

一种高效的快速近似控制向量参数化方法

李国栋¹ 胡云卿² 刘兴高¹

摘要 控制向量参数化 (Control vector parameterization, CVP) 方法是目前求解流程工业中最优操作问题的主流数值方法, 然而, 该方法的主要缺点之一是计算效率较低, 这是因为在求解生成的非线性规划 (Nonlinear programming, NLP) 问题时, 需要随着控制参数的调整, 反复不断地求解相关的微分方程组, 这也是 CVP 方法中最耗时的部分. 为了提高 CVP 方法的计算效率, 本文提出一种新颖的快速近似方法, 能够有效减少微分方程组、函数值以及梯度的计算量. 最后, 两个经典的最优控制问题上的测试结果及与国外成熟的最优控制软件的比较研究表明: 本文提出的快速近似 CVP 方法在精度和效率上兼有良好的表现.

关键词 流程工业, 最优控制, 控制向量参数化, 计算效率, 快速近似

引用格式 李国栋, 胡云卿, 刘兴高. 一种高效的快速近似控制向量参数化方法. 自动化学报, 2015, 41(1): 67–74

DOI 10.16383/j.aas.2015.c140031

An Efficient Fast Approximate Control Vector Parameterization Method

LI Guo-Dong¹ HU Yun-Qing² LIU Xing-Gao¹

Abstract The control vector parameterization (CVP) method is currently popular for solving optimal control problems in process industries. However, one of its main disadvantages is the low computational efficiency, because the relevant differential equations in solving the generated nonlinear programming (NLP) problem should be calculated repeatedly with the adjustment of control, which is the most time-consuming part of the CVP method. A new fast approximate approach with less computational expenses on differential equations, function values and gradients is therefore proposed to improve the computational efficiency of the CVP method in this paper. The proposed approach is demonstrated to have marked advantages in terms of accuracy and efficiency, in contrast to mature optimal control softwares, on two classic optimal control problems.

Key words Process industry, optimal control, control vector parameterization (CVP), computational efficiency, fast approximate

Citation Li Guo-Dong, Hu Yun-Qing, Liu Xing-Gao. An efficient fast approximate control vector parameterization method. *Acta Automatica Sinica*, 2015, 41(1): 67–74

在流程工业领域, 复杂生产过程的优化理论与算法研究已成为当前发展趋势, 并最终可归结为最优控制问题^[1]. 最优控制的概念自 20 世纪 50 年代中期提出以来, 由于现实应用价值大, 已引起了国内外相关领域众多学者的关注和研究, 其典型应用包括系统节能、降耗、挖潜、改造中的最优决策,

过程最佳操作参数的确定, 污染物的最佳治理与排放等.

目前, 最优控制问题的数值算法可以分为两大类: 间接法和直接法. 间接法通过求解原问题的最优性条件 (即必要条件), 间接获得原问题的最优解. 典型的间接法^[2–3] 包括控制矢量迭代 (Control vector iteration, CVI)、边界条件迭代 (Boundary condition iteration, BCI) 和拟线性化 (Quasilinearization). 直接法^[4–5] 则是通过离散化, 将无限维的最优控制问题转化为有限维的非线性规划 (Nonlinear programming, NLP) 问题再求解的方法. 仅离散化控制向量而保持状态向量不变的直接法又称为控制向量参数化 (Control vector parameterization, CVP) 方法, 它是当前求解最优控制问题的主流数值方法. 国际上许多优秀的最优控制软件 (如 ACADO、DOTcvp 等) 都是基于该方法开发出来的^[6–7].

收稿日期 2014-01-14 录用日期 2014-05-28
Manuscript received January 14, 2014; accepted May 28, 2014
国家高技术研究发展计划 (863 计划) (2006AA05Z226), 国家自然科学基金 (U1162130), 浙江省杰出青年科学基金项目 (R4100133) 资助
Supported by National High Technology Research and Development Program of China (863 Program) (2006AA05Z226), National Natural Science Foundation of China (U1162130), and Outstanding Youth Science Foundation of Zhejiang Province (R4100133)
本文责任编辑 李少远
Recommended by Associate Editor LI Shao-Yuan
1. 浙江大学工业控制技术国家重点实验室 杭州 310027 2. 南车株洲电力机车研究所有限公司 株洲 412001
1. State Key Laboratory of Industry Control Technology, Zhejiang University, Hangzhou 310027 2. CSR Zhuzhou Institute Co. Ltd., Zhuzhou 412001

CVP 方法具有求解精度高、离散化后的 NLP 问题规模相对较小等优点. 该方法中, 离散化时的时间网格通常是事先确定的, 并且在优化过程中不会改变. 网格划分的疏密会影响到求解结果逼近最优控制轨迹的程度. 若要较好地逼近最优控制轨迹, 就需要将时间网格划分得很细, 但这样会大大增加 NLP 问题的维数和计算时间. 因此, 如何有效解决 CVP 方法逼近精度和计算时间之间的矛盾, 是国际上众多学者都在探索的一个课题^[8-14].

文献 [8, 11, 15] 提出了自适应网格划分的 CVP 方法, 在迭代过程中通过对上一步解的小波分析来调整时间网格的划分, 对某些时间网格进行细化或合并, 从而获得比均匀划分网格的 CVP 方法更高的优化效率. Srinivasan 等^[9] 基于最优控制的必要条件研究了最优控制的各段轨迹与必要条件的对应情况, 将必要条件划分为约束和灵敏度两部分, 提出了一种具自适应控制向量参数化方法. Szymkat 等^[10] 给出了控制和状态约束最优控制问题的结构估计方法, 通过不断增减网格节点和弧段来实现结构调整. 文献 [12-13, 16] 提出了一种控制参数化增强转换方法, 该方法把原时间节点映射到一个新的时间尺度上, 并将原有网格的宽度作为优化变量, 在求解得到最优控制参数的同时获得最佳的网格划分. 这些方法都是通过对时间网格的调整来提高 CVP 方法的效率.

不同于调整时间网格的策略, 本文从一个新的角度来提高 CVP 方法的优化效率. 我们注意到, 在求解 NLP 问题时, 不管是计算函数的值, 还是它们的梯度, 都需要对相关的微分方程组进行求解. 而且, 随着控制参数的不断优化, 微分方程组的计算会反复不断地进行, 直至迭代结束, 是 CVP 方法中最耗时的部分. 当前国际上通常采用自适应变步长的数值积分方法对微分方程组进行计算, 并且为了保证 NLP 问题的优化精度, 微分方程组的求解精度往往会设置得比 NLP 问题的优化精度高 1~2 个数量级^[8, 11, 16-22]. 这样的处理方式虽然能够保证 NLP 问题的精度, 但是会大大增加微分方程组的计算时间, 从而影响 CVP 方法的计算效率. 因此, 为了提高 CVP 方法的整体效率, 本文采用一种新的快速近似策略进行数值积分计算, 大幅减少了微分方程组、函数值和梯度的计算量, 并且使用分段线性连续的参数化方式提高对最优控制轨迹的逼近能力, 在此基础上提出了快速近似的控制向量参数化 (Fast approximate CVP, Fast-CVP) 方法, 所生成的非线性规划问题可用成熟的 NLP 算法快速求解. 与传统的 CVP 处理方法相比, Fast-CVP 更侧重于在有

限时间内对最优控制轨迹的逼近能力. 本文通过两个经典的最优控制问题的测试表明了 Fast-CVP 方法的快速和有效性.

1 问题描述

本文考虑的最优控制问题 P1, 数学描述如下:

$$\min g_0[\mathbf{u}(t)] = \Phi_0[\mathbf{x}(t_f), t_f] + \int_{t_0}^{t_f} L_0[\mathbf{u}(t), \mathbf{x}(t), t] dt \quad (1a)$$

$$\dot{\mathbf{x}}(t) = F[\mathbf{u}(t), \mathbf{x}(t), t] \quad (1b)$$

$$g_i[\mathbf{u}(t)] = \Phi_i[\mathbf{x}(t_f), t_f] + \int_{t_0}^{t_f} L_i[\mathbf{u}(t), \mathbf{x}(t), t] dt = 0, \quad i \in \varepsilon \quad (1c)$$

$$g_i[\mathbf{u}(t)] = \Phi_i[\mathbf{x}(t_f), t_f] + \int_{t_0}^{t_f} L_i[\mathbf{u}(t), \mathbf{x}(t), t] dt \geq 0, \quad i \in \iota \quad (1d)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (1e)$$

$$\underline{\mathbf{u}} \leq \mathbf{u}(t) \leq \bar{\mathbf{u}} \quad (1f)$$

$$t_0 \leq t \leq t_f \quad (1g)$$

其中, $\mathbf{u}(t) \in \mathbf{R}^{n_u}$ 是控制向量, $\mathbf{x}(t) \in \mathbf{R}^{n_x}$ 是状态向量, 操作开始时刻 t_0 和结束时刻 t_f 都是固定的. 过程的特性由常微分方程组 (1b)、规范等式约束 (1c) 和规范不等式约束 (1d) 描述, 其初始可行状态为 $\mathbf{x}_0$, 上下边界约束分别为 $\bar{\mathbf{u}}$ 和 $\underline{\mathbf{u}}$. $g_i, i \in \{0\} \cup \varepsilon \cup \iota$ 是该过程的目标和约束函数, 其中 ε 和 ι 分别为规范等式约束和规范不等式约束的序号集, Φ_i 和 L_i 分别是 g_i 的终值项和积分项.

设 U 是 $\mathbf{u}(t)$ 的可行控制集, 则最优控制问题 (1a)~(1g) 可以简单描述为: 在可行初始条件 (1e) 下, 从可行控制集 U 中求得针对系统 (1b)~(1d) 的最优控制策略 $\mathbf{u}^*(t)$, 使得目标函数 (1a) 的值最小.

2 理论推导及算法描述

2.1 控制向量参数化

控制向量参数化方法把时间域 $[t_0, t_f]$ 划分成 N 个阶段, 记各阶段时间节点为

$$t_0 < t_1 < \cdots < t_N = t_f \quad (2)$$

在时间段 $[t_{k-1}, t_k]$, $k = 1, 2, \cdots, N$ 内, 将控制变量近似地表示为一组参数的函数. 设控制向量的第 j

个分量为 $u_j(t)$, $j = 1, 2, \dots, nu$, 则其参数化的近似表示为

$$u_{jk}(t) \approx \tilde{u}_{jk}(t) = \sum_{l=1}^{n+1} \sigma_{jkl} \phi_{jkl}(t), \quad t \in [t_{k-1}, t_k] \quad (3)$$

其中, σ_{jkl} 为参数, $\phi_{jkl}(t)$ 为基函数, 可以是多项式、泰勒级数、傅里叶级数、切比雪夫级数等, n 为基函数的阶次, j 为控制分量的序号, k 为时间段的序号, l 为在第 k 个时间段上对第 j 个控制分量进行参数化所用参数的序号. 特别地, 当 $\phi_j(t) = 1$ 时, 为分段常量参数化, 这是 CVP 方法中最常用的参数化形式, 具有简便易行、参数数目少、生成的 NLP 问题规模较小等优点. 然而, 分段常量参数化对最优控制轨迹的逼近能力有限, 若要取得较高的逼近精度, 则需要将控制时域划分成很多段, 不利于提高 CVP 方法的效率. 因此, 本文采用逼近能力更高的分段线性连续参数化 (如图 1 所示), 在相同分段数下, 仅比分段常量多 1 个参数, 生成的 NLP 问题规模也与之相近.

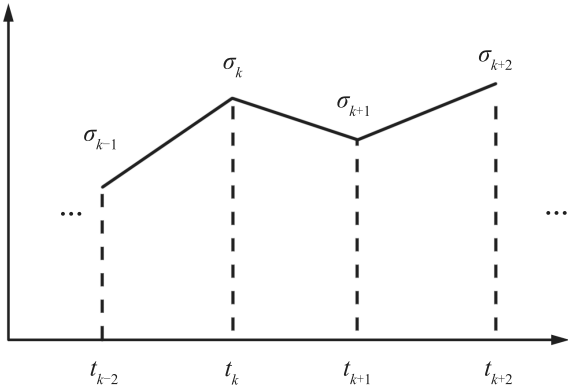


图 1 分段线性连续参数化示意图

Fig. 1 The schematic diagram of piecewise linear continuous parameterization

将控制向量进行分段线性连续参数化后, 记第 j 个控制分量的表征参数依次为 $\sigma_{j,0}, \sigma_{j,1}, \dots, \sigma_{j,N}$, 并记向量 $\sigma_k = [\sigma_{1,k}, \sigma_{2,k}, \dots, \sigma_{nu,k}]^T$, $\sigma = [(\sigma_0)^T, \dots, (\sigma_N)^T]^T$, 则在时间段 $[t_{k-1}, t_k]$ 上, $u_{jk}(t)$ 可近似表示为

$$\tilde{u}_{jk}(t) = \sigma_{j,k-1} \frac{t_k - t}{t_k - t_{k-1}} + \sigma_{j,k} \frac{t - t_{k-1}}{t_k - t_{k-1}} \quad (4)$$

控制向量 $\mathbf{u}(t)$ 的边界约束可以直接转换为控制参数 σ 的边界约束. 通过分段线性连续参数化, 原始的求解最优控制轨迹 $\mathbf{u}(t)$ 的最优控制问题就转化为确定

控制参数 σ 的静态优化问题. 因此, 最优控制问题 P1 可转化为如下形式的非线性规划问题 P2:

$$\min G_0(\sigma) \quad (5a)$$

$$G_i(\sigma) = 0, \quad i \in \varepsilon \quad (5b)$$

$$G_i(\sigma) \geq 0, \quad i \in \iota \quad (5c)$$

$$\underline{\sigma} \leq \sigma \leq \bar{\sigma} \quad (5d)$$

其中, $G_i(\sigma) = \Phi_i[\mathbf{x}(t_f|\sigma), t_f] + \int_{t_0}^{t_f} L_i[\mathbf{u}(t|\sigma), \mathbf{x}(t|\sigma), t] dt$, $i \in \{0\} \cup \varepsilon \cup \iota$ 是控制参数 σ 的函数, $\mathbf{u}(\cdot|\sigma)$ 和 $\mathbf{x}(\cdot|\sigma)$ 分别表示由 σ 所确定的控制向量和状态向量, $\bar{\sigma}$ 和 $\underline{\sigma}$ 分别为控制参数 σ 的上下界, 可直接从边界约束 (1g) 获得. 参数化后的非线性规划问题可使用成熟的、基于梯度的优化算法 (例如 SQP 方法、内点法等) 进行求解. 问题 P1 和 P2 的解之间的关系由下列定理保证 (证明过程可参阅文献 [23–24]).

定理 1^[23]. 设 $\mathbf{u}^*$ 是问题 P1 的解, $\mathbf{u}_N^*$ 是分段数为 N 时问题 P2 的解, 则有 $\lim_{N \rightarrow \infty} g_0(\mathbf{u}_N^*) = g_0(\mathbf{u}^*)$.

定理 2^[24]. 设 $\mathbf{u}_N^*$ 是分段数为 N 时问题 P2 的解, 如果序列 $\{\mathbf{u}_N^*\}_{N=1}^{\infty}$ 在控制时域 $[t_0, t_f]$ 上处处收敛于某一控制轨迹 $\tilde{\mathbf{u}}$, 那么 $\tilde{\mathbf{u}}$ 就是问题 P1 的一个最优控制.

2.2 梯度计算

对控制参数 σ 的梯度可以通过伴随方法计算. 定义函数 G_i , $i \in \{0\} \cup \varepsilon \cup \iota$ 的哈密顿函数为

$$H_i[\sigma, \mathbf{x}(t), \lambda_i(t), t] = L_i + \lambda_i(t)^T F \quad (6)$$

其伴随方程可表示为

$$\dot{\lambda}_i(t) = - \left[\frac{\partial H_i[\sigma, \mathbf{x}(t), \lambda_i(t), t]}{\partial \mathbf{x}(t)} \right]^T \quad (7)$$

$$\lambda_i(t_f) = \left[\frac{\partial \Phi_i[\mathbf{x}(t_f|\sigma), t_f]}{\partial \mathbf{x}(t)} \right]^T \quad (8)$$

则函数 G_i 对所有控制参数的梯度信息可以由式 (9) 推导获得, 证明过程可参考文献 [25].

$$\frac{\partial G_i}{\partial \sigma_{j,r}} = \int_{t_0}^{t_f} \frac{\partial H_i}{\partial \sigma_{j,r}} dt, \quad r = 0, 1, \dots, N \quad (9)$$

接下来, 对式 (9) 做进一步推导, 并给出积分时域更短的表达形式, 有利于减少计算量.

$$\begin{aligned}
\frac{\partial G_i}{\partial \sigma_{j,k}} &= \int_{t_0}^{t_f} \frac{\partial H_i}{\partial \sigma_{j,k}} dt = \int_{t_0}^{t_f} \frac{\partial H_i}{\partial u_j} \frac{\partial u_j}{\partial \sigma_{j,k}} dt = \\
&\int_{t_{k-1}}^{t_k} \frac{\partial H_i}{\partial u_j} \frac{\partial u_j}{\partial \sigma_{j,k}} dt + \int_{t_k}^{t_{k+1}} \frac{\partial H_i}{\partial u_j} \frac{\partial u_j}{\partial \sigma_{j,k}} dt = \\
&\int_{t_{k-1}}^{t_k} \frac{\partial H_i}{\partial u_j} \frac{t - t_{k-1}}{t_k - t_{k-1}} dt + \\
&\int_{t_k}^{t_{k+1}} \frac{\partial H_i}{\partial u_j} \frac{t_{k+1} - t}{t_{k+1} - t_k} dt, \\
&k = 1, \dots, N-1
\end{aligned} \quad (10)$$

同理,

$$\frac{\partial G_i}{\partial \sigma_{j,0}} = \int_{t_0}^{t_1} \frac{\partial H_i}{\partial u_j} \frac{t_1 - t}{t_1 - t_0} dt \quad (11)$$

$$\frac{\partial G_i}{\partial \sigma_{j,N}} = \int_{t_{N-1}}^{t_f} \frac{\partial H_i}{\partial u_j} \frac{t - t_{N-1}}{t_f - t_{N-1}} dt \quad (12)$$

2.3 快速近似

在 NLP 问题 P2 的求解过程中, 需要不断获取函数及梯度信息, 不仅要表达式中的积分项进行计算, 还不可避免的需要对微分方程组进行反复求解 (计算结构如图 2 所示). 传统 CVP 方法中, 通常采用自适应变步长的数值积分方法进行以上运算, 虽然能够保障精度, 但耗时较长.

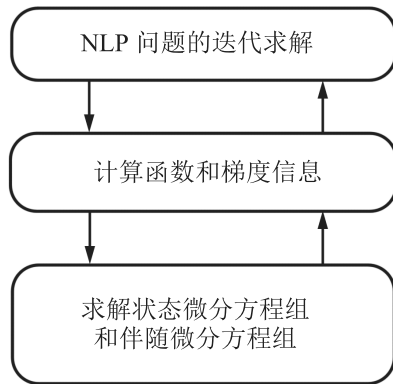


图 2 CVP 方法计算结构图

Fig. 2 The calculation structure of CVP method

本文注意到, 当积分时域较短时, 积分运算可用梯形公式进行近似. 因此, 当分段数 N 非常大时, 时间段长度趋近于零, 梯度计算公式可近似表示为

$$\frac{\partial G_i}{\partial \sigma_{j,k}} \approx \left. \frac{\partial H_i}{\partial u_j} \right|_{t=t_k} \frac{t_{k+1} - t_{k-1}}{2}, \quad k = 1, \dots, N-1 \quad (13)$$

$$\frac{\partial G_i}{\partial \sigma_{j,0}} \approx \left. \frac{\partial H_i}{\partial u_j} \right|_{t=t_0} \frac{t_1 - t_0}{2} \quad (14)$$

$$\frac{\partial G_i}{\partial \sigma_{j,N}} \approx \left. \frac{\partial H_i}{\partial u_j} \right|_{t=t_f} \frac{t_f - t_{N-1}}{2} \quad (15)$$

进一步, 如果控制时域 $[t_0, t_f]$ 被等分为 N 段, 则有

$$\frac{t_f - t_0}{N} = h \quad (16)$$

其中, h 是常数. 从而得到当时间段长度 h 较小情况下的梯度计算公式

$$\frac{\partial G_i}{\partial \sigma_{j,k}} \approx h \left. \frac{\partial H_i}{\partial u_j} \right|_{t=t_k}, \quad k = 1, \dots, N-1 \quad (17)$$

$$\frac{\partial G_i}{\partial \sigma_{j,0}} \approx \frac{h}{2} \left. \frac{\partial H_i}{\partial u_j} \right|_{t=t_0} \quad (18)$$

$$\frac{\partial G_i}{\partial \sigma_{j,N}} \approx \frac{h}{2} \left. \frac{\partial H_i}{\partial u_j} \right|_{t=t_f} \quad (19)$$

式 (17)~(19) 中用常数相乘代替了积分, 计算效率自然会大大提高.

类似地, $G_i(\sigma)$ 中的积分项也可做近似处理, 可得

$$\begin{aligned}
&\int_{t_0}^{t_f} L_i[\mathbf{u}(t|\sigma), \mathbf{x}(t|\sigma), t] dt \approx \\
&\left(\frac{1}{2} L_i|_{t=t_0} + \sum_{k=1}^{N-1} L_i|_{t=t_k} + \frac{1}{2} L_i|_{t=t_f} \right) h \quad (20)
\end{aligned}$$

另一方面, 用式 (17)~(19) 求取梯度信息之前, 必须先求解状态方程组 (1b) 的初值问题和伴随方程组 (7) 的终值问题. 由于采用等间隔的时间网格划分且 h 值较小, 可以考虑用固定步长的数值积分方法求解状态方程组和伴随方程组, 并将直接将时间网格节点作为积分节点.

这里采用四阶 Runge-Kutta 法作为求解状态方程组初值问题和伴随方程组终值问题的方法, 并将 h 作为积分步长、时间网格节点作为积分节点, 推导出的公式为

$$k_1 = F[\sigma_{k-1}, \mathbf{x}(t_{k-1}), t_{k-1}] \quad (21)$$

$$k_2 = F\left[\frac{\sigma_{k-1} + \sigma_k}{2}, \mathbf{x}(t_{k-1}) + \frac{k_1 h}{2}, t_{k-1} + \frac{h}{2}\right] \quad (22)$$

$$k_3 = F\left[\frac{\sigma_{k-1} + \sigma_k}{2}, \mathbf{x}(t_{k-1}) + \frac{k_2 h}{2}, t_{k-1} + \frac{h}{2}\right] \quad (23)$$

$$k_4 = F[\sigma_k, \mathbf{x}(t_{k-1}) + k_3 h, t_k] \quad (24)$$

$$\mathbf{x}(t_k) = \mathbf{x}(t_{k-1}) + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

$$k = 1, \dots, N \quad (25)$$

用式 (21)~(25) 联合计算得到的 $\mathbf{x}(t_k)$ 与真实值的误差为 $O(h^5)$, 是 h^5 的同阶无穷小, 所以当 h 的值很小时, 可得到高精度的求解结果. 类似地, 也可以得到伴随方程组终值问题的计算公式.

通过对梯度和函数计算中的积分计算进行梯形近似, 使用定步长的数值积分方法代替自适应变步长方法来计算微分方程组, 并将网格节点直接作为积分节点, 使得各积分节点的信息得到充分利用, 能够有效减少 NLP 问题求解时, 各个控制参数梯度信息和函数值的计算量. 值得注意的是, 快速近似后的函数和梯度信息, 与近似前相比会存在一定的误差, 但这种误差会随着分段数的增加逐渐趋近于零. 因此, 对于某一确定的精度要求, 一定存在一个正整数 $\bar{N}$, 当分段数 $N > \bar{N}$ 时, 近似后的结果与近似前的结果在该精度条件下相等. 所以, 当 $N \rightarrow \infty$ 时, 对于本文提出的方法, 定理 1 和定理 2 依然成立.

2.4 解最优控制问题的快速近似算法

基于以上描述, 本小节给出快速近似控制向量参数化方法的算法步骤:

步骤 1. 设置分段数 N 和初始控制参数 $\sigma^{(0)}$, 用式 (16) 计算时间段长度 h ;

步骤 2. 通过式 (21)~(25), 解微分方程组的初值问题 (1b) 和 (1e), 得到 $\mathbf{x}(t)$ 的值;

步骤 3. 类似地, 解伴随方程组的终值问题 (7) 和 (8), 得到 $\lambda(t)$ 的值;

步骤 4. 利用式 (17)~(20) 快速获取 $G_i, i \in \{0\} \cup \varepsilon \cup \iota$ 的梯度和函数值;

步骤 5. 基于函数和梯度信息解非线性规划问题 P2, 得到优化后的控制参数 σ^* .

3 实例测试

本节采用两个经典的最优控制问题对提出的

快速近似控制向量参数化方法进行测试. 作为对比, 本文开发了未经快速近似处理的分段线性连续 CVP 程序. 分段数 N 分别取 10、50 和 100, 并将本方法的测试结果与所开发的分段线性连续 CVP 程序及国际上两个优秀的 CVP 动态优化软件 ACADO 和 DOTcnp 的测试结果进行比较, 所有测试均在由 Intel i5 3450/3.1 GHz CPU 和 DDR3/1 600 MHz 内存组成的普通个人微机上运行.

3.1 间歇反应饱和度控制问题

管式间歇反应器中发生并行反应: $A \rightarrow B$ 和 $A \rightarrow C$, 反应速率分别为 k_1 和 k_2 , B 是目标产物, C 是副产物, 目标是控制饱和度使得产物 B 在最终时刻的产量最大. 该问题的数学描述为

$$\begin{aligned} \max \quad & J = x_2(1) \\ \text{s.t.} \quad & \dot{x}_1(t) = -[u(t) + 0.5u^2(t)]x_1(t) \\ & \dot{x}_2(t) = u(t)x_1(t) \\ & \mathbf{x}(0) = [1 \ 0]^T \\ & 0 \leq u(t) \leq 5 \\ & 0 \leq t \leq 1 \end{aligned}$$

初始控制参数 $\sigma^{(0)}$ 统一设置为 1. 表 1 列出了用快速近似控制向量参数化 (Fast-CVP) 方法得到的结果. 当分段数 N 由 10 增加到 100 时, 目标函数值不断增加, 最终耗时 3.6859 s 得到最优目标函数值

表 1 间歇反应饱和度控制问题测试结果
Table 1 Results of batch reactor problem with control saturation

测试对象	分段数 N	目标函数 J	计算时间 (s)
Fast-CVP	10	0.570984	0.1419
	50	0.573534	1.4929
	100	0.573543	3.6859
分段线性连续 CVP	10	0.573298	6.9178
	50	0.573536	65.4533
	100	0.573543	174.3769
ACADO	10	0.572242	0.2537
	50	0.573501	2.1601
	100	0.573534	3.9828
DOTcnp	10	0.572241	9.2660
	50	0.573500	167.2385
	100	0.573533	531.0938

$J^* = 0.573543$. 与分段线性连续 CVP 相比, 快速近似前后目标函数值的误差随着分段数的增加不断减少, 并在 $N = 100$ 时趋近于 0, 但从计算时间上可以看出, 快速近似后的优化效率更高, $N = 100$ 时取得相同结果的优化时间比近似前缩短了 97.9%. 与 ACADO 和 DOTcvp 的结果相比, 不仅耗时更短, 且在 $N = 50$ 和 100 时结果更优. 值得注意的是, 当 Fast-CVP 分段数取 50 时, 所得结果与这两款软件在 100 段时的结果基本一致 (相差都不到十万分之二), 但计算时间要比 ACADO 短 62.5%, 比 DOTcvp 短 99.7%, 再次表明本文所提出的 Fast-CVP 具有更高的优化效率.

图 3(a) 和图 3(b) 分别是 $N = 100$ 时的控制轨迹和状态轨迹. 从图 3(a) 可以看出, 该问题的最

优控制轨迹包含一段曲率较大的弧线. 而分段线性连续的参数化方式比通常使用的分段常量参数化对该类曲线具有更好的逼近能力, 这也可以解释为什么 Fast-CVP 会具有更高的效率.

3.2 Van der Pol 振荡器问题

这是一个最优控制算法的经典测试算例, 其数学模型如下:

$$\begin{aligned} \min \quad & J = x_3(5) \\ \text{s.t.} \quad & \dot{x}_1(t) = [1 - x_2^2(t)]x_1(t) - x_2(t) + u(t) \\ & \dot{x}_2(t) = x_1(t) \\ & \dot{x}_3(t) = x_1^2(t) + x_2^2(t) + u^2(t) \\ & \mathbf{x}(0) = [0 \ 1 \ 0]^T \\ & -0.3 \leq u(t) \leq 1 \\ & 0 \leq t \leq 5 \end{aligned}$$

初始控制参数 $\sigma^{(0)}$ 统一设置为 0.7. 表 2 列出了用 Fast-CVP 方法在不同分段数下得到的结果. 当分段数 N 由 10 增加到 100 时, 目标函数值及其在快速近似前后的误差均不断减少, 最终耗时 0.3372 s 得到最优目标函数值 $J^* = 2.867294$, 相比快速近似前节约了 97.7% 的寻优时间. 同样的, 与 ACADO 和 DOTcvp 的结果相比, 不仅耗时更短, 且结果更优. 当分段数分别取 10、50 和 100 时, Fast-CVP 的计算耗时比 ACADO 平均少 57.6%, 比 DOTcvp 平

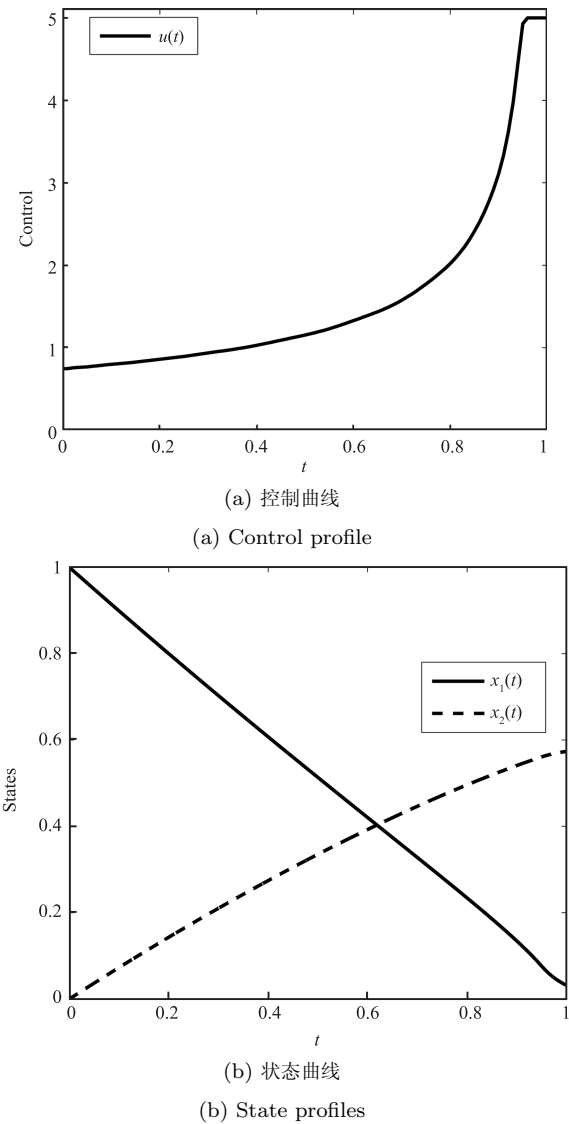


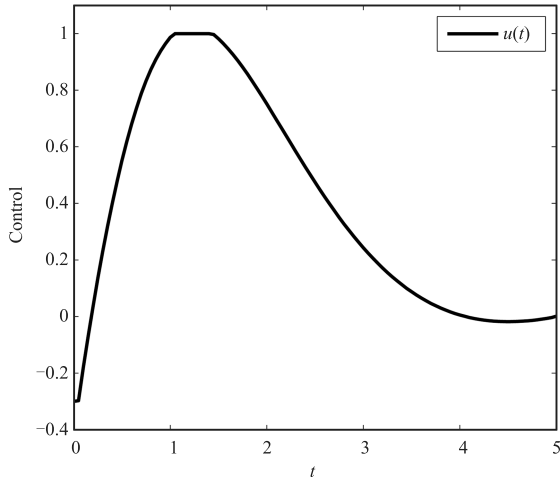
图 3 间歇反应饱和度控制问题控制和状态曲线
Fig.3 Control and state profiles for the batch reactor problem with control saturation

表 2 Van der Pol 振荡器问题测试结果
Table 2 Results of Van der Pol oscillator problem

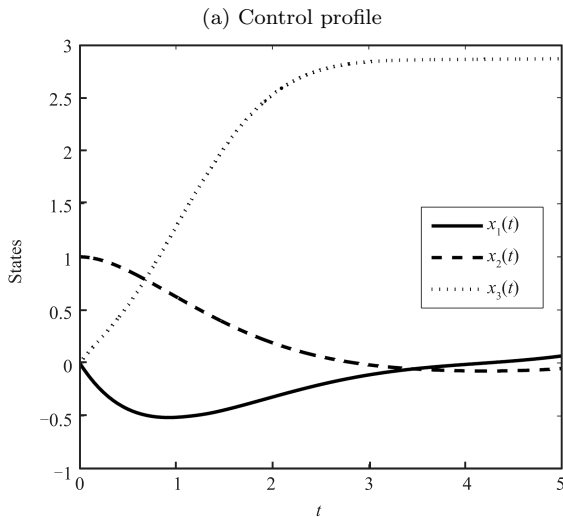
测试对象	分段数 N	目标函数 J	计算时间 (s)
Fast-CVP	10	2.908619	0.0488
	50	2.868554	0.1308
	100	2.867294	0.3372
分段线性连续 CVP	10	2.908613	2.5332
	50	2.868554	7.0167
	100	2.867294	14.9207
ACADO	10	2.926195	0.0885
	50	2.869366	0.3131
	100	2.867781	1.1185
DOTcvp	10	2.926194	3.3228
	50	2.869367	14.6602
	100	2.867782	115.7866

均少 99.1%, 又一次表明本文提出的 Fast-CVP 算法的快速和有效性.

图 4(a) 和图 4(b) 分别为 $N = 100$ 时得到的控制轨迹和状态轨迹. 从图 4(a) 可以看出, 控制变量在 $t \in [0, 5]$ 内满足关于控制变量的路径约束 $-0.3 \leq u(t) \leq 1$. 这也从另一方面反映了本文提出的算法的有效性.



(a) 控制曲线



(b) 状态曲线

(b) State profiles

图 4 Van der Pol 振荡器问题控制和状态曲线

Fig. 4 Control and state profiles for Van der Pol oscillator problem

4 结论

本文针对传统 CVP 方法提出了一种高效的快速近似算法, 采用等间隔分段线性连续的参数化方式, 将各时间分段上函数和梯度公式中的积分项简化为乘积公式, 并且把时间网格节点直接作为积分

节点, 使用固定步长数值积分方法求解微分方程组, 提高了传统 CVP 方法的逼近精度和计算效率.

本文对提出的快速近似控制向量参数化算法在两个经典的最优控制问题上进行了测试, 并将得到的结果与开发的分段线性连续 CVP 程序及国际上两个优秀的 CVP 最优控制软件 ACADO 和 DOTcvp 的结果进行了对比. 在第 3 节设置的参数条件下, 本文提出的算法具有更高的寻优效率, 分别比分段线性连续 CVP、ACADO 和 DOTcvp 平均节约 98%、43% 和 99% 的计算时间, 表明了本文所提出的快速近似算法的高效性.

由于本文提出的 Fast-CVP 是基于快速近似的策略, 适用于分段数较大的情况, 因此如何快速准确地确定参数化的段数来获得满足一定精度要求的解, 还有待更进一步的研究.

References

- Grossmann I E, Biegler L T. Future perspective on optimization. *Computers and Chemical Engineering*, 2004, **28**(8): 1193–1218
- Kirk D E. *Optimal Control Theory: An Introduction*. New York: Dover Publications, 2004.
- Peng Hai-Jun, Gao Qiang, Wu Zhi-Gang, Zhong Wan-Xie. A mixed variable variational method for optimal control problems with applications in aerospace control. *Acta Automatica Sinica*, 2011, **37**(10): 1248–1255
(彭海军, 高强, 吴志刚, 钟万颢. 求解最优控制问题的混合变量变分方法及其航天控制应用. *自动化学报*, 2011, **37**(10): 1248–1255)
- Hu Yun-Qing, Liu Xing-Gao, Xue An-Ke. A penalty method for solving inequality path constrained optimal control problems. *Acta Automatica Sinica*, 2013, **39**(12): 1996–2001
(胡云卿, 刘兴高, 薛安克. 带不等式路径约束最优控制问题的惩罚函数法. *自动化学报*, 2013, **39**(12): 1996–2001)
- Sun Yong, Zhang Mao-Rui, Liang Xiao-Ling. Improved Gauss pseudospectral method for solving nonlinear optimal control problem with complex constraints. *Acta Automatica Sinica*, 2013, **39**(5): 672–678
(孙勇, 张卯瑞, 梁晓玲. 求解含复杂约束非线性最优控制问题的改进 Gauss 伪谱法. *自动化学报*, 2013, **39**(5): 672–678)
- Houska B, Ferreau H J, Diehl M. ACADO toolkit — an open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 2011, **32**(3): 298–312
- Hirmajer T, Balsa-Canto E, Banga J. Dotcvp: Dynamic Optimization Toolbox with Control Vector Parameterization Approach for Handling Continuous and Mixed-integer Dynamic Optimization Problems. Technical Report, Instituto de Investigaciones Marinas, 2008.
- Binder T, Cruse A, Cruz Villar C A, Marquardt W. Dynamic optimization using a wavelet based adaptive control vector parameterization strategy. *Computers and Chemical Engineering*, 2000, **24**(2–7): 1201–1207

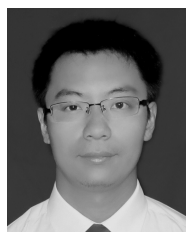
- 9 Srinivasan B, Palanki S, Bonvin D. Dynamic optimization of batch processes: I. Characterization of the nominal solution. *Computers and Chemical Engineering*, 2003, **27**(1): 1–26
- 10 Szymkat M, Korytowski A. Method of monotone structural evolution for control and state constrained optimal control problems. In: *Proceedings of the 2003 European Control Conference (ECC)*. Cambridge: University of Cambridge, 2003.
- 11 Schlegel M, Stockmann K, Binder T, Marquardt W. Dynamic optimization using adaptive control vector parameterization. *Computers and Chemical Engineering*, 2005, **29**(8): 1731–1751
- 12 Loxton R C, Teo K L, Rehbock V. Optimal control problems with multiple characteristic time points in the objective and constraints. *Automatica*, 2008, **44**(11): 2923–2929
- 13 Loxton R C, Teo K L, Rehbock V, Yiu K F C. Optimal control problems with a continuous inequality constraint on the state and the control. *Automatica*, 2009, **45**(10): 2250–2257
- 14 Hu Yun-Qing, Liu Xing-Gao. Methods to deal with control variable path constraints in dynamic optimization problems. *Acta Automatica Sinica*, 2013, **39**(4): 440–449
- 15 Assassa F, Marquardt W. Dynamic optimization using adaptive direct multiple shooting. *Computers and Chemical Engineering*, 2014, **60**: 242–259
- 16 Teo K L, Jennings L S, Lee H W J, Rehbock V. The control parameterization enhancing transform for constrained optimal control problems. *The Journal of the Australian Mathematical Society, Series B: Applied Mathematics*. 1999, **40**(3): 314–335
- 17 Vassiliadis V S. Computational Solution of Dynamic Optimization Problems with General Differential-algebraic Constraints [Ph. D. dissertation], Imperial College London (University of London), UK, 1993.
- 18 Vassiliadis V S, Canto E B, Banga J R. Second-order sensitivities of general dynamic systems with application to optimal control problems. *Chemical Engineering Science*, 1999, **54**(17): 3851–3860
- 19 Canto E B, Banga J R, Alonso A A, Vassiliadis V S. Restricted second order information for the solution of optimal control problems using control vector parameterization. *Journal of Process Control*, 2002, **12**(2): 243–255
- 20 Banga J R, Balsa-Canto E, Moles C G, Alonso A A. Dynamic optimization of bioprocesses: efficient and robust numerical strategies. *Journal of Biotechnology*, 2005, **117**(4): 407–419
- 21 Sager S. *Numerical Methods for Mixed-Integer Optimal Control Problems*. Tönning, Lübeck, Marburg: Der Andere Verlag, 2005.
- 22 Betts J T. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming (2nd Edition)*. Philadelphia, PA: SIAM, 2010.
- 23 Teo K L, Goh C J, Wong K H. *A Unified Computational Approach to Optimal Control Problems*. Essex: Longman Scientific and Technical, 1991.
- 24 Loxton R C, Lin Q, Rehbock V, Teo K L. Control parameterization for optimal control problems with continuous inequality constraints: new convergence results. *Numerical Algebra, Control, and Optimization*, 2012, **2**(3): 571–599
- 25 Biegler L T. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. New York: Society for Industrial and Applied, 2010.



mal control.)

李国栋 浙江大学控制科学与工程系博士研究生. 主要研究方向为过程最优控制. E-mail: liguodong_008@163.com

(**LI Guo-Dong** Ph. D. candidate in the Department of Control Science and Engineering, Zhejiang University. His main research interest is process opti-



胡云卿 博士, 南车株洲电力机车研究所有限公司工程师. 主要研究方向为动态优化. E-mail: huyunq@126.com

(**HU Yun-Qing** Ph. D., engineer at CSR Zhuzhou Institute Co., Ltd.. His main research interest is dynamic optimization.)



刘兴高 浙江大学控制科学与工程系教授. 主要研究方向为工业过程建模、优化与控制. 本文通信作者.

E-mail: liuxg@iipc.zju.edu.cn

(**LIU Xing-Gao** Professor in the Department of Control Science and Engineering, Zhejiang University. His research interest covers modeling, optimization and control of industrial process. Corresponding author of this paper.)