第 19 卷第 5 期
2002 年 10 月

控制理论与应用
CONTROL THEORY AND APPLICATIONS

Vol.19 No.5
Oct. 2002

文章编号：1000-8152(2002)05-0793-04

分布式预测控制优化算法*

杜晓宁，席裕庚，李少远
(上海交通大学 自动化研究所，上海 200030)

摘要：复杂大规模预测控制系统的在线实施一直是工业界十分关注的问题之一，本文针对工业环境下分布式网络结构的特点，提出了一种基于纳什最优的分布式预测控制优化算法，在以低成本在线实施的同时，保持了良好的控制性能，文中进一步给出了分布式线性模型预测控制算法的收敛条件，仿真结果表明了该算法的有效性.

关键词：分布式控制；模型预测控制；纳什最优；多智能体

中图分类号：TP273　　**文献标识码：**A

Distributed optimization algorithm for predictive control

DU Xiao-ning, XI Yu-geng, LI Shao-yuan
(Institute of Automation, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: On-line optimization of predictive control strategy for the complex large-scale systems has been one of the focuses in industry. In this paper, a distributed model predictive control algorithm based on Nash optimality is proposed, which accords with the characteristics of distributed network under the industrial environment and can greatly improve the control performance with low cost. The convergence condition for distributed linear model predictive control algorithm is also given, and the simulation results demonstrate its effectiveness.

Key words: distributed control; model predictive control; Nash optimality; multi-agent

1 引言(Introduction)

随着计算机网络的迅速发展，控制作为一种信息处理手段已不限于集中式的控制，而更多地为分布式控制所取代，这就给传统的控制问题带来了新的挑战和要求.预测控制作为一种先进的过程控制方法已被广泛地应用在各种工业过程中[1]，实际系统中大量而广泛存在的是复杂高维的大规模系统，其变量和约束的数目常常是几十个甚至是上百个，对于这样一类大规模预测控制的在线实现问题就显得十分重要[2,3].如果采用集中式的整体求解，则对计算机的性能和处理速度等要求很高，而在实际工业系统中必须考虑经济性，因此人们更愿意采用多台廉价的微机取代高性能计算机对复杂的工业对象进行控制，同时希望保持良好的控制性能，通信网络的建立和现场总线的发展为这种需求提供了可能性.我们可以把一个复杂大规模系统的在线求解问题分散到各个子系统中去分布实现，同时各个子系统之间通过网络进行通信和信息共享，从而降低了问题的规模和复杂性，提高了控制性能.本文正是基于这样的目的，提出了一种适合于分布式网络结构的分布式预测控制算法，它不同于文献[4]中提出的分散预测控制，因为考虑到了子系统之间的相互通信，因此大大提高了系统的控制性能，使整个系统具有纳什最优.

2 分布式模型预测控制(Distributed model predictive control)

模型预测控制，又叫做滚动时域控制，是一种基于模型的开环最优控制，它在每一采样时刻对有限时域上的性能指标进行在线优化，得到一组最优控制序列，并把序列中的第一个控制作用到对象上去.我们这里提出的分布式预测控制在实施的结构上借鉴了多智能体中的理论和思想，把网络环境下的每一个控制器(子系统)看作为一个智能体，每个智能体能够相对独立地进行局部控制和工作，但不能独立完成整个系统的任务，它们通过网络彼此互连，共享资源，相互通信，来共同完成整个系统的任务.假设整个系统的行为可以看作是由 N 个智能体共同作用的结果，在 k 时刻预测系统未来 P 个时刻的输出为：

* 基金项目：国家自然科学基金重点项目(69934020)和国家 973 计划(G1998030415)资助项目.
收稿日期：2000-06-26；收修改稿日期：2001-04-16.

$$Y_{PM}(k) = f(Y_{P0}(k), \Delta u_{1,M}(k), \cdots, \Delta u_{N,M}(k)). \tag{1}$$

这里

$$Y_{PM}(k) = [Y(k+1 \mid k) \quad \cdots \quad Y(k+P \mid k)]^{\mathrm{T}},$$
$$Y_{P0}(k) = [Y_0(k+1 \mid k) \quad \cdots \quad Y_0(k+P \mid k)]^{\mathrm{T}}$$

为 k 时刻的初始预测值,

$$\Delta u_{i,M}(k) = [\Delta u_i(k) \quad \cdots \quad \Delta u_i(k+M-1)]^{\mathrm{T}},$$

P 为预测时域,M 为控制时域,f 为映射函数向量,其中的元素是满足某些平滑条件的非线性函数.系统的输入输出变量满足如下的约束条件:

$$\begin{cases} \Delta u_{i,\min} \leqslant \Delta u_i(\cdot) \leqslant \Delta u_{i,\max}, \\ u_{i,\min} \leqslant u_i(\cdot) \leqslant u_{i,\max}, \\ Y_{\min} \leqslant Y(\cdot) \leqslant Y_{\max}, \end{cases} \tag{2}$$

并令

$$\begin{cases} \Delta u_i(k+j) = \Delta u_i(k+M-1), \\ j = M, \cdots, P, \ i = 1, \cdots, N. \end{cases} \tag{3}$$

系统的优化性能指标为:

$$\min_{\Delta u_{1,M}(k), \cdots, \Delta u_{N,M}(k)} J = \sum_{j=1}^{P} L[Y(k+j \mid k), \Delta u_{1,M}(k), \cdots, \Delta u_{N,M}(k)]. \tag{4}$$

L 为输出变量和输入增量的非线性函数,它满足某些平滑条件使问题的解存在.整个系统的目标就是每一时刻在预测方程(1)下求解性能指标(4),并满足约束条件(2).显然,当系统的维数很高时,加上控制时域 M 的影响,优化变量 $\Delta u_{1,M}(k), \cdots, \Delta u_{N,M}(k)$ 有很高的维数,这样一个大规模非线性优化问题的求解必须采用计算能力极强的高性能计算机.

为了在网络环境下用常规计算机实现上述预测控制算法,我们把整体问题分散到 N 个智能体中分布求解.假设性能指标(4)对于 N 个子系统是可分的,则可把第 i 个智能体的性能指标取作

$$\min_{\Delta u_{i,M}} J_i = \sum_{j=1}^{P} L_i[y_i(k+j \mid k), \Delta u_{i,M}(k \mid k)]. \tag{5}$$

其未来的预测输出根据(1)式可表示为:

$$y_{i,PM}(k) = f[y_{i,P0}(k), \Delta u_{1,M}(k), \cdots, \Delta u_{N,M}(k)] \ (i = 1, \cdots, N). \tag{6}$$

式中

$$Y_{i,PM}(k) = [y_i(k+1 \mid k) \quad \cdots \quad y_i(k+P \mid k)]^{\mathrm{T}},$$
$$Y_{i,P0}(k) = [y_{i,0}(k+1 \mid k) \quad \cdots \quad y_{i,0}(k+P \mid k)]^{\mathrm{T}}.$$

这样,在分布式控制中,整体性能指标被分解到每个智能体中,但每个智能体的输出仍然与所有的输入变量有关.解决这种不同目标的分布控制问题可以借助纳什最优方法,即每一智能体在假定已知其它智能体最优解的前提下,只用自己的输入变量对自己的目标进行优化,即

$$\min_{\Delta u_{i,M}(k)} J_i \mid_{\Delta u^*_{j,M}(k)(j=1,\cdots,N, j \neq i)} \tag{7}$$

$$\text{s.t.} \quad y_{i,PM}(k) = f_i[y_{i,P0}(k), \Delta u^*_{1,M}(k), \cdots, \Delta u^*_{i-1,M}(k), \Delta u_{i,M}(k), \Delta u^*_{i+1,M}(k), \cdots, \Delta u_{N,M}(k)],$$
$$\Delta u_{i,\min} \leqslant \Delta u_i(\cdot) \leqslant \Delta u_{i,\max},$$
$$u_{i,\min} \leqslant u_i(\cdot) \leqslant u_{i,\max},$$
$$y_{i,\min} \leqslant y_i(\cdot) \leqslant y_{i,\max}.$$

这样导致的最优解满足纳什最优性条件

$$J_i(\Delta u^*_{1,M}(k), \cdots, \Delta u^*_{i,M}(k), \cdots, \Delta u^*_{N,M}(k)) \leqslant J_i(\Delta u^*_{1,M}(k), \cdots, \Delta u_{i,M}(k), \cdots, \Delta^*_{N,M}(k)). \tag{8}$$

可以看出每个智能体在求纳什最优解时,都需要知道其它智能体的最优解,才能使整个系统达到纳什最优,这是不可能实现的.但由于分布式结构提供了网络通信环境,各智能体可以互通信息,这就为迭代求解提供了基础.首先,每一智能体对 k 时刻的初始最优解进行预估,在了解其它智能体预估最优解的基础上求出自身的纳什最优解;然后每个智能体将新求出的最优解与上次计算的最优解进行比较,根据终端迭代条件判断是否需要继续进行迭代,并通过网络相互通报各自新的最优解和检查结果.如果两次迭代结果的误差不满足给定的精度,说明整个系统还未达到纳什平衡,进一步改变控制决策还可以使局部和整体的性能指标变得更好,因此需要继续迭代,直至两次结果接近,满足给定的精度,此时,迭代过程结束.如果算法是收敛的,那么在某一轮迭代后系统可达到纳什平衡,此时每一个智能体求出的最优解均满足纳什最优性条件,该时刻的求解即告结束,下一时刻重复上述的优化过程.

通过上述分布式滚动求解,一个大规模的在线优化问题被各智能体小规模的分布优化所取代,从而降低了对计算机的性能要求;同时由于各智能体之间的信息通信,克服了传统全分散预测控制由于信息不足带来的性能下降,仍能使系统的总体控制性能保持在较优的水平上.因此,它适合于复杂大规模系统带有约束的非线性预测控制问题.下面给出分布式滚动求解的算法步骤:

Step 1 在 k 时刻,每个智能体给出控制量的预估初值,并通知给其它各智能体,令 $l = 0$,

$$\Delta u^l_{i,M}(k) =$$

$[\Delta u_i^l(k), \Delta u_i^l(k+1), \cdots, \Delta u_i^l(k+M-1)]^{\mathrm{T}} (i = 1, \cdots, N)$.

Step 2　每个智能体并行地计算各自的预测控制优化问题(7)，得到本次迭代的最优解 $\Delta u_{i,M}^{l+1}(k) (i = 1, \cdots, N)$.

Step 3　检查所有智能体的预估迭代收敛条件是否满足，即对给定的精度 $\varepsilon_i (i = 1, \cdots, N)$，是否有

$$\| \Delta u_{i,M}^{l+1}(k) - \Delta u_{i,M}^{l}(k) \| \leqslant \varepsilon_i (i = 1, \cdots, N).$$

如果所有智能体的迭代收敛条件均成立，则 $\Delta u_{i,M}^{*}(k) = \Delta u_{i,M}^{l+1}(k) (i = 1, \cdots, N)$，迭代计算结束，转 Step 4；否则，令 $\Delta u_{i,M}^{l}(k) = \Delta u_{i,M}^{l+1}(k) (i = 1, \cdots, N)$, $l = l+1$，返回 Step 2.

Step 4　计算 k 时刻即时控制律 $\Delta u_i(k) = [I \ \cdots \ 0] \Delta u_{i,M}^{*}(k) (i = 1, \cdots, N)$，并将该即时控制律作为智能体的最优控制律作用到各智能体中.

Step 5　滚动移位到下一时刻，即 $k+1 \to k$，返回到 Step 1，重复以上过程.

本节提出的分布式优化控制算法适合于一般的非线性约束系统，为了更好地说明该算法，下面将结合无约束线性系统的特例具体分析和说明.

3　基于纳什最优的分布式线性模型预测控制算法(Distributed linear model predictive control algorithm based on Nash optimality)

考虑一个多输入多输出的大规模控制系统，控制的目标是使系统的输出达到期望设定值，同时满足整体性能指标最优.我们采用实际中常用的动态矩阵控制(DMC)算法，整个系统的预测输出模型可表示为[5]：

$$Y_{PM}(k) = Y_{P0}(k) + A \Delta u_M(k). \tag{9}$$

其性能指标为：

$$\min J(k) = \| \omega(k) - Y_{PM}(k) \|_Q^2 + \| \Delta u_M(k) \|_R^2. \tag{10}$$

这里 $\omega(k)$ 为系统期望输出，$\Delta u_M(k) = [\Delta u_{1,M}(k) \ \cdots \ \Delta u_{N,M}(k)]^{\mathrm{T}}$，$A$ 为动态矩阵.现采用分布式进行求解，将整个系统分解为 N 个智能体子系统，其中第 i 个智能体的预测模型可描述为：

$$y_{i,PM}(k) = y_{i,p0}(k) + A_{ii} \Delta u_{i,M}(k) + \sum_{\substack{j=1 \\ j \neq i}}^{n} A_{ij} \Delta u_{j,M}(k) \ (i = 1, \cdots, N). \tag{11}$$

上式右边第三项反映了其它智能体输入对第 i 个智能体输出的影响，A_{ii}，A_{ij} 分别为第 i 个智能体的动态矩阵和第 j 个智能体对第 i 个智能体的阶跃响应矩阵，有

$$A_{ij} = \begin{bmatrix} a_{ij}(1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ a_{ij}(M) & \cdots & a_{ij}(1) \\ \vdots & \vdots & \vdots \\ a_{ij}(P) & \cdots & a_{ij}(P-M+1) \end{bmatrix},$$

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{N1} & \cdots & A_{NN} \end{bmatrix}.$$

$a_{ij}(k) (i, j = 1, \cdots, N; k = 1, 2, \cdots)$ 为阶跃响应序列.

对性能指标(10)按 N 个智能体进行分解，则第 i 个智能体子系统的性能指标为：

$$\min J_i = \| \omega_i(k) - y_{i,PM}(k) \|_{Q_i}^2 + \| \Delta u_{i,M}(k) \|_{R_i}^2 (i = 1, \cdots, N). \tag{12}$$

这里，$\omega_i(k) = [\omega_i(k+1) \ \cdots \ \omega_i(k+P)]^{\mathrm{T}} (i = 1, \cdots, N)$ 为第 i 个智能体的期望输出值，Q_i，R_i 为相应的权矩阵.

根据纳什最优的概念，由(7)式解出智能体 i 的纳什最优解为：

$$\Delta u_{i,M}^{*}(k) = D_{ii} [\omega_i(k) - y_{i,p0}(k) - \sum_{\substack{j=1 \\ j \neq i}}^{n} A_{ij} \Delta u_{j,M}^{*}(k)] (i = 1, \cdots, N). \tag{13}$$

其中 $D_{ii} = (A_{ii}^{\mathrm{T}} Q_i A_{ii} + R_i)^{-1} A_{ii}^{\mathrm{T}} Q_i$，则在 k 时刻实施的即时控制律为：

$$\Delta u_i^{*}(k) = [I \ \cdots \ 0] \Delta u_{i,M}^{*}(k) \ (i = 1, \cdots, N). \tag{14}$$

由(13)式和上一节给出的算法步骤，在 k 时刻智能体 i 的新一轮迭代最优解可写为：

$$\Delta u_{i,M}^{l+1}(k) = D_{ii} [\omega_i(k) - y_{i,p0}(k) - \sum_{\substack{j=1 \\ j \neq i}}^{n} A_{ij} \Delta u_{j,M}^{l}(k)]. \tag{15}$$

令

$$w(k) = [\omega_1(k) \ \cdots \ \omega_N(k)]^{\mathrm{T}},$$
$$y_{P0}(k) = [y_{1,p0}(k) \ \cdots \ y_{N,P0}(k)]^{\mathrm{T}},$$

如果迭代算法是收敛的，则 k 时刻整个分布式系统的最优解可写为：

$$\Delta u_M^{l+1}(k) = D_1 [w(k) - y_{P0}(k)] + D_0 \Delta u_M^l(k). \tag{16}$$

其中

$$D_1 = \begin{bmatrix} D_{11} & & & \\ & D_{22} & & \\ & & \ddots & \\ & & & D_{NN} \end{bmatrix},$$

$$D_0 = \begin{bmatrix} 0 & -D_{11}A_{12} & \cdots & -D_{11}A_{1N} \\ -D_{22}A_{21} & 0 & \cdots & -D_{22}A_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ -D_{NN}A_{N1} & \cdots & \cdots & 0 \end{bmatrix}.$$

为此我们有如下的收敛性定理.

定理 基于纳什最优的分布式动态矩阵控制算法的收敛条件为 $|\rho(D_0)| < 1$,其中 $\rho(D_0)$ 为矩阵 D_0 的谱半径.

证 在 k 时刻 $w(k)$, $y_{P0}(k)$ 均为已知,(16)式中的第一项 $D_1[w(k) - y_{P0}(k)]$ 为与迭代无关的常项,因此式(16)的收敛性即等价于

$$\Delta u_M^{l+1}(k) = D_0 \Delta u_M^l(k) \tag{17}$$

的收敛性,由此得出该分布式优化算法的收敛条件为:

$$|\rho(D_0)| < 1.$$

证毕.

对于线性无约束系统这一特例,我们可以推导出每个智能体分布式求解的解析表达式,这一结果与文献[5]中的解耦设计在形式上是一样的,但实质上则有很大的差别.解耦设计是将控制律事先离线设计好,在实施过程中则不再考虑彼此耦合的影响,而分布式优化算法则是在滚动求解的每一步都要考虑到其它智能体的影响,并将这种影响在优化中考虑进去,通过彼此通信达到 k 时刻的纳什平衡,因此它适用于一般的系统,且能够很好地处理约束问题.由于将整个系统的求解转化为几个小规模的智能体的分布式求解,因而大大降低了在线计算的复杂性.

4 仿真研究(Simulation study)

考虑一个3输入3输出系统

$$G(s) = \begin{bmatrix} \dfrac{e^{-2s}}{100s+1} & \dfrac{e^{-6s}}{100s+1} & \dfrac{e^{-4s}}{200s+1} \\ \dfrac{-1.25e^{-2s}}{50s+1} & \dfrac{3.75e^{-6s}}{50s+1} & \dfrac{e^{-3s}}{50s+1} \\ \dfrac{-2e^{-2s}}{200s+1} & \dfrac{2e^{-4s}}{200s+1} & \dfrac{3.5e^{-2s}}{100s+1} \end{bmatrix}.$$

系统的期望输出设定值为1.我们采用分布式进行求解,将其分散为3个智能体子系统

系统1 $\quad G_1(x) = \dfrac{e^{-2s}}{100s+1}.$

系统2 $\quad G_2(x) = \dfrac{3.75e^{-6s}}{50s+1}.$

系统3 $\quad G_3(x) = \dfrac{3.5e^{-3s}}{100s+1}.$

取预测时域 $P = 8$,控制时域 $M = 3$,性能指标(11)中的加权矩阵为 $Q_i = I, R_i = 0.5I\ (i = 1,2,3)$,采样时间为20s,误差精度 $\varepsilon_i = 0.01\ (i = 1,2,3)$,仿真结果如图1所示.可见,每一智能体输出均能很好地满足期望设定值.由于该分布式控制是在纳什意义下的最优,各智能体可以独立地进行优化,其预测时域,控制时域,权矩阵和采样周期等均可独立地选择,这比集中控制方法要灵活方便得多.

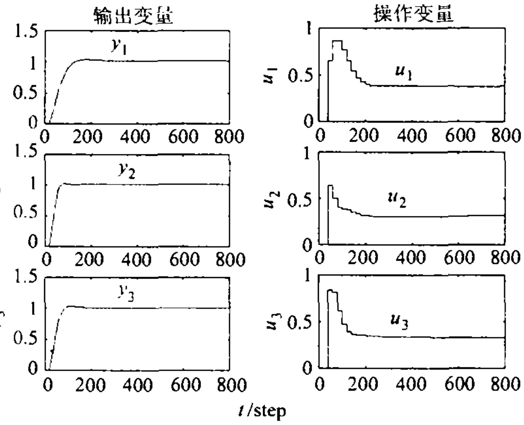

图1 分布式预测控制中的输出变量和操作变量

Fig. 1 Output variables and manipulated variables in DLMPC

5 结论(Conclusion)

本文针对大规模预测控制在线优化实施的复杂性,结合工业环境下分布式网络结构的特点,提出了一种基于纳什最优的分布式预测控制算法.它将一个大规模的在线优化问题转化为各智能体小规模的分布式优化,从而降低了对计算机的性能要求.优化时充分考虑了相互的通信,使每一智能体通过网络充分地获取信息并进行控制,不仅实现了计算的分散化,而且提高了系统的控制性能,因此,它适合于带有约束的复杂大规模系统的预测控制,仿真结果表明了该算法的有效性.

参考文献(References)

[1] Mayne D Q, Rawlings J B, Rao C V, et al. Constrained model predictive control: stability and optimality [J]. Automatica, 2000, 36(6):789-814

[2] Van Antwerp J G, Braatz R D. Model predictive control of large scale process [J]. J. of Process Control, 2000, 10(1):1-8

[3] Katebi M R, Johnson M A. Predictive control design for large-scale systems [J]. Automatica, 1997,33(3): 421-425

[4] Xu X M, Xi Y G, Zhang Z J. Decentralized predictive control (DPC) of large scale systems [J]. Information and Decision Technologies, 1988,14(3):307-322

[5] Xi Yugeng. Predictive Control [M]. Beijing: Defense Industry Press, 1993 (in Chinese)

本文作者简介

杜晓宁 1973年生.现为上海交通大学自动化所博士研究生.研究方向为预测控制,满意决策与控制.

席裕庚 见本刊2002年第2期第187页.

李少远 见本刊2002年第1期第145页.