

文章编号: 1001-0920(2009)11-1757-04

一种改进的 CVP 方法及其在动态优化中的应用

王 平, 田学民

(中国石油大学 信息与控制工程学院, 山东 东营 257061)

摘 要: 针对控制向量参数化方法敏感度方程求解耗时长、时间节点数难确定等问题, 提出一种改进的控制向量参数化方法. 首先利用分段常数对系统敏感度方程进行近似处理, 有效地得到了敏感度方程的近似解析解, 避免了对高维敏感度方程数值积分的计算负担; 然后根据目标函数关于控制参数的敏感度来选择需要细化的控制参数, 得到满足优化精度要求的最优时间节点数. 针对非线性 CSTR 的仿真研究验证了所提出算法的可行性和有效性.

关键词: 控制向量参数化; 敏感度方程; 时间节点细化; 动态优化

中图分类号: TP273

文献标识码: A

Enhanced control vector parameterization method and its application in dynamic optimization

WANG Ping, TIAN Xue-min

(College of Information and Control Engineering, China University of Petroleum, Dongying 257061, China.
Correspondent: TIAN Xue-min, E-mail: tianxm@hdpu.edu.cn)

Abstract: An enhanced control vector parameterization method is presented for the solution of dynamic optimization problems. An analytical solution of sensitivity equations is obtained with approximating the derivative information by using piecewise constant, which avoids expensive computation time to integrate high dimensions sensitivity equations. Then according to the sensitivity of objective with respect to control parameters, the parameters that need further refinement are chosen, and the optimal number of the time intervals under a certain optimization precision is obtained. Finally, simulation study on a nonlinear continuous stirred tank reactor(CSTR) shows the feasibility and effectiveness of the proposed method.

Key words: Control vector parameterization; Sensitivity equation; Time grid refinement; Dynamic optimization

1 引 言

工业过程模型大都是复杂的非线性模型, 存在各种约束条件, 一般用间接方法和迭代动态规划方法难以解决这类模型的动态优化问题. 近年来, 动态优化算法的研究主要集中在直接方法上^[1]. 控制向量参数化方法(CVP)是一种解决过程动态优化问题的有效方法^[2]. 该方法首先将控制变量离散化为有限时间段; 然后在优化时域内对系统模型积分, 将优化问题中的系统模型约束完全消去, 减少了非线性规划(NLP)中的变量个数; 最后用 NLP 方法求解, 以得到优化控制参数.

CVP 的优点在于它是一种可行路径方法, 寻优变量少. 但在每次优化迭代中, CVP 都需要对系统

模型进行数值积分, 耗时长. 因此, CVP 比较适合解决具有多状态低自由度的优化问题, 而不适宜解决具有长优化时域和多自由度的优化问题^[1]. 近几年出现了一些 CVP 的改进算法: Binder^[3]等提出一种自适应 CVP 方法, 该方法通过小波函数分析前一时刻得到的优化控制参数, 自适应地选择需要细化的控制参数, 降低了整体优化的计算量; Schlegel^[4]对该方法做了进一步改进, 并将其应用于间歇过程动态实时优化^[5]和非线性预测控制^[6]; Balsa-Canto^[7]等提出了一种基于二阶导数信息的 CVP 方法, 并利用改进的截断牛顿法求解 NLP 问题^[8], 其结果表明算法具有较高的精度.

为有效地求解动态优化问题, 本文提出一种改

收稿日期: 2008-12-24; 修回日期: 2009-03-19.

基金项目: 国家 863 计划项目(2007AA04Z193).

作者简介: 王平(1983—), 男, 山东胶南人, 博士生, 从事复杂系统的先进控制与优化的研究; 田学民(1955—), 男, 山东文登人, 教授, 博士生导师, 从事过程动态模拟、先进控制与优化等研究.

进的控制向量参数化算法(ECVP).该算法通过对系统敏感度方程的分析,提出用分段常数对敏感度方程进行近似处理,得到其近似解析解,避免了对高维敏感度方程进行数值积分计算,有效地得到了NLP的梯度信息.另外,在分析影响目标函数因素的基础上,根据目标函数与控制参数的敏感度选择需要细化的控制参数,得到了满足优化精度要求的最佳时间节点数.最后通过仿真研究表明了所提出算法的可行性和有效性.

2 动态优化问题描述

工业过程模型通常可以用非线性微分方程组(ODEs)或微分-代数方程组(DAEs)描述.由于DAEs在一定条件下可转化为ODEs,这里只考虑系统模型为ODEs的情况.动态优化是指对过程的操作变量实施控制,使其经济性能指标最优^[1,9].典型的连续过程动态优化问题P1可表示为

$$\min_{\mathbf{u}} J(\mathbf{u}) = \Phi(\mathbf{x}(t_f), t_f) + \int_0^{t_f} \Psi(\mathbf{x}(t), \mathbf{u}(t)) dt. \quad (1)$$

$$\text{s. t. } \frac{d\mathbf{x}}{dt} = f(\mathbf{x}(t), \mathbf{u}(t), t), \mathbf{x}(0) = \mathbf{x}_0; \quad (2)$$

$$h(\mathbf{x}(t), \mathbf{u}(t)) \leq 0. \quad (3)$$

式中: $\mathbf{x} \in R^{n_x}$, $\mathbf{u} \in R^{n_u}$ 分别为状态变量和控制变量; $f(\cdot)$ 和 $h(\cdot)$ 为相应维数的连续可微非线性函数,并假设目标函数是关于 \mathbf{u} 的光滑函数.式(2)定义了系统模型和状态初值条件;式(3)为状态变量和控制变量的路径约束.

CVP首先通过控制参数将控制变量离散化;然后利用离散的控制量在优化时域内对模型(2)积分,将优化问题中的系统模型约束完全消去.一般可用分段常数函数、分段线性函数或高阶B-样条函数进行参数化^[3,4],且大多情况下分段常数函数能够很好地完成对控制量的逼近^[3,4,7].

将优化时域 $[0, t_f]$ 分割为 N 个时间单元,在各时间单元内用分段常数函数逼近连续控制量,即

$$\mathbf{u}(t) = \mathbf{u}_i, t_{i-1} \leq t < t_i, i = 1, 2, \dots, N. \quad (4)$$

式中: $t_i - t_{i-1}$ 为时间单元 i 的长度, N 为时间节点数.状态变量在相邻两个时间单元处连续,即

$$\mathbf{x}(t_i^+) = \mathbf{x}(t_i^-). \quad (5)$$

式中: t_i^- 为第 i 个时间单元的末端时刻, t_i^+ 为第 $i+1$ 个时间单元的起始时刻.参数化后,对于 n_u 维控制量而言,优化变量的维数为 $n_u \times N$ 维,是一个复杂的高维NLP问题.用序列二次规划(SQP)等梯度优化算法求解时,需要计算目标函数关于控制参数的梯度,甚至海森矩阵,计算量将随着节点数 N 的增加而急剧增加.因此,如何快速有效地计算这些梯度

信息,选择合适的节点数 N 便成为CVP成功实施的关键^[3].

3 敏感度方程计算

目标函数关于控制参数的梯度计算方法主要有有限差分法、伴随方程法和敏感度方程法.文献[10]对这3种方法进行了比较,认为与伴随方程法相比,敏感度方程法具有更高的效率.定义状态变量关于控制参数 \mathbf{u}_j 的敏感度为

$$\mathbf{s}_j(t) = \frac{\partial \mathbf{x}(t)}{\partial \mathbf{u}_j}, \frac{\partial \mathbf{x}(0)}{\partial \mathbf{u}_j} = 0, \quad t_{i-1} \leq t < t_i, i, j = 1, \dots, N. \quad (6)$$

对系统模型关于控制参数求偏导,由微分链式法则得系统敏感度方程

$$\frac{d\dot{\mathbf{x}}(t)}{dt} = \left(\frac{\partial f}{\partial \mathbf{x}}\right) \frac{\partial \mathbf{x}(t)}{\partial \mathbf{u}_j} + \left(\frac{\partial f}{\partial \mathbf{u}}\right) \frac{\partial \mathbf{u}(t)}{\partial \mathbf{u}_j}. \quad (7)$$

将式(6)代入(7)得

$$\dot{\mathbf{s}}_j(t) = \left(\frac{\partial f}{\partial \mathbf{x}}\right) \mathbf{s}_j(t) + \left(\frac{\partial f}{\partial \mathbf{u}}\right) \frac{\partial \mathbf{u}(t)}{\partial \mathbf{u}_j}. \quad (8)$$

为获得敏感度方程(8)的解,一般是将式(8)与(2)联立,得到一个 $n_x \times (n_u \times N + 1)$ 维的增广微分方程,然后用数值积分算法求解^[7,8].但这在每次优化迭代中都要求解一个高维微分方程,积分运算量大,有时甚至占整个优化时间的90%以上^[1,7].

式(8)实际上是一个线性连续时变系统,其初值为 $\partial \mathbf{x}(0)/\partial \mathbf{u}_j = 0$.输入 $\partial \mathbf{u}(t)/\partial \mathbf{u}_j$ 是一个脉冲函数,即

$$\frac{\partial \mathbf{u}(t)}{\partial \mathbf{u}_j} = \begin{cases} \mathbf{I}, & t \in [t_{j-1}, t_j]; \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

如果参数化节点数足够多,则在每个时间单元内,敏感度方程都可以近似为一个线性时不变系统.进而对整个优化时域而言,可以用分段常数函数来近似敏感度方程中的 $(\partial f/\partial \mathbf{x})$ 和 $(\partial f/\partial \mathbf{u})$,得到敏感度方程(8)的近似解析解.其具体推导过程如下:

定义

$$f_x(t_i) = \left(\frac{\partial f}{\partial \mathbf{x}}\right) \Big|_{x_i, u_i}, f_u(t_i) = \left(\frac{\partial f}{\partial \mathbf{u}}\right) \Big|_{x_i, u_i},$$

$$A_i = e^{f_x(t_i)\tau}, B_i = \int_0^{\Delta t_i} e^{f_x(t_i)\tau} d\tau f_u(t_i),$$

$$\mathbf{z}_{i,j} = \frac{\partial \mathbf{x}(t_i)}{\partial \mathbf{u}_j}, \mathbf{v}_{i,j} = \frac{\partial \mathbf{u}(t_i)}{\partial \mathbf{u}_j},$$

则可以得到式(8)的离散化形式为

$$\mathbf{z}_{i+1,j} = A_i \mathbf{z}_{i,j} + B_i \mathbf{v}_{i,j}. \quad (10)$$

因为未来的控制量对当前状态量没有影响,所以当 $i = j$ 时, $\mathbf{v}_{i,j} = \mathbf{I}, \mathbf{z}_{i,j} = 0$;当 $i > j$ 时, $\mathbf{v}_{i,j} = 0$.因此有

$$\mathbf{z}_{i+1,j} = \begin{cases} A_i \mathbf{z}_{i,j}, & i > j; \\ B_j, & i = j. \end{cases}$$

递推得

$$z_{i+1,j} = A_i A_{i-1} \cdots A_{j+1} B_j, \quad (11)$$

即得 $s_j(t) = z_{i,j}, t_{i-1} \leq t < t_i, i, j = 1, 2, \dots, N$.

由 $s_j(t)$ 可以进一步计算得到目标函数关于控制参数的导数

$$\begin{aligned} \frac{\partial J(\mathbf{u})}{\partial \mathbf{u}_i} &= \left(\frac{\partial \Phi}{\partial \mathbf{x}(t_f)}\right) s_i(t_f) + \\ &\int_0^{t_f} \left[\left(\frac{\partial \Psi}{\partial \mathbf{x}(t)}\right) s_i(t) + \left(\frac{\partial \Psi}{\partial \mathbf{u}(t)}\right) \frac{\partial \mathbf{u}(t)}{\partial \mathbf{u}_i} \right] dt. \quad (12) \end{aligned}$$

在求解灵敏度方程的过程中需计算大量的导数信息. 利用有限差分求解时, 耗时长, 精度较低, 有时甚至导致优化提前收敛或无法求解^[11]. 自动微分 (AD) 是采用链式求导法则对函数精确求导的一项技术^[12], 当过程模型比较复杂时, 可使用 AD 计算所需导数信息.

4 时间节点选择

一般而言, 选择的时间节点数 N 越大, 划分优化时域得到的时间单元越小, 控制参数逼近控制量的效果越好, 优化结果越精确. 但这也将导致灵敏度方程和 NLP 问题维数越高, 计算负担加重. 因此, 在保证一定优化计算精度的前提下, 希望选择尽量少的节点数 N_1 , 以降低计算量. 本文提出一种新的时间单元划分方法. 该方法首先使用较少的节点数 N_1 , 优化结束后根据目标函数关于控制参数敏感度的绝对值, 选择需要进一步细化的时间单元, 增加相应时间节点, 最终得到满足精度要求的节点数.

为简化推导过程, 这里只考虑控制量是一维的情况. 定义 $\mathbf{u}^1 = [u_1^1, \dots, u_{N_1}^1]$ 和 $J_1^*(\mathbf{u}^1)$ 是节点数为 N_1 时的最优控制参数和最优目标函数. 当优化结果不满足精度要求时, 需要细化优化时域, 则应选择增加那些使优化目标函数下降最大的控制参数. 定义 $\mathbf{u}^2 = [u_1^2, \dots, u_{N_2}^2]$ 和 $J_2^*(\mathbf{u}^2)$ 是节点数为 N_2 时的最优控制参数和目标函数. 不失一般性, 假设在 $t_{N_1-1} \leq t \leq t_{N_1} = t_f$ 内增加一个节点, 即 $N_2 = N_1 + 1$, 如图 1 示. 因为 $J^*(\mathbf{u}^*)$ 是关于 \mathbf{u}^* 的光滑函数, 所以将 $(J_2^* - J_1^*)$ 一阶泰勒展开, 得

$$\begin{aligned} \Delta J^* &= J_2^* - J_1^* = \\ J(\mathbf{u}^{2*}) - J(\mathbf{u}^{1*}) &\approx \left. \frac{\partial J}{\partial \mathbf{u}_i} \right|_{\mathbf{u}_i = \mathbf{u}^{1*}} \Delta \mathbf{u}^*, \quad (13) \end{aligned}$$

式中

$$\begin{aligned} \Delta \mathbf{u}^* &= [(u_{N_1}^{2*} - u_{N_1}^{1*}), \dots, (u_{N_1}^{2*} - u_{N_1}^{1*}), \\ &(u_{N_1+1}^{2*} - u_{N_1}^{1*})]. \end{aligned}$$

由式 (13) 可知, 目标函数下降幅度 (ΔJ^*) 与敏感度的近似关系, 从而可通过设定一个敏感度阈值 $\epsilon \geq 0$ 来判断控制参数 \mathbf{u}_i 是否需要细化: 当

$|\partial J / \partial \mathbf{u}_i| \geq \epsilon$ 时, 细化控制参数 \mathbf{u}_i ; 当 $|\partial J / \partial \mathbf{u}_i| < \epsilon$ 时, 保持 \mathbf{u}_i 不变.

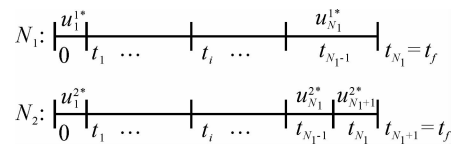


图 1 优化时域的细化过程 ($N_1 \rightarrow N_2$)

5 仿真研究

针对连续搅拌反应器 CSTR^[8] 进行仿真研究. 该 CSTR 内发生 4 个化学反应, 控制量为 3 个进料流量和 1 个电机驱动电量, 其模型如下:

$$\begin{aligned} \dot{x}_1 &= u_4 - (u_1 + u_2 + u_4)x_1 - \\ &17.6x_1x_2 - 23x_1x_6u_3, \\ \dot{x}_2 &= u_1 - (u_1 + u_2 + u_4)x_2 - \\ &17.6x_1x_2 - 146x_2u_3, \\ \dot{x}_3 &= u_2 - (u_1 + u_2 + u_4)x_3 - 73x_2u_3, \\ \dot{x}_4 &= -(u_1 + u_2 + u_4)x_4 + \\ &35.2x_1x_2 - 51.3x_4u_5, \\ \dot{x}_5 &= -(u_1 + u_2 + u_4)x_5 + \\ &219x_2x_3 - 51.3x_4x_5, \\ \dot{x}_6 &= -(u_1 + u_2 + u_4)x_6 + \\ &102.6x_4x_5 - 23x_1x_6u_3, \\ \dot{x}_7 &= -(u_1 + u_2 + u_4)x_7 + 46x_1x_6u_3. \quad (14) \end{aligned}$$

其动态实时优化问题可描述为

$$\begin{aligned} \max_{\mathbf{u}} J &= \\ &\int_0^{t_f} \{ 5.8[(u_1 + u_2 + u_4)x_1 - u_4] - \\ &3.7u_1 - 4.1u_2 + (u_1 + u_2u_4)(23x_4 + \\ &11x_5 + 28x_6 + 35x_7) - 5u_3^2 - 0.099 \} dt; \\ \text{s. t. } &[0, 0, 0, 0]^T \leq \mathbf{u}(t) \leq [20, 6, 4, 20]^T, \\ &x(0) = \\ &[0.1883, 0.2507, 0.0467, 0.0899, \\ &0.1804, 0.1394, 0.1046]^T, \\ &0 \leq t \leq t_f, t_f = 0.2. \quad (15) \end{aligned}$$

定义目标函数为一个新的状态变量 x_8 , 则目标函数转化为 $\max_{\mathbf{u}} J = x_8(t_f)$, 选择控制变量初值 $\mathbf{u}(0) = [10, 3, 2, 6]^T$, 此时目标函数初值 $J_0 = 16.8537$. 使用 ODE45 进行积分运算, 选择绝对积分误差为 1.0×10^{-5} , 利用 Matlab 优化工具箱 fmincon 函数求解 NLP 问题, 其容许误差为 1.0×10^{-5} , 设定敏感度阈值 $\epsilon = 0.02$. 最终优化平均耗时 281 s, 得到的最优节点数 $N = 25$, 优化结果如图 2 ~ 图 4 所示.

由图 2 可以看出, 当节点数 $N = 25$ 时, 目标函数关于控制参数敏感度满足阈值条件, 优化迭代结

束,得到了设定精度下的最小节点数.图3是在优化时域 $[0,0.2]$ 内的最优控制轨迹,相应的目标函数随优化时间的变化曲线如图4示.在同样的计算时间内,ECVP得到的目标值为21.8115,优于Balsa-Canto等采用基于二阶敏感度信息的CVP算法^[7]所得到的21.807,接近于文献[8]所报道的最好结果21.8167.

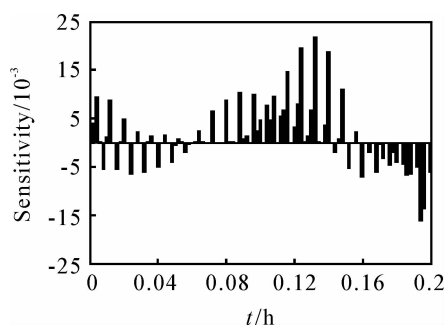


图2 目标函数关于控制参数的敏感度($N=25$)

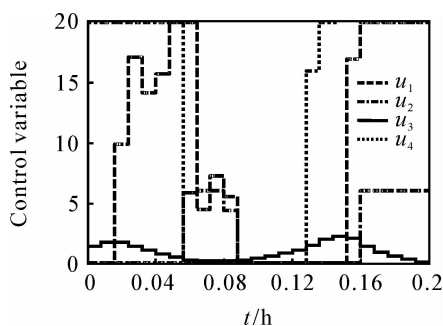


图3 最优控制轨迹($N=25$)

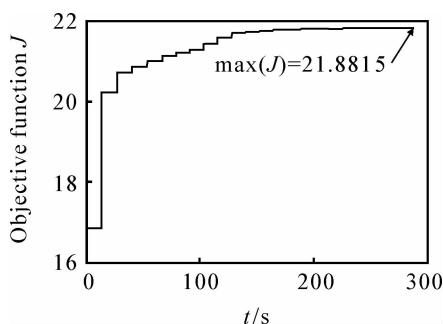


图4 目标函数值随优化时间变化

6 结论

基于过程动态模型的动态优化,对于提升产品质量、增加企业效益具有重要意义.为有效地求解动态优化问题,本文提出了一种改进的控制向量参数化方法.该方法利用分段常数对系统敏感度方程进行近似处理,得到了敏感度方程的近似解析解以及NLP问题的梯度信息,避免了用于数值积分的大量计算时间.另外,利用目标函数与控制参数的敏感度来选择需要细化的控制参数,得到了最佳时间节点

数.CSTR的仿真结果验证了所提出算法的可行性和有效性.

参考文献(References)

- [1] Kameswaran S, Biegler T. Simultaneous dynamic optimization strategies: Recent advances and challenges [J]. *Computers Chemical Engineering*, 2006, 30(10-12): 1560-1575.
- [2] Banga J R, Balsa-Canto E, Carmen G. Dynamic optimization of bioprocesses: Efficient and robust numerical strategies [J]. *J Biotechnol*, 2005, 117(4): 407-419.
- [3] Binder T, Cruse A, Marquardt W. Dynamic optimization using a wavelet based adaptive control vector parameterization strategy [J]. *Computers Chemical Engineering*, 2000, 24(2-7): 1201-1207.
- [4] Schlegel M, Stockman K, Binder T, et al. Dynamic optimization using adaptive control vector parameterization [J]. *Computers Chemical Engineering*, 2005, 29(8): 1731-1751.
- [5] Schlegel M, Marquardt W. Detection and exploitation of the control switching structure in the solution of dynamic optimization problems [J]. *J of Process Control*, 2006, 16(3): 275-290.
- [6] Hartwich A, Schlegel M, Marquardt W. Adaptive control vector parameterization for nonlinear model-predictive control [J]. *Robust Nonlinear Control*, 2008, 18(8): 845-861.
- [7] Balsa-Canto E, Banga J R, Alonso A. Dynamic optimization of chemical and biochemical processes using restricted second order information [J]. *Computers Chemical Engineering*, 2001, 25(4-6): 539-546.
- [8] Balsa-Canto E, Banga J R, Antonio A. Restricted second order information for the solution of optimal control problems using control vector parameterization [J]. *J of Process Control*, 2002, 12(2): 243-255.
- [9] Engell S. Feedback control for optimal process operation [J]. *J of Process Control*, 2007, 17(3): 203-219.
- [10] Rosen O, Luus R. Evaluation of gradients for piecewise constant optimal control [J]. *Computers Chemical Engineering*, 1991, 15(4): 273-281.
- [11] Tolsma J E, Berton P I. On computational differentiation [J]. *Computers Chemical Engineering*, 1998, 22(5): 475-490.
- [12] Li X, Shao Z J, Qian J X. Module-oriented automatic differentiation in chemical process systems optimization [J]. *Computers Chemical Engineering*, 2005, 28(9): 1551-1561.