的数目也很多,这一规划问题通常是大规模的,这就对在线实施提出了较高的要求。

3 预测控制优化问题的分解协调方法

处理大系统问题的困难之一是由于它的维数太高、变量太多,由此引起计算量大、运算效率低。分解协调方法把整个系统分解为若干相互关联的低维子系统,通过切断或固定相互的关联,将这些子系统作为独立系统进行处理。同时通过对这些子系统进行协调,达到与原问题等价的实现。

分解协调方法已在许多问题中都得到应用^[2,3],但大多是对系统进行空间上的分解,以降低系统的空间维数。Tamura 在1974年提出了离散系统的三级分解协调算法,考虑了时间方向的分解^[4]。对于预测控制方法,当优化步数较长时,时间尺寸也成为影响问题规模及实时性能的一个重要因素,这时时间方向上分解的作用就显得更为突出。为此将一个长区间上优化问题分解为各个时刻的子优化问题,切断各子问题之间的联系,独立各时刻的优化问题。通过协调每个子问题的优化目标,来获得整个区间上的最优解。

若引入协调变量 \mathbf{a} ,则原优化目标可修改并分解为

$$\min_{\mathbf{X}, \mathbf{U}} L(\mathbf{X}, \mathbf{U}, \mathbf{a}) = \sum_{k=0}^p \min_{\mathbf{x}(i+k), \mathbf{u}(i+k)} L_k(\mathbf{x}(i+k), \mathbf{u}(i+k), \mathbf{a}),$$

子问题成为

$$\begin{cases} \min_{\mathbf{x}(i+k), \mathbf{u}(i+k)} L_k(\mathbf{x}(i+k), \mathbf{u}(i+k), \mathbf{a}) \\ \text{s.t. } \mathbf{g}[\mathbf{x}(i+k), \mathbf{u}(i+k)] \geq 0. \end{cases} \quad (k=0, 1, \dots, p).$$

而协调级的任务为通过修改协调变量 \mathbf{a} 来协调各子问题的目标函数 L_k

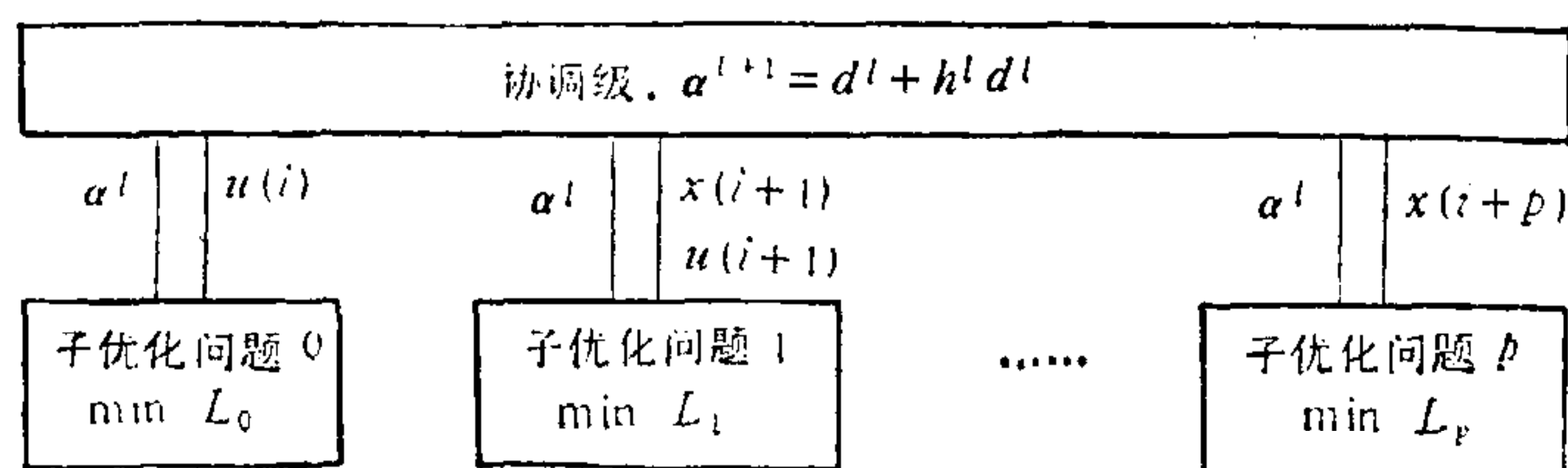


图1 时间分解递阶算法结构

$$\mathbf{a}^{l+1} = \mathbf{a}^l + h^l \mathbf{d}^l$$

其中 h^l 为迭代步长, \mathbf{d}^l 为变化方向, 它一般可从各个子问题返回到协调级的优化结果中得到。

若状态和控制变量的维数分别为 n 和 m , 则子问题的优化变量为 $(m+n)$ 个, 而在整体求解时的数目为 $p \times (m+n)$ 个。由此可见, 当优化步长 p 较长时, 两者的计算任务相差是比较大的。

整个算法结构如图 1 所示。

由于分解时切断了各子问题间的关联, 所以各子问题的处理是独立进行的, 从而具有并行处理的性质。这对于利用并行处理设备、提高运行速度、改善实时性是十分有利的。

4 预测控制优化计算的二级时间分解算法

设当前时刻为零时刻, 优化长度为 p , 控制向量和状态向量分别为 $\mathbf{U} = [\mathbf{u}^T(0), \mathbf{u}^T(1), \dots, \mathbf{u}^T(p-1)]^T$, $\mathbf{X} = [\mathbf{x}^T(1), \mathbf{x}^T(2), \dots, \mathbf{x}^T(p)]^T$, 初始条件为 $\mathbf{x}(0) = \mathbf{x}_0$ 。将前述的预测控制问题定义为原问题

$$(P) \begin{cases} \min_U J = \sum_{k=0}^p J_k[\mathbf{x}(k), \mathbf{u}(k)], \\ \text{s.t. } \mathbf{x}(k) = \mathbf{f}[\mathbf{x}(k-1), \mathbf{u}(k-1)], & (k=1, 2, \dots, p), \\ \mathbf{g}[\mathbf{x}(k), \mathbf{u}(k)] \geq \mathbf{0} & (k=0, 1, \dots, p), \\ \mathbf{x}(0) = \mathbf{x}_0. \end{cases}$$

引入协调变量 $\mathbf{a} = \{\mathbf{a}^T(1), \mathbf{a}^T(2), \dots, \mathbf{a}^T(p)\}^T$, 定义对偶问题

$$(DP) \begin{cases} \max_{\mathbf{a}} \{ \min_{\mathbf{X}, \mathbf{U}} L(\mathbf{X}, \mathbf{U}, \mathbf{a}) \} \\ \text{s.t. } \mathbf{g}[\mathbf{x}(k), \mathbf{u}(k)] \geq \mathbf{0}, & (k=0, 1, \dots, p), \\ \mathbf{x}(0) = \mathbf{x}_0. \end{cases}$$

其中 $L(\mathbf{X}, \mathbf{U}, \mathbf{a}) = J + \sum_{k=1}^p \mathbf{a}^T(k)(\mathbf{x}(k) - \mathbf{f}[\mathbf{x}(k-1), \mathbf{u}(k-1)])$ 。

当给定 \mathbf{a} 后, 对 $L(\mathbf{X}, \mathbf{U}, \mathbf{a})$ 进行时间方向上的分解

$$\begin{aligned} L(\mathbf{X}, \mathbf{U}, \mathbf{a}) &= \sum_{k=0}^p J_k[\mathbf{x}(k), \mathbf{u}(k)] + \sum_{k=1}^p \mathbf{a}^T(k)(\mathbf{x}(k) - \mathbf{f}[\mathbf{x}(k-1), \mathbf{u}(k-1)]) \\ &= \sum_{k=0}^p L_k(\mathbf{x}(k), \mathbf{u}(k), \mathbf{a}), \end{aligned}$$

其中

$$\begin{aligned} L_0(\mathbf{u}(0), \mathbf{a}) &= J_0[\mathbf{u}(0)] - \mathbf{a}^T(1)\mathbf{f}[\mathbf{x}(0), \mathbf{u}(0)], \\ L_k(\mathbf{x}(k), \mathbf{u}(k), \mathbf{a}) &= J_k[\mathbf{x}(k), \mathbf{u}(k)] + \mathbf{a}^T(k)\mathbf{x}(k) \\ &\quad - \mathbf{a}^T(k+1)\mathbf{f}[\mathbf{x}(k), \mathbf{u}(k)], \quad (k=1, 2, \dots, p-1), \\ L_p(\mathbf{x}(p), \mathbf{a}) &= J_p[\mathbf{x}(p)] + \mathbf{a}^T(p)\mathbf{x}(p). \end{aligned}$$

由于各个时刻的 $\mathbf{x}(k), \mathbf{u}(k)$ 作为独立的优化变量, 所以 $L(\mathbf{X}, \mathbf{U}, \mathbf{a})$ 的优化在时间上可分, 所以有

$$\min_{\mathbf{x}, \mathbf{u}} L(\mathbf{X}, \mathbf{U}, \mathbf{a}) = \sum_{k=0}^p \min_{\mathbf{x}(k), \mathbf{u}(k)} L_k(\mathbf{x}(k), \mathbf{u}(k), \mathbf{a}).$$

由此得到子问题 Sub_P_k ($k = 0, 1, 2, \dots, p$) 的表示

$$(\text{Sub_P_}k) \begin{cases} \min_{\mathbf{x}(k), \mathbf{u}(k)} L_k(\mathbf{x}(k), \mathbf{u}(k), \mathbf{a}), \\ \text{s.t. } \mathbf{g}[\mathbf{x}(k), \mathbf{u}(k)] \geq 0. \end{cases}$$

这一子问题的优化变量仅仅是该时刻的 \mathbf{x}, \mathbf{u} , 因而优化问题的规模大大减小.

在协调级, 为求解 $\max_{\mathbf{a}} \{\min_{\mathbf{x}, \mathbf{u}} L(\mathbf{X}, \mathbf{U}, \mathbf{a})\}$, 可选择其梯度或共轭梯度方向作为协调变量迭代方向. 梯度的计算可由子问题的计算结果求得 (l 为协调级迭代步数)

$$\nabla_{\mathbf{a}} L^l = \left. \frac{\partial L}{\partial \mathbf{a}} \right|_{\mathbf{a}=\mathbf{a}^l} l = \begin{bmatrix} \mathbf{x}^l(1) - \mathbf{f}[\mathbf{x}^l(0), \mathbf{u}^l(0)] \\ \mathbf{x}^l(2) - \mathbf{f}[\mathbf{x}^l(1), \mathbf{u}^l(1)] \\ \dots \\ \mathbf{x}^l(p) - \mathbf{f}[\mathbf{x}^l(p-1), \mathbf{u}^l(p-1)] \end{bmatrix}.$$

选用梯度法时的协调级任务为

$$(C) \quad \mathbf{a}^{l+1} = \mathbf{a}^l + h^l \nabla_{\mathbf{a}} L.$$

由此得到时间分解算法如下:

- 1) 给定协调变量初值 $\mathbf{a}^0 = \{\mathbf{a}^0(1)^T, \mathbf{a}^0(2)^T, \dots, \mathbf{a}^0(p)^T\}^T$.
- 2) 求解各个子问题

$$\begin{cases} \min_{\mathbf{x}(k), \mathbf{u}(k)} L_k(\mathbf{x}(k), \mathbf{u}(k), \mathbf{a}^l), \\ \text{s.t. } \mathbf{g}[\mathbf{x}(k), \mathbf{u}(k)] \geq 0, \end{cases}$$

得到子问题最优解 $\mathbf{x}^l(k), \mathbf{u}^l(k)$, ($k = 0, 1, \dots, p$).

- 3) 计算协调变量的梯度方向 $\nabla_{\mathbf{a}} L^l$.
- 4) 如果 $\|\nabla_{\mathbf{a}} L^l\| > \varepsilon$ (ε 为给定精度), 进行协调级计算

$$\mathbf{a}^{l+1} = \mathbf{a}^l + h^l \nabla_{\mathbf{a}} L^l,$$

令 $l = l + 1$, 返回步骤 2), 否则, 计算结束.

5 仿真结果

选取两例, 对每例都分别采用了整体的非线性规划方法和上述时间分解递阶方法, 其具体做法为罚函数法结合共轭梯度法.

例 1. 子优化问题可以得到解析解的情况.

性能指标为

$$\min J = \sum_{k=1}^{50} \{0.05(x_1(k) - 1)^2 + 0.05(x_2(k) - 1)^2 + 0.1u_1^2(k-1) + 0.05u_2^2(k-1)\}.$$

系统模型为

$$\begin{aligned}x_1(k+1) &= 0.9x_1(k) + 0.1x_2(k) + 0.1u_1(k), \\x_2(k+1) &= 0.2x_1(k) + 0.1x_2(k) - 0.1x_2^2(k) + 0.1u_2(k), \\(k &= 0, 1, \dots, 49).\end{aligned}$$

初始条件 $x_1(0) = 10, x_2(0) = 4.5$.

各时刻子问题的最优解为

$k = 0$ 时, 有

$$u_1(0) = \alpha_1(1)/2, \quad u_2(0) = \alpha_2(1).$$

$k = 1, 2, \dots, 49$ 时, 有

$$u_1(k) = \alpha_1(k+1)/2, \quad u_2(k) = \alpha_2(k+1),$$

$$x_1(k) = 1 - 10\alpha_1(k) + 9\alpha_1(k+1) + 2\alpha_2(k+1),$$

$$x_2(k) = [1 - 10\alpha_2(k) + \alpha_1(k+1) + \alpha_2(k+1)]/[1 + 2\alpha_2(k+1)].$$

$k = 50$ 时, 有

$$x_1(50) = 1 - 10\alpha_1(50), \quad x_2(50) = 1 - 10\alpha_2(50).$$

仿真表明两种算法得到了相同的结果(见图 2), 但整体非线性规划方法计算时间是 19.33 秒, 时间分解方法的计算时间是 0.44 秒。后者的速度比前者要快得多, 这在一定程度上是由于可以通过显式直接求得子问题最优解的缘故。但当无法获得解析解, 而需要进行迭代计算时, 计算速度也会随之降低, 见例 2。

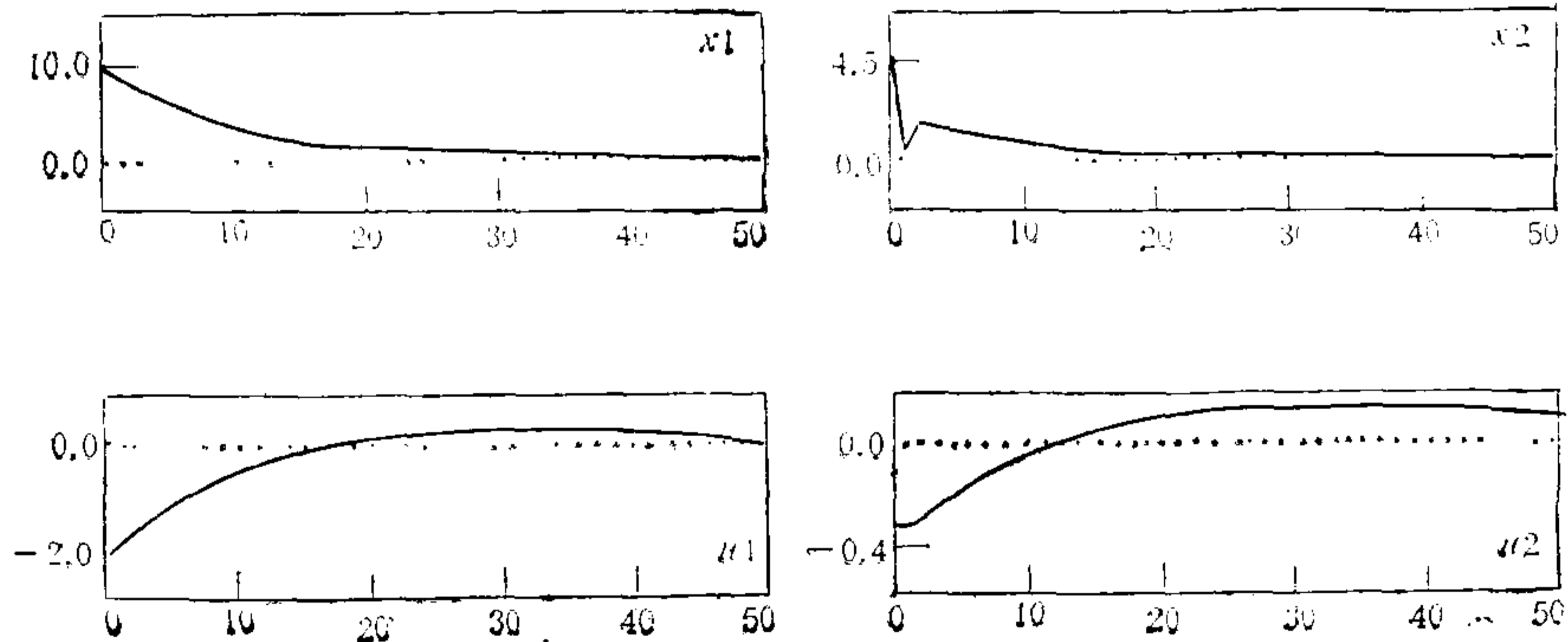


图 2 控制变量及状态变量最优解

例 2. 不存在子问题显式解的情况。

性能指标为

$$\min J = \sum_{k=1}^{20} \{x_1^2(k) + x_2^2(k) + u_1^2(k-1) + u_2^2(k-1)\}.$$

系统模型为

$$x_1(k+1) = x_1(k) - 0.1x_2(k) + 0.1x_1(k) \sin(x_2(k)) + 0.1u_1(k),$$

$$x_2(k+1) = x_2(k) + 0.1x_1(k) + 0.1u_2(k) + 0.2,$$

$$(k = 0, 1, \dots, 19).$$

初始条件为 $x_1(0) = 0.5, x_2(0) = 0.1$.

整体求解时的变量数目为 $20 \times (2 + 2) = 80$, 在采用时间分解方法时, 问题分解为 21 个子问题, 除第一和最后一个子问题只有两个优化变量外, 其余每个子问题均只有四变量, 由此可以看到问题规模的降低。这里采用非线性规划方法求解子问题, 用罚函数法将其化为无约束问题, 然后再用共轭梯度法进行寻优。

计算结果为, 整体非线性规划方法需 30.80 秒, 时间分解方法需 96.56 秒。后者的计算时间大于前者, 但后者是在完全串行的方式下计算的, 如果考虑到它具有的并行结构, 则可以引用并行处理器来提高整体计算速度。在没有并行计算设备的情况下, 通过串行计算的结果来估计出相当于并行处理时的速度。此例在单台设备速度不变的情况下, 用多台设备并行处理, 计算时间将近似为 9.11 秒。

为解决长时段预测控制的优化计算问题, 根据大系统的分解协调方法, 对原问题进行时间方向上的分解。从仿真结果可以看出, 这种算法能够简化问题的规模, 在某些情况下, 还可以因此而获得解析解。此外, 它的并行结构也为采用并行设备来提高预测控制的实时计算效率提供了可能。

参 考 文 献

- [1] Zhang Z J, Xi Y G, Xu X M. Hierarchical predictive control of large scale industrial systems. 10th IFAC World Congress, 1987, 7:109—114.
- [2] Singh M G. Dynamical hierarchical control. North-Holland, 1981.
- [3] 席裕庚. 动态大系统方法导论. 国防工业出版社, 1988.
- [4] Tamura H. Decentralized optimization for distributed_Lag models of discrete systems, *Automatica*, 1975, 11:593—602.
- [5] Hassen M, Singh M G. The optimization of non-linear systems using a new two level method, *Automatica*, 1976, 12:359—363.

TIME DECOMPOSITION FOR MOVING HORIZON OPTIMIZATION IN PREDICTIVE CONTROL

YANG JIAN XI YUGENG ZHANG ZHONGJUN

(Institute of Automation, Shanghai Jiao Tong University 200030)

ABSTRACT

Based on the decomposition-coordination strategy for large scale system, a time decomposition algorithm with parallel structure for predictive control systems is presented, to improve the computational efficiency of the moving horizon optimization to be served online. The effectiveness of the new algorithm is illustrated by simulation results.

Key words: Predictive control, moving horizon optimization, decomposition-coordination, parallel processing.



杨 健 1965 年生于安徽。1988 年于中国科技大学自动化系获学士学位, 1990 年于上海交通大学自动控制系获硕士学位, 现在上海交通大学自动控制系攻读博士学位, 研究方向为工业过程的预测控制。

席裕庚 照片、简介见本刊第 17 卷第 6 期。

张钟俊 照片、简介见本刊第 17 卷第 6 期。

1996 IEEE 系统、人与控制论国际会议 征 文 通 知

1996 IEEE 系统、人与控制论国际会议将于 1996 年 10 月 14 日至 17 日在北京召开, 会议将由中国系统工程学会和清华大学经济管理学院共同主办。

1 会议主题

信息·智能·系统(Information, Intelligence and Systems)

2 征文内容

涉及系统工程、管理科学、自动化、机器人、人工智能、计算机科学等方面的有关问题、技术、理论方法等内容的论文均属本届会议的征集范围。

3 征文要求

a) 论文详细摘要中、英文(800 英文单词)各一份, 1996 年 1 月 15 日前寄至会议秘书处, 过期不予受理。

b) 应征论文在理论、方法、应用方面应有独创性、开拓性的见解, 在学术上、技术上有较大发展、提高或较大的实用价值, 概念清晰、论点明确、分析严谨。

c) 论文应以新的研究成果为主, 且未在国际会议或国内外刊物上发表过。

d) 征文请自留底稿, 均不退稿。

e) 论文投寄地址: 北京清华大学经济管理学院 SMC'96 秘书组 陈剑收

邮编: 100084 联系电话: (10)2595536 Fax: (10) 2595876