Biochemical Engineering Journal 111 (2016) 63–74

Contents lists available at ScienceDirect

Biochemical Engineering Journal

journal homepage: www.elsevier.com/locate/bej

Regular article

Novel non-uniform adaptive grid refinement control parameterization approach for biochemical processes optimization

Ping Liu [a], Guodong Li [a], Xinggao Liu [a,*], Zeyin Zhang [b]

[a] State Key Laboratory of Industry Control Technology, College of Control Science & Engineering, Zhejiang University, Hangzhou 310027, China
[b] Department of Mathematics, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 22 December 2015
Received in revised form 28 February 2016
Accepted 12 March 2016
Available online 15 March 2016

Keywords:
Bioprocess optimization
Biochemical processes
Grid refinement
Non-uniform grid
Control parameterization
Dynamic optimization

ABSTRACT

Dynamic optimization is a very important way to increase the productivity or profitability of biochemical processes. As an efficient approach for solving these biochemical dynamic optimization problems, control vector parameterization encounters the difficulty of selecting an optimal discretization level which balances the computational cost with the desired solution quality to obtain high accuracy solution. To tackle this issue, a new slope analysis is proposed to analyze the control variables and discretization time grid, results find that low slope time grid nodes have less effect on the improvement of performance index and can be regarded as unnecessary nodes, while high ones are important time points. Based on this, a novel non-uniform adaptive grid refinement control parameterization approach is therefore presented, where the slope analysis is applied to refine or to coarsen the time grid so as to obtain a suitable discretization level with a small number of control intervals. By application in three well-known biochemical optimization problems, results show that the proposed method is able to achieve similar or even better performance indexes with small numbers of control intervals and lower computational costs.

1. Introduction

Many biochemical processes are naturally dynamic systems [1], which can be described by complex nonlinear differential equations. In recent years, plenty of methods have been proposed to increase the productivity or profitability of these processes, where one of the most important ways is the model-based dynamic optimization. Many efforts have been devoted to obtain high quality optimal operating policies so as to get the best performance index [2–7].

Generally, dynamic optimization methods can be classified in three main groups: iterative dynamic programming (IDP), indirect and direct methods. Iterative dynamic programming, which applies the Bellman optimality conditions to obtain high precision solutions, is firstly presented by Luus [8] and has been used by several authors for the optimization of industrial processes [9,10]. However, Vassiliadis et al. [11] pointed out that IDP method would be computationally very costly. Indirect methods, inspired by the principles of variational calculus, are based on Pontryagin's Minimum Principle and have been used in biochemical reactors. Alternatively, direct methods transform the original problem into a nonlinear programming (NLP) problem by two strategies: complete parameterization (CP) [12] and control vector parameterization (CVP) [13,14], then the results can be obtained by solving the NLP problem. Furthermore, another existing direct discretization method is the parametrization approach [15], in which analytical expressions obtained via Pontryiagin's Minimum Principle are used in a direct approach. Once the analytical expressions for the set of optimal arcs are derived and the optimal sequence of arcs is found, only the switching points between the different arcs have to be optimized. As such this approach yields the most appropriate/adapted parametrization and the best cost value. The advantage of this approach is that it is numerically well conditioned and the parameterization is exact and in most cases parsimonious. This approach has been widely used in biochemical processes, typical application can be found in [16]. Also results comparing this parametrization to an adaptive scheme have been reported in for instance [17]. Meanwhile, stochastic approaches also apply the evolutionary algorithms (EA) to solve dynamical system optimization in recent years, the detailed work can be seen in the survey of Conway in [18]. Compared with indirect methods and stochastic algorithms, direct methods are less costly and there is no requirement to set up and solve a multipoint boundary value problem associated with Pontryagin's Maxium Principle [19]. Consequently, many literatures focus on the improvement of direct methods [1,2,5,11,20–23].

* Corresponding author.
E-mail address: lxg@zju.edu.cn (X. Liu).

http://dx.doi.org/10.1016/j.bej.2016.03.006
1369-703X/

Since the dimension of the NLP problem in CP is much greater than in CVP, this paper focuses on the CVP method. Usually, the solution quality of CVP method greatly depends on the discretization level, accurate solutions require fine discretization. However, a very fine discretization may make the discretized NLP problem become very large scale and/or ill-conditioned [19,20]. Hence, there rises a question to select the optimal discretization level which balances the computational cost due to the number of parameters with the desired approximation quality.

One possible way is the optimization of the discretization grid points. For example, Teo et al. [24–26] presented the well-known time-scaling technique to optimize the original time points, where the length of the time intervals are regarded as additional decision variables. This method is very efficient for bang–bang control problems and has been widely used. While, there is a drawback that the discretization NLP problem will be more complex and difficult to solve [20]. Another possible way is the refinement of the discretization mesh. For instance, Schlegel et al. [20] employed the fast wavelet transformation and the resulting control representations in wavelet space for grid refinement. García et al. [27] applied a refinement to all positions in the time grid by halving the step size from previous refinement iteration until the stopping criteria were fulfilled. Hadiyanto et al. [19] proposed a sensitivity-based step size refinement method to improve the product quality of baking optimization. Bittner et al. [28] presented an automatic density function-based mesh refinement algorithm (DENMRA) for aircraft trajectory optimization problem.

Since the solution of CVP method greatly depends on the chosen discretization level of time grid, a new slope analysis is proposed to analyze the relationship between the control parameters and time grid, results show that low slope points have less effect on the improvement of performance index and can be regarded as unnecessary nodes, while high ones are important points. Based on this, a novel slope analysis-based time grid refinement method is therfore proposed to select optimal time grid nodes so as to obtain high-quality solution for biochemical dynamic optimization problems with a small number of parameters and low computational cost. To limit the computational effort, the proposed method starts with a coarse discretization level for the control input, the coresponding time grid nodes are then automatically refined by subdividing the high slope nodes and/or excluding the low ones. With this refinement, optimization is continued for a selected group of input parameters in successive iterations. The proposed method is applied in three well-known biochemical dynamic optimization systems to verify the efficiency of the proposed method.

This paper is structured as follows. The statement of biochemical dynamic optimization problem is discussed in Section 2 and the uniform discretization-based control vector parameterization is presented in Section 3. Resolution of the time grid is discussed in Section 4 and the non-uniform adaptive time grid refinement procedure is presented in Section 5. Section 6 outlines the implementation of the proposed approach, and the simulation results are discussed in Section 7. Finally, the conclusion is drawn in Section 8.

2. Statement of dynamic optimization problem

A class of typical biochemical dynamic optimization problems can be stated as the following Bolza form:

2.1. Problem (P1)

$$\min_{\boldsymbol{u}(t)} J(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \Phi_0(\boldsymbol{x}(t_f), t_f) + \int_{t_0}^{t_f} L_0(\boldsymbol{u}(t), \boldsymbol{x}(t), t)dt \tag{1}$$

Subject to:

$$\begin{aligned}
&\dot{x}(t) = \boldsymbol{f}[\boldsymbol{u}(t), \boldsymbol{x}(t), t]\\
&\boldsymbol{E}[\boldsymbol{x}_0(t_0), t_0, \boldsymbol{x}(t_f), t_f] = 0\\
&\boldsymbol{C}[t, \boldsymbol{x}(t|\boldsymbol{u}), \boldsymbol{u}(t)] \geq 0\\
&t_0 \leq t \leq t_f\\
&\underline{\boldsymbol{u}} \leq \boldsymbol{u}(t) \leq \bar{\boldsymbol{u}}
\end{aligned} \tag{2}$$

where $\boldsymbol{x}(t)$ is an $(n \times 1)$ state vector, $\boldsymbol{u}(t)$ is an $(m \times 1)$ control vector, $J(\boldsymbol{x}(t), \boldsymbol{u}(t))$ is the performance index and consists of the performance of the function $\Phi_0(\boldsymbol{x}(t_f), t_f)$ at the terminal time t_f and the performance of function $\int_{t_0}^{t_f} L_0(\boldsymbol{u}(t), \boldsymbol{x}(t), t)dt$ during $t \in [t_0, t_f]$. $\boldsymbol{f}[\boldsymbol{u}(t), \boldsymbol{x}(t), t]$ is the differential equation of dynamic system. The boundary conditions are $\boldsymbol{E}[\boldsymbol{x}_0(t_0), t_0, \boldsymbol{x}(t_f), t_f]$ and the path constraints are $\boldsymbol{C}[t, \boldsymbol{x}(t|\boldsymbol{u}), \boldsymbol{u}(t)]$. $\underline{\boldsymbol{u}}$ and $\bar{\boldsymbol{u}}$ are the lower and upper bounds on $\boldsymbol{u}(t)$, respectively. The objective of this problem is to find optimal profile $\boldsymbol{u}(t)$ that minimizes $J(\boldsymbol{x}(t), \boldsymbol{u}(t))$ during $t \in [t_0, t_f]$.

3. Uniform discretization-based control vector parameterization

The idea of control vector parameterization is to discretize and approximate the control vector by a basis function with a limited number of parameters [29,30]. Then a low order B-spline function (for instance piecewise constant function) can be used to represent the control vector.

Firstly, partition the time horizon $[t_0, t_f]$ into N even stages $[t_{k-1}, t_k]$, $k = 1, 2, \cdots, N$, called time grid,

$$t_0 = t_0 < t_1 < t_2 < \ldots < t_N = t_f \tag{3}$$

where t_k is the time grid node. Next, the control vector can be expressed as

$$\boldsymbol{u}(t) \approx \tilde{\boldsymbol{u}}^N(t) = \sum_{k=1}^{N} \boldsymbol{\sigma}^k \chi_{[t_{k-1}, t_k)}(t) \tag{4}$$

where $\underline{\boldsymbol{u}} \leq \boldsymbol{\sigma}^k \leq \bar{\boldsymbol{u}}$ for $k = 1, \ldots, N$ and $\chi_{[t_{k-1}, t_k)}$ is defined as

$$\chi_{[t_{k-1}, t_k)}(t) = \begin{cases} 1, & \text{if} t \in [t_{k-1}, t_k) \\ 0, & \text{if} t \notin [t_{k-1}, t_k) \end{cases}, k = 1, 2, \ldots, N. \tag{5}$$

By using the piecewise-constant basis function, the control vector $\boldsymbol{u}(t)$ in time stage $[t_{k-1}, t_k)$ is approximated as follows:

$$\boldsymbol{u}(t) \approx \tilde{\boldsymbol{u}}^N(t) = \boldsymbol{\sigma}^k, t \in [t_{k-1}, t_k), k = 1, 2, \ldots, N. \tag{6}$$

Although Eq. (6) does not define the value of $\tilde{\boldsymbol{u}}^N(t)$ at $t = t_f$, it does not affect the evolution of the state trajectory [31].

Through this approximate method, the original Problem (P1) is transformed into a finite-dimensional NLP problem.

3.1. Problem (P2)

$$\min_{\boldsymbol{\sigma}} J = \Phi_0[\boldsymbol{x}(t_f), t_f] + \sum_{k=1}^{N} \int_{t_{k-1}}^{t_k} L_0[\boldsymbol{\sigma}^k, \boldsymbol{x}(t), t]\mathrm{d}t \tag{7}$$

Subject to:

$$\begin{aligned}
&\text{Subject to:} \dot{x}(\mathrm{t}) = \sum_{k=1}^{N} \int_{t_{k-1}}^{t_k} L_0\left[\sigma^k, x(t)t\right] \mathrm{d}t\\
&E\left[x_0(t_0), t_0 x\left(t_f\right), t_f\right] = 0\\
&\sum_{k=1}^{N} C(t, x(t|\sigma^k), \sigma^k)\chi_{(t_{k-1}, t_k)}(t) \geq 0\\
&\underline{u} \leq \sigma^k \leq \bar{u}, \mathrm{k} = 1, 2, \ldots, N
\end{aligned} \tag{8}$$