A Quasi-Sequential Approach to Large-Scale Dynamic Optimization Problems

Weirong Hong and Shuqing Wang
National Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China

Pu Li
Institute of Automation and Systems Engineering, Technical University of Ilmenau, Ilmenau 98684, Germany

Günter Wozny
Institute of Process and Plant Technology, Technical University of Berlin, KWT9, Berlin 10623, Germany

Lorenz T. Biegler
Dept. of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

DOI 10.1002/aic.10625
Published online September 2, 2005 in Wiley InterScience (www.interscience.wiley.com).

A novel sequential approach for solving dynamic optimization problems containing path constraints on state variables is presented and its performance analyzed. As in the simultaneous approach, we discretize both state and control variables using collocation on finite elements, so that path constraints can be guaranteed inside each element. The state variables are solved in a manner similar to that in the sequential approach; this eliminates the discretized differential-algebraic equations and state variables, so that the problem is reduced to a smaller problem only with inequality constraints and control variables. Therefore, it possesses advantages of both the simultaneous and the sequential approach. Furthermore, the elimination of the equality constraints substantially simplifies the line search problem and thus larger steps can be taken by successive quadratic programming (SQP) toward the optimum. We call this dynamic optimization method a quasi-sequential approach. We compare this new approach with the simultaneous approach in terms of computational cost and by analyzing the solution path. A highly nonlinear reactor control and the optimal operation of a heat-integrated column system are used to demonstrate the effectiveness of this approach. As a result, it can be concluded that this quasi-sequential approach is well suited for solving highly nonlinear large-scale optimal control problems. © 2005 American Institute of Chemical Engineers
AIChE J, 52: 255–268, 2006
Keywords: dynamic optimization, quasi-sequential approach, nonlinear programming, collocation

Introduction

The past decade has been marked by increasingly widespread applications of large-scale dynamic optimization in process industries. Typical examples include finding an optimal trajectory to perform state transition tasks (such as start-up and shutdown, product and feed switchover, and batch processing) and to implement a nonlinear model predictive control. These tasks require the solution of a dynamic optimization problem based on a nonlinear process model. Thus the development of numerical approaches to dynamic optimization problems has made impressive progress.[1] Efficient approaches

Correspondence concerning this article should be addressed to P. Li at pu.li@tu-ilmenau.de.

have been developed for solving large-scale problems modeled with differential-algebraic equations (DAEs). Sequential approaches were proposed and found wide applications in the 1990s.[2-4] In recent years, the increased interest in dynamic optimization has also concentrated on simultaneous approaches.[5-8] For practical applications, one needs to trade off the computational advantages of both approaches to solve a specific problem. Often, this choice is problem dependent.

In this article, we present and analyze a hybrid approach that is suitable for solving large-scale dynamic optimization problems with state path constraints; this combines advantages of both the simultaneous and sequential strategies. In this approach, both the states and the controls are discretized with collocation on finite elements, so that path constraints remain feasible inside each element. On the other hand, a simulation step is used to eliminate equality constraints and state variables, so that it is reduced to a small problem only with inequality constraints and control variables. Therefore, it possesses the advantages of both the simultaneous and the sequential approach. Because this approach requires solution of the model equations at each iterate, we call it a *quasi-sequential approach*. Although this approach has been proposed and used for dynamic optimization of distillation processes,[9-11] a detailed analysis of this approach has not been made until now. Herein, we derive the approach from a simultaneous perspective and analyze its performance. We then compare this approach with the simultaneous approach in terms of computational cost. The comparison between the two approaches is also made by analyzing the solution paths.

Because of the elimination of the equality constraints, this quasi-sequential approach is well suited for solving large-scale optimization problems. Moreover, the elimination of the equalities appreciably relieves the difficulty of the line search. In other words, this approach is very suitable for solving highly nonlinear optimization problems in which the magnitude of equality constraints varies significantly during the course of solution. On the other hand, solution of the model equations does require more computational effort. Therefore this approach will be more efficient when the constraints are relatively easy to converge with Newton's method. In particular, we have found that the quasi-sequential approach is appropriate for optimal control problems where an operating point is to be maintained and disturbances need to be rejected. As the effort to converge the equality constraints increases, the advantage of the quasi-sequential approach will disappear.

In the second section we describe the major contributions leading to the recent development of dynamic optimization approaches. The third section introduces the quasi-sequential approach and presents a comparison of the quasi-sequential and the simultaneous approaches. The quasi-sequential approach on two process optimization problems is then considered. Finally, a set of conclusions and future research perspectives are outlined.

Approaches for Dynamic Optimization

The constrained dynamic optimization problem of interest in this article can be stated as follows

$$\min_{x,u} \varphi(z, u) \tag{1}$$

$$\text{s.t.} \quad F(\dot{z}, z, u, t) = 0 \tag{2}$$

$$z_{\min} \leq z(t) \leq z_{\max} \tag{3}$$

$$u_{\min} \leq u(t) \leq u_{\max} \tag{4}$$

$$z(0) = z_0 \tag{5}$$

In this formulation, $z(t)$ are state (dependent) variables and $u(t)$ are control (independent) variables. Equation 2 defines a general system of DAEs; Eqs. 3 and 4 the path constraints on the state variables and control variables, respectively; and Eq. 5 the initial condition of the state variables. Note that any inequality constraint functions have been reformulated through the addition of slack variables with lower bounds of zero, which are now added to the state variable vector. Correspondingly, the reformulated inequality constraint functions are added to the system of DAEs.

There is extensive literature—grouped into two general classes—on approaches to such DAE optimization problems: indirect and direct approaches (see van Schijndel and Pistikopoulos[12]). The indirect approaches (or variational approaches) focus on obtaining a solution to the necessary conditions of optimality that take the form of a two-point boundary value problem. In the direct approach the infinite-dimensional dynamic problem is transformed into a finite-dimensional nonlinear programming (NLP) problem. Because of their easy-to-handle properties direct approaches have gained wide application. The direct approaches recently developed can be further categorized into two solution approaches: the sequential and the simultaneous approach. We analyze these approaches in more detail as follows.

The sequential approach

The sequential approach decomposes the whole system into the control and state spaces. Only the control variables are discretized and remain as degrees of freedom in the NLP solver. Thus this technique is also known as a *control parameterization approach*.[13] Given the initial conditions and a set of control parameters, the process model requires a solution with a DAE solver at each NLP iteration. This solution eliminates the DAEs and provides the value of the objective function, which is used by the NLP solver to find optimal coefficient values in the next iterate. So the number of optimization variables for the NLP solver is reduced dramatically. In this formulation, the control variables are represented as piecewise polynomials or piecewise constant functions[3,4] and optimization is performed with respect to the polynomial coefficients. The gradients of the objective function with respect to the control variables are calculated from sensitivity equations of the DAE system.[14] The sequential approach follows a feasible path, that is, at each NLP iterate the DAE system is solved. The main disadvantage of the sequential approach is that difficulties will arise when there are path constraints on the state variables because the state variables are not directly included in the NLP solver. In addition, the solution step might be too expensive to converge or even fail at an intermediate trial point, particularly while integrating an unstable system.[15]

A method for solving dynamic optimization problems that

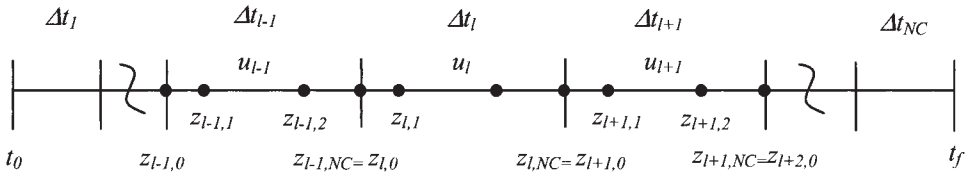

Figure 1. Radau collocation on finite elements (*NC* = 3).

contain path constraints on the state variables was studied by Feehery.[16] The idea is to use an algorithm for constrained dynamic simulation so that any admissible combination of the control parameters produces an initial value problem (IVP) that is feasible with respect to the path constraints. It results in a hybrid discrete/continuous dynamic optimization problem in which the differential index may fluctuate along the solution trajectory. Meanwhile, it leads to a high-index DAE system and their related sensitivities. The method of dummy derivatives can be used to solve the high-index DAE system and their sensitivities. However, this approach may require solution of second-order sensitivity equations for some problems.[17]

We also mention the related *multiple-shooting approach*, which offers additional flexibility over the sequential approach. It serves as a bridge between sequential approaches and simultaneous approaches that are based on a complete discretization of the state and control variables. Here the time domain is partitioned into smaller time elements and a DAE solver to the process model is applied in each element.[18,19] Control variables are treated in the same manner as in the sequential approach. Moreover, to obtain gradient information, sensitivities are obtained for both the control variables as well as the initial conditions of the states in each element. Finally, equality constraints are added to the nonlinear program to link the elements and ensure that the states are continuous across each element. With this approach, inequality constraints for states and controls can be imposed directly at the grid points. For piecewise constant or linear controls this approximation is accurate enough, although path constraints for the states may be difficult to satisfy between elements.

The simultaneous approach

In the simultaneous approach the dynamic system is completely discretized and both the state and the control variables on the discrete time points are included in the optimization problem. This leads to a large-scale NLP problem,[5,6,20] but the DAE system is solved only once at the optimal point. Therefore intermediate solutions that may not exist or require excessive computational effort can be avoided. Moreover, simultaneous approaches have advantages for problems with unstable modes.[6]

With the simultaneous approach, the time period (from t_0 to t_f) is divided into the intervals shown in Figure 1 and the DAE optimization problem expressed in Eqs. 1–5 is converted into an NLP problem by approximating state profiles with collocation on finite elements (time intervals). Orthogonal collocation applies a polynomial approximation to the differential equations and requires satisfaction of the equations at the discrete collocation points. The polynomial solution is thus a continuous function of t that is often as accurate as a finite-difference solution using many more points.[21] In this report, the control variables are represented, for ease of presentation, as piecewise constants in each element.

In Figure 1, *NC* is the number of collocation points in one element and *NL* is the number of elements in the time period, respectively; z denotes the discretized state variables; and u the control variables. The state variables will be represented here as a linear combination of a set of orthogonal polynomials, using a Lagrange polynomial basis

$$z_\ell(t) = \sum_{j=0}^{NC} \ell_j(t) z_{\ell,j} = \sum_{j=0}^{NC} \left(\prod_{\substack{i=0 \\ i \neq j}}^{NC} \frac{t - t_{\ell,i}}{t_{\ell,j} - t_{\ell,i}} \right) z_{\ell,j}$$
$$\ell = 1, \ldots, NL,\ i = 1, \ldots, NC \tag{6}$$

With this representation the state variables and differential terms in Eq. 2 on the collocation points of a time interval will be

$$z_\ell(t_{\ell,i}) = \sum_{j=0}^{NC} \ell_j(t_{\ell,i}) z_{\ell,j} = z_{\ell,i} \qquad i = 1, NC \tag{7}$$

$$\frac{dz_\ell(t_{\ell,i})}{dt} = \sum_{j=0}^{NC} \frac{d\ell_j(t_{\ell,i})}{dt} z_{\ell,j} \qquad i = 1, NC \tag{8}$$

For the continuity of the state variables between two intervals, the last collocation point of an element is used as the initial point for the next element rather than an extrapolation of the polynomials. This *Radau collocation* improves the convergence of the computation because on the collocation points the model equations are satisfied.[9] For a clearer presentation, let z and u denote the components of the discretized state and control variables, with z an m-vector and u an $(n - m)$-vector. Thus, the dynamic problem has n variables and m constraints (note that the number of equality constraints is equal to the number of state variables).

Substituting Eqs. 6–8 into Eqs. 1–5 leads to the following NLP problem

$$\min_{x \in \Re^n} f(x)$$
$$\text{s.t.} \quad c(x) = 0$$
$$x_U \geq x \geq x_L \tag{9}$$

where $x^T = (z^T u^T)$ and $c(x)$ represent, respectively, the vector of all discretized variables (both controls and states) and equalities in Eqs. 6–8. We note that $x \in \Re^n$, $f: \Re^n \rightarrow \Re$, and c :

$\Re^n \rightarrow \Re^m$ are smooth functions. For ease of derivation we consider double-bounded inequalities on x, without loss of generality. Problem 9 can be solved with a number of methods, although it may be nonconvex and thus contains local solutions. We choose to apply local NLP solvers, although recent work related to the determination of globally optimal solutions[22,23] may be considered. We note that, although such algorithms provide a guarantee of finding global solutions, they can be significantly more expensive than local solvers.

In particular, we consider the solution of Eq. 9 with two approaches: an active set successive quadratic programming (SQP) strategy and a barrier SQP strategy. The simultaneous strategy causes the NLP to grow with the size of the DAE system, and solution of such a large NLP problem requires efficient techniques and careful initialization of the optimization variables. Here the reduced-Hessian successive quadratic programming method (rSQP) is efficient for solving DAE optimization problems by decoupling the search direction into range and null spaces and solving a smaller quadratic programming (QP) subproblem at each iterate, especially when the dimension of state variables is much larger than that of control variables. We consider two forms of this method next.

Active set SQP

At each iteration k, a search direction d_k is obtained by solving a QP subproblem:

$$\min_{d\in\Re^n} g(x_k)^T d_k + \frac{1}{2} d_K^T W(x_k) d_k$$

$$\text{s.t.} \quad c(x_k) + A(x_k)^T d_k = 0$$

$$x_U \geq x_k + d_k \geq x_L \tag{10}$$

where $g(x)$ denotes the gradient of $f(x)$; W denotes the Hessian (with respect to x) of the Lagrangian function $L = f(x) + c(x)^T\lambda + (\nu_U - \nu_L)^T x$; and $A(x)$ stands for the $n \times m$ matrix of constraint gradients. Here λ, ν_U, and ν_L are Lagrangian multipliers corresponding to the equalities and inequalities, repectively. To solve problem 10 the variables are partitioned into m dependent and $n - m$ independent variables. The partition of A takes the form

$$A_k^T = [C_k \quad N_k] \tag{11}$$

where the $m \times m$ basis matrix C_k is nonsingular. By defining the null and range space basis matrices

$$Q_k = \begin{bmatrix} -C_k^{-1}N_k \\ I \end{bmatrix} \qquad R_k = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{12}$$

the search direction can be written as

$$d_k = R_k d_R + Q_k d_Q \tag{13}$$

Note that the matrix Q_k satisfies $A_k^T Q_k = 0$ and is therefore a null-space basis matrix for A_k^T. The range-space direction d_R is now determined by solving

$$d_R = -C_k^{-1} c_k \tag{14}$$

and the null-space direction d_Q is obtained from the following reduced QP subproblem

$$\min_{d_Q\in\Re^{n-m}} (Q_k^T g_k + Q_k^T W_k R_k d_R)^T d_Q + \frac{1}{2} d_Q^T (Q_k^T W_k Q_k) d_Q$$

$$\text{s.t.} \quad x_U \geq x_k + R_k d_R + Q_k d_Q \geq x_L \tag{15}$$

where we have omitted constant terms involving d_R. Assuming that $Q_k^T W_k Q_k$ is positive definite, the solution of Eq. 15 is then

$$d_Q = -(Q_k^T W_k Q_k)^{-1}[Q_k^T g_k + w_k + Q_k^T(\nu_{U,k} - \nu_{L,k})] \tag{16}$$

where $\nu_{U,k}$ and $\nu_{L,k}$ are the multipliers corresponding to the inequalities in Eq. 15 and $w_k = Q_k^T W_k R_k d_R$. This determines the search direction for the NLP solver. It is practical to approximate the reduced Hessian $Q_k^T W_k Q_k$ by a positive definite quasi-Newton matrix B_k [that is, $(n - m) \times (n - m)$ matrix] that can be updated by the BFGS algorithm. Approximating the cross term w_k, can be done in three ways:

(1) Often $w_k = 0$ and this corresponds to a classical reduced space approach with a two-step superlinear convergence.

(2) Biegler et al.[24] describe a method that updates a quasi-Newton approximation, $S_k = Q_k^T W_k$, and defines $w_k = S_k R_k d_R$.

(3) A method to update w_k through finite differences of $Q^T g$ was also proposed. The detailed algorithm can be found in Biegler et al.[24]

Barrier SQP method

Solution of the QP problem 15 and the combinatorial complexity of finding the active set may be expensive. Instead, we now replace the inequalities by a logarithmic barrier term that is added to the objective function to give

$$\min_{x\in\Re^n} \varphi_\mu(x) = f(x) - \mu\left\{\sum_{i=1}^{n} \ln[x^{(i)} - x_L] + \sum_{i=1}^{n} \ln[x_U - x^{(i)}]\right\} \tag{17}$$

$$\text{s.t.} \quad c(x) = 0 \tag{18}$$

with the barrier parameter $\mu > 0$. Here, $x^{(i)}$ denotes the ith component of the vector x. In the primal–dual approach, multiplier estimates, ν_L and ν_U, are introduced and the optimality conditions for Eqs. 17 and 18 are written in the following form

$$\nabla f(x) + A(x)\lambda - \nu_L + \nu_U = 0 \tag{19}$$

$$(X - X_L)V_L - \mu e = 0 \tag{20}$$

$$(X_U - X)V_U - \mu e = 0 \tag{21}$$

$$c(x) = 0 \tag{22}$$

where $A(x) = \nabla c(x)$; λ is the vector of Lagrange multipliers for the equality constraints (Eq. 18); $e = [1, \ldots, 1]^T$; and X and V are diagonal matrices with x and ν on their diagonals, respectively. Newton's method may be used to solve this system of nonlinear Eqs. 19–22. Then the search direction $(d_k^x, d_k^\lambda, d_k^{\nu L}, d_k^{\nu U})$ at an iterate $(x_k, \lambda_k, \nu_{L,k}, \nu_{U,k})$ is obtained as a solution of the linearization of Eqs. 19–22, that is

$$\begin{bmatrix} W_k & A_k & -I & I \\ A_k^T & 0 & 0 & 0 \\ V_{L,k} & 0 & X - X_L & 0 \\ V_{U,k} & 0 & 0 & X_U - X \end{bmatrix} \begin{bmatrix} d_k^x \\ d_k^\lambda \\ d_k^{\nu L} \\ d_k^{\nu U} \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) + A_k \lambda_k - \nu_{L,k} + \nu_{U,k} \\ c_k \\ (X_k - X_L)\nu_{L,k} - \mu e \\ (X_U - X_k)\nu_{U,k} - \mu e \end{bmatrix} \quad (23)$$

where $W_k = \nabla^2_{xx} L(x_k, \lambda_k, \nu_{L,k}, \nu_{U,k})$. Eliminating $d_k^{\nu L}$ and $d_k^{\nu U}$ leads to the following system:

$$\begin{bmatrix} W_k + \Sigma_k & A_k \\ A_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k^x \\ \lambda_k + d_k^\lambda \end{pmatrix} = - \begin{pmatrix} \nabla \varphi_\mu(x_k) \\ c_k \end{pmatrix} \quad (24)$$

with $\Sigma_k = (X_U - X_k)^{-1} V_{U,k} + (X_k - X_L)^{-1} V_{L,k}$; and $d_k^{\nu L} = (X_k - X_L)^{-1}(\mu_k e - V_{L,k} d_k^x) - \nu_{L,k}$ and $d_k^{\nu U} = (X_U - X_k)^{-1}(\mu_k e - V_{U,k} d_k^x) - \nu_{U,k}$ as the search directions for the multiplier estimates.

Obtaining d_k^x from Eq. 24 is equivalent to solving the following quadratic problem:

$$\min_{d \in \Re^n} \nabla \varphi(x_k)^T d^x + \frac{1}{2} (d^x)^T (W_k + \Sigma_k) d^x \quad (25)$$

$$\text{s.t.} \quad c(x_k) + A(x_k)^T d^x = 0 \quad (26)$$

if the matrix $W_k + \Sigma_k$ is positive definite in the null space of A_k^T. The problem described by Eqs. 25 and 26 is similar to the QP problem 10 and the overall primal step can be partitioned into a range and a null space, using Eq. 13. However, now the reduced space QP is unconstrained and its solution can directly be computed as

$$d_Q = -[Q_k^T (W_k + \Sigma_k) Q_k]^{-1} [Q_k^T \nabla \varphi_\mu(x_k) + w_k] = -[B_k + Q_k^T \Sigma_k Q_k]^{-1} [Q_k^T \nabla \varphi_\mu(x_k) + w_k] \quad (27)$$

with

$$w_k = Q_k^T (W_k + \Sigma_k) R_k d_R \quad (28)$$

To summarize, the simultaneous approach in terms of rSQP uses a decomposition of the discretized DAEs and the solution of the NLP problem is performed in the reduced space of the independent variables. It is thus well suited for large-scale problems with relatively few degrees of freedom. In addition, it introduces the primal–dual interior-point method to solve the inequality constrained optimization problems. When a large number of bounds are active, the barrier method is more efficient than the active-set methods because it eliminates the combinatorial problem of selecting an active set.[8]

The Quasi-Sequential Approach

The quasi-sequential approach presented here is based on a complete discretization of the state and control variables with collocation on finite elements such as the simultaneous approach, and it eliminates the state variables and the DAEs in the same manner as in the sequential approach. Only the control variables and the inequalities will be handled in the NLP solver. By using the collocation method, the differential equations of the DAE system are converted to a set of algebraic equations. The DAE model is solved successively in each element at each NLP iterate, and the sensitivities of the state variables with respect to control variables are computed in parallel with the DAE solution. Thus this approach possesses the advantage of the sequential approach. On the other hand, both equality and inequality constraints are imposed at the collocation points in the NLP solver, so path constraints for state variables will be satisfied at every collocation point. Therefore, this quasi-sequential approach overcomes the difficulty in the sequential approach of satisfying state path constraints more frequently. A number of previous studies have also considered collocation along with the quasi-sequential optimization approach.[2,9,25,26] Also, a similar approach was developed by Pytlak[27] using various implicit Runge–Kutta discretizations. In this study, we demonstrate a strong link with the simultaneous approach along with a detailed treatment of the NLP and decomposition steps of both approaches.

The quasi-sequential approach uses a two-layer strategy. Only the control variables and inequalities constraints are included in the optimization layer, whereas the state variables are solved through solution of the model Eq. 18 in the simulation layer. Again we define the partition in Eq. 11 by

$$A(x)^T = [C(x) \mid N(x)] = [\nabla_z c^T \mid \nabla_u c^T] \quad (29)$$

At iteration k in the optimization layer, the Newton algorithm is usually used to solve $c(z, u_k) = 0$ for z. The model equations are linearized through the Taylor expansion and Newton steps are generated by

$$\Delta z_{kk} = -\left(\frac{\partial c}{\partial z}\right)^{-1} c(z_{kk}, u_k) = -C^{-1}(z_{kk}, u_k) c(z_{kk}, u_k) \quad (30)$$

Through $z_{kk+1} = z_{kk} + \Delta z_{kk}$ (where kk denotes the Newton iteration), we update the values of the state variables and expect to converge to the solution z_k^* that satisfies $c(z_k^*, u_k) = 0$, where we define $z(u_k) = z_k^*$. After the solution of model equations, the problem is now reduced to the form

$$\min_{u \in \Re^{n-m}} f[z(u), u]$$
$$\text{s.t.} \quad z_U \geq z(u) \geq z_L$$
$$u_U \geq u \geq u_L \quad (31)$$

This problem consists only of control variables as well as the inequality constraints that are enforced at each collocation

point. We use a standard SQP solver to solve the reduced optimization problem 31. The following QP subproblem will be solved at each iteration k:

$$\min_{du\in\Re^{n-m}} \left(\frac{df}{du}\right)_k^T \Delta u_k + \frac{1}{2}\Delta u_k^T \tilde{W}_k \Delta u_k$$

$$\text{s.t.} \quad z_U \geq z(u_k) + \left(\frac{dz}{du}\right)_k^T \Delta u_k \geq z_L$$

$$u_U \geq u_k + \Delta u_k \geq u_L \tag{32}$$

where df/du is the reduced gradient of $f[x(u), u]$ and $\tilde{W}$ denotes the reduced Hessian of the Lagrangian function. Linearization at a feasible point, $C(z_k^*, u_k)dz + N(z_k^*, u_k)du = 0$, leads to $(dz/du)^T = -C^{-1}N$. Moreover, the reduced gradient and reduced Hessian can be computed from

$$\left.\frac{df}{du}\right|_{c(x)=0} = \nabla_u f + \frac{dz}{du}\nabla_z f = Q^T g = Q^T \nabla L(x, \lambda, \nu)$$

$$= \left.\frac{dL(x, \lambda, \nu)}{du}\right|_{c(x)=0} \tag{33}$$

and

$$\tilde{W} = \frac{d}{du}\left[\frac{dL(x, \lambda, \nu)}{du}\right] = Q^T W Q + \sum_i \left[\frac{dQ^{(i)}}{du}\right]^T \nabla L(x, \lambda, \nu) \tag{34}$$

where $Q^{(i)}$ is the ith column of Q. Substituting this information into Eq. 32 and noting that the last term in Eq. 34 vanishes at the optimum, we have the QP

$$\min_{\Delta u\in\Re^{n-m}} (Q_k^T g_k)^T \Delta u + \frac{1}{2}\Delta u^T (Q_k^T W_k Q_k)\Delta u$$

$$\text{s.t.} \quad x_U \geq x_k + Q_k \Delta u \geq x_L \tag{35}$$

As in Eq. 15 the reduced Hessian can be constructed by a BFGS update based on changes in the reduced gradient $Q_k^T g_k$ at successive iterations. The inequality constrained problem 31 can be solved with a standard NLP solver that forms the QP Eq. 35.[28] Note that Eq. 35 can also be derived from Eq. 15 by setting $d_R = 0$ and $w_k = 0$. It is interesting to note the difference in approximation of the Hessian between the two approaches. The quasi-sequential uses the form of Eq. 34, which is the second-order "total" differential of the Lagrangian. This will be updated by BFGS. So in fact the quadratic term in Eq. 35 should be $\tilde{W}$ for an exact computation of $dQ^{(i)}/du$. On the other hand, as indicated in Eq. 15, the simultaneous approach uses $Q^T W Q$ as the Hessian that is the second-order "partial" differential of the Lagrangian. Compared with Eq. 35, however, there is a linear term in Eq. 15 that is not in Eq. 35. The linear term is needed to give one-step superlinear convergence. Without it, the convergence rate is two-step and there are a few examples in the literature that illustrate the two-step behavior.[29,30] Interestingly, the quasi-sequential approach does not have this problem and the convergence rate should be one-step superlinear.

Because the equality constraints remain converged in this NLP algorithm, one can claim the following algorithmic advantages over the rSQP method:

- The correction term w_k need not be calculated.
- The line search does not need to consider the infeasibility of the equality constraints. As we shall see below, the influence of infeasibilities, especially on poorly scaled nonlinear constraints, can lead to very small step sizes in the line search. An example of this is the Maratos effect.[34]
- Quasi-Newton updates are constructed only at feasible points and therefore are likely to use better curvature information in the construction of the reduced Hessian and therefore require fewer SQP iterations.

On the other hand, repeated convergence of the equality constraints leads to more computational effort than with rSQP and the quasi-sequential method fails if convergence cannot be achieved at intermediate points.

Sensitivity computation

To calculate $C^{-1}N$, we use an elemental decomposition. Because the DAEs are solved successively from element to element at each NLP iterate, the sensitivities of the state variables with respect to control variables can be computed in parallel with the DAE solution. Here we consider the sensitivities of state variables with respect to their initial values in one element. Consider the discretized DAE system in the kth iteration of NLP and in the ℓth element, the model equations at the collocation points can be described as

$$c_\ell(z_{\ell,0}^k, z_\ell^k, u_\ell^k) = 0 \qquad \ell = 1, NL \tag{36}$$

where $z_{\ell,0}^k$ is the initial value of z_ℓ^k of the element. Through the first-order Taylor expansion of Eq. 36 we obtain

$$\nabla_{z_{\ell,0}} c_\ell^T \Delta z_{\ell,0} + \nabla_{z_\ell} c_\ell^T \Delta z_\ell + \nabla_{u_\ell} c_\ell^T \Delta u_\ell = 0 \tag{37}$$

or

$$C_{\ell 0}\Delta z_{\ell,0} + C_\ell \Delta z_\ell + N_\ell \Delta u_\ell = 0$$

From Eq. 37 we can determine the sensitivities of the state variables with respect to control variables and initial states, respectively

$$\frac{dz_\ell}{du_\ell} = -\nabla_{z_\ell}^{-T} c_\ell^T \cdot \nabla_{u_\ell} c_\ell = -C_\ell^{-1} N_\ell \tag{38}$$

$$\frac{dz_\ell}{dz_{\ell,0}} = -\nabla_{z_\ell}^{-T} c_\ell \cdot \nabla_{z_{\ell,0}} c_\ell^T = -C_\ell^{-1} C_{\ell 0} \tag{39}$$

Taking the last collocation point of element ℓ as the initial point of $\ell + 1$, we have $z_{\ell+1,0}^k = D z_\ell^k$. Here D is the mapping matrix from z_ℓ^k to $z_{\ell+1,0}^k$. Then the constraint gradients of the discretized equations can be written as

$$A^T = \begin{bmatrix} \begin{bmatrix} C_1 & 0 \\ -D & C_{20} \end{bmatrix} & & & & \vdots \begin{bmatrix} N_1 \\ 0 \end{bmatrix} & \\ & \begin{bmatrix} C_2 & 0 \\ -D & C_{30} \end{bmatrix} & & & \vdots & \begin{bmatrix} N_2 \\ 0 \end{bmatrix} \\ & & \begin{bmatrix} C_3 & 0 \\ -D & C_{40} \end{bmatrix} & & \vdots & & \begin{bmatrix} N_3 \\ 0 \end{bmatrix} \\ & & & \ddots \quad \ddots & \vdots & & & \ddots \\ & & & & \begin{bmatrix} C_{NL} & 0 \\ -D & C_f \end{bmatrix} \vdots & & & \begin{bmatrix} N_{NL} \\ 0 \end{bmatrix} \end{bmatrix} = [C \mid N] \tag{40}$$

Note that the C matrix in Eq. 40 has a block diagonal form. By premultiplying each row of A^T by the inverse of C_i, we can develop a forward Gaussian elimination that allows us to calculate $C^{-1}c$, $C^{-T}g^c$, and $C^{-1}N$. Here we use the subroutine DLINRG of IMSL MATH[31] to obtain LU factorizations of C_i. For instance, at iteration k, $C^{-1}N$ is a lower triangular matrix with the following form

$$-C^{-1}N = \begin{bmatrix} s_1 & & & & & \\ s_{2,1} & s_2 & & & & \\ \vdots & \ddots & \ddots & & & \\ s_{\ell,1} & & & s_\ell & & \\ \vdots & & s_{i,j} & \ddots & \ddots & \\ s_{NL,1} & s_{NL,2} & \cdots & s_{NL,\ell} & \cdots & s_{NL} \\ u_1 & & u_\ell & & u_{NL} & \\ \text{interval 1} & & \text{interval } \ell & & \text{interval } NL & \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} z_1, \text{ interval 1} \\ \\ z_\ell, \text{ interval } \ell \\ \\ z_{NL}, \text{ interval } NL \end{matrix} \tag{41}$$

Using Eqs. 38–40 we see that $s_\ell = (dz_\ell^k/du_\ell^k)^T = -C_\ell^{-1}N_\ell$ and

$$s_{i,j} = \left(\frac{dz_i^k}{du_j^k}\right)^T = -C_i^{-1}\hat{C}_i(-C_{i-1}^{-1}\hat{C}_{i-1})\cdots(-C_j^{-1}N_j)$$

$$= \left(\frac{dz_i^k}{dz_{0,i}^k}\right)^T \left(\frac{dz_{i-1}^k}{dz_{0,i-1}^k}\right)^T \cdots \left(\frac{dz_j^k}{du_j^k}\right)^T$$

$$i, j = 1, NL, i > j$$

where $\hat{C}_i = C_{i0}D$. Because of enforcing state profile continuity, that is, the last collocation point of state variables is used as the starting point of the next element, the sensitivities can be transferred, using the chain rule, from element to element, and the sensitivity matrix $C^{-1}N$ is easily calculated. The algorithm of the quasi-sequential approach can be described as follows:

(1) Given a fixed number of elements, discretize the control variables with piecewise constant, and state variables with the collocation method.

(2) Provide an initial (guessed) value of control variables, u_0, and bounds for all variables.

(3) Solve equality constraints (model equations) from Eq. 18.

(4) Evaluate the sensitivities with Eq. 41.

(5) Evaluate $f(u_k)$, $\nabla f(u_k)$ with Eq. 33.

(6) Call the NLP solver to solve the QP Eq. 35 and receive the search direction Δu_k.

(7) If the convergence tolerance is satisfied, STOP; otherwise, go back to step (3).

It can be seen that the implementation of this approach is straightforward. As with the sequential approach, one needs a standard NLP solver for the solution of model equations, and an algorithm for sensitivity computation. It should be noted that this sensitivity computation is not appropriate for unstable systems. To handle dynamic instabilities, a permutation of the columns in Eq. 40 is required to find a well-conditioned basis matrix C. This approach is developed in Cervantes et al.[6]

We now consider some conceptual differences with this quasi-sequential approach and the simultaneous approach.

Comparison of Solution Paths

In this subsection we analyze the difference between the simultaneous and the quasi-sequential approaches, and provide an example with a graphical interpretation. Both the quasi-sequential and simultaneous approaches have similarly sized subproblems that are derived from similar information. Because both methods are Newton-like, one would expect them to require about the same number of iterations *as long as full steps are taken in the line search.* In fact, we will see that the line search is a distinguishing feature between the quasi-sequential and simultaneous methods for the following reasons.

First, for the quasi-sequential method, the line search is evaluated only at feasible points and requires a reduction only in the objective function. On the other hand, with the simultaneous approach, the line search requires a trade-off between reducing infeasibility and the objective function. This is done either through a penalty-based merit function or through the filter method.[32] Balancing these trade-offs is not always clear, especially with nonlinear and poorly scaled constraints, and therefore small line search steps and slow convergence can result. Finally, with the barrier NLP approach the line search step size is first determined to keep the variables within the bounds. This can lead to a further truncation of the line search step and may lead to even slower convergence. An example of this can be seen in the Maratos effect.[34]

Because of these characteristics, the quasi-sequential approach does not suffer the line search difficulties associated with highly nonlinear and poorly scaled constraints. However, this advantage comes at the expense of additional Newton iterations for the constraints; assessing these benefits is therefore often dependent on the application. Nevertheless, if the

constraint functions are easy to converge with Newton iterations, then the quasi-sequential approach may have a significant advantage over the simultaneous approach.

We illustrate this behavior with the following example. Consider a two-dimensional equality constrained optimization problem:

$$\min f(x, y) = (x - 5)^2 + y^2 \tag{42}$$

$$\text{s.t.} \quad c(x, y) = y - x^3 = 0 \tag{43}$$

In the sequential approach, the variable x is regarded as the control variable. In the simultaneous approach, likewise, the variable x is regarded as the independent variable that occupies the null space. We solved the problem by both the simultaneous approach according to the algorithm described in Biegler et al.[7] and the quasi-sequential approach. The initial point is set at $(-1.0, -1.0)$. The solution point should be at (1.056, 1.178). The solution paths of both approaches are presented in Figures 2a and 2b.

From Figure 2a, because the sequential approach is a feasible-path method, the solution points ("best" vertex) always move on the equality constraint curve (solid line). The simultaneous approach is an infeasible path method; the points will be away from the curve of the equation. In Figure 2b, surface A shows the value of the equation $c = y - x^3$; the plane B intersects surface A when the equation $c = y - x^3 = 0$. The search space of the sequential approach is limited to the curve that is the intersection line between A and B, whereas the search space of the simultaneous approach is on surface A. Therefore the value of $c(x, y)$ strongly affects the solution path in the simultaneous approach. The step size of the iterations by the quasi-sequential and simultaneous approaches is shown in Figure 3. In Figure 3, the quasi-sequential approach is called QS for short and the program package IPOPT[33] is used for the simultaneous approach.

To further illustrate the consequences of the solution path, we consider another simple example problem

$$\min f(x, y) = \frac{1}{2}(x^2 + y^2) \tag{44}$$

$$\text{s.t.} \quad c(x, y) = \frac{1}{xy} - b = 0 \qquad x, y \geq 0 \tag{45}$$

Starting from the point $(1, 1/b)$, which satisfies the constraint 45 the computational performance for this problem is shown in Table 1. Because the computation time is negligible for this simple problem, only the iteration numbers required are given in Table 1.

In Table 1, IPOPT[33] is an interior point simultaneous algorithm for large-scale nonlinear optimization, and its package is released under the Common Public License (CPL). When implementing the IPOPT package, we choose the reduced space option to generate the search direction, follow the Augmented Lagrangian line-search method, and use symmetric rank one update (SR1) to approximate the reduced Hessian. Full Space SQP, which is an active set QP solver from Schittkowski,[28] means that the problem is solved directly by the SQP subroutine (without any decomposition). Table 1 shows that the quasi-sequential approach takes far fewer iterates compared to the simultaneous approach and full space SQP. To discuss the reason for this result, we consider an iterate at the feasible point $(x_k, y_k) = (1, 1/b)$, and generate a search direction $d_k = (d_x, d_y)$ by solving the quadratic subproblem 10 for Eqs. 44 and 45 with a Hessian matrix approximation assumed set to identity at iteration k.

Because

$$g(x_k, y_k) = \begin{bmatrix} 1 \\ \frac{1}{b} \end{bmatrix} \qquad A^T(x_k, y_k) = \begin{bmatrix} -b \\ -b^2 \end{bmatrix}$$

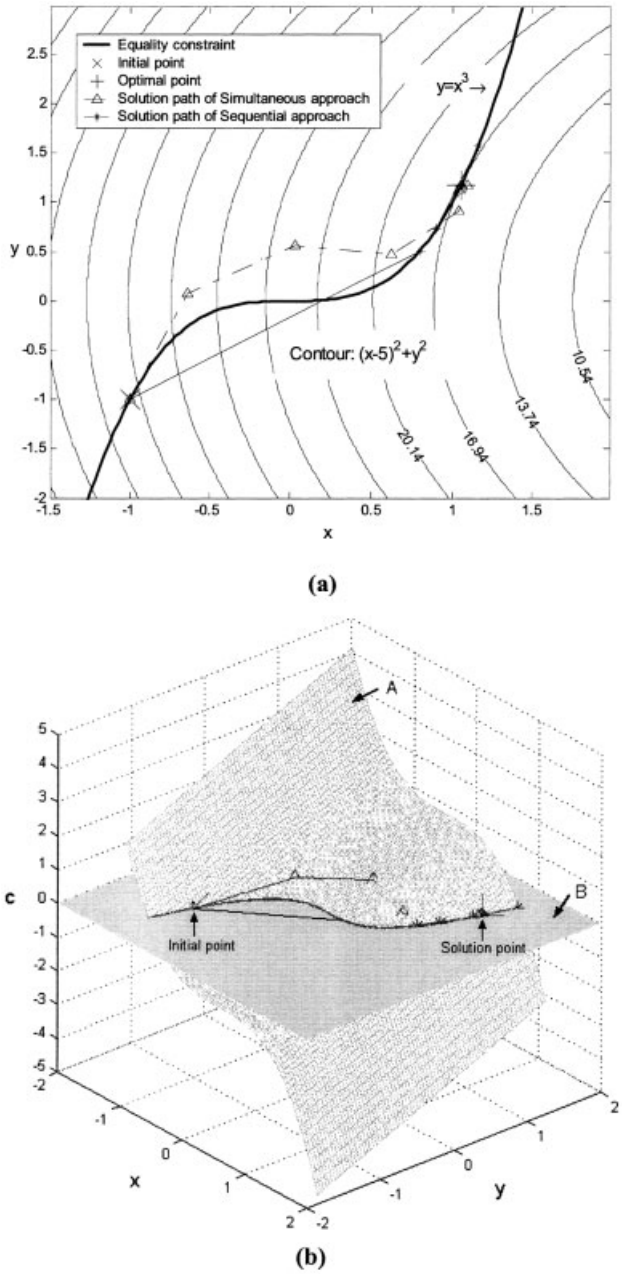

Figure 2. Solution paths of the optimization problem expressed in Eqs. 42 and 43: (a) in 2D and (b) in 3D.

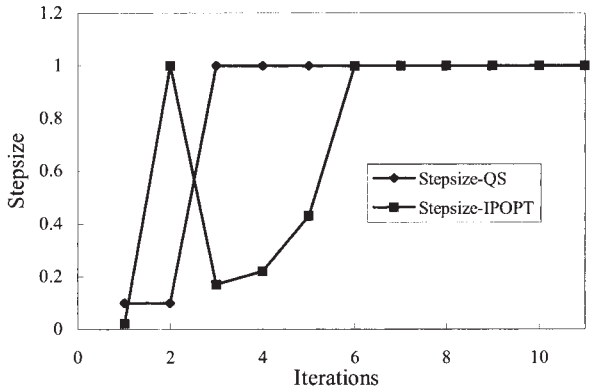

Figure 3. Comparison of convergence paths for solving the problem expressed in Eqs. 42 and 43.

the quadratic subproblem takes the form

$$\min d_x + \frac{1}{b} d_y + \frac{1}{2} d_x^2 + \frac{1}{2} d_y^2$$

$$\text{s.t.} \quad d_x = -b \cdot d_y$$

By solving this subproblem, we obtain

$$d_k = \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} -\dfrac{b^2 - 1}{b^2 + 1} \\ \dfrac{b^2 - 1}{b(b^2 + 1)} \end{bmatrix}$$

which yields a new trial point

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} 1 \\ \dfrac{1}{b} \end{bmatrix} + \begin{bmatrix} -\dfrac{b^2 - 1}{b^2 + 1} \\ \dfrac{b^2 - 1}{b(b^2 + 1)} \end{bmatrix} = \begin{bmatrix} \dfrac{2}{b^2 + 1} \\ \dfrac{2b}{b^2 + 1} \end{bmatrix}$$

Thus the objective function and constraints at the new trial point (x_{k+1}, y_{k+1}) are

$$f(x_{k+1}, y_{k+1}) = \frac{1}{2} (x_{k+1}^2 + y_{k+1}^2) = \frac{2}{b^2 + 1}$$

$$c(x_{k+1}, y_{k+1}) = \frac{1}{x_{k+1} \cdot y_{k+1}} - b = \frac{(b^2 - 1)^2}{4b}$$

If $b = 1000$, the value of the objective function f_{k+1} and the constraint c_{k+1} will be

Table 1. Iteration Number Required for the Simple Problem 44–45

	Quasi-Sequential	Simultaneous (IPOPT*)	Full Space SQP
$b = 100$	21	60	76
$b = 1000$	32	1324	869

*See Wächter.[33]

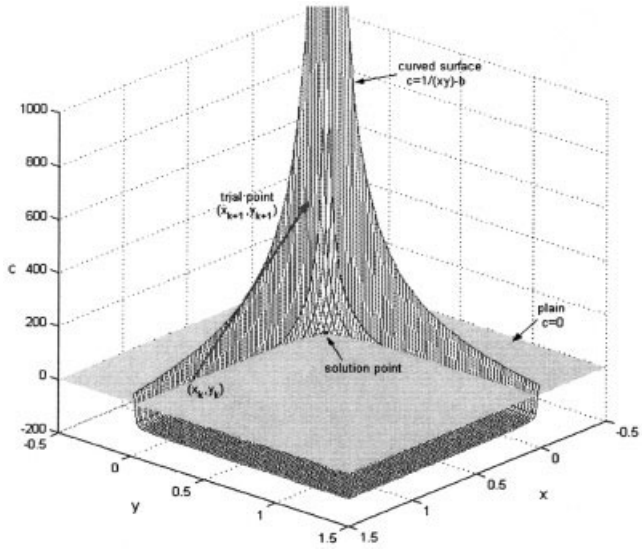

Figure 4. Sketch map of the simple optimization problem expressed in Eqs. 44 and 45.

$$f(x_{k+1}, y_{k+1}) = 2.0 \times 10^{-6} \qquad c(x_{k+1}, y_{k+1}) = 2.5 \times 10^{8}$$

whereas $f(x_k, y_k) = 0.5$ and $c(x_k, y_k) = 0$.

Note that, although the objective function decreases significantly, the constraint violation increases dramatically over this step. Even if $b = 100$, the constraint violation is also very high [that is, $c(x_{k+1}, y_{k+1}) = 2.5 \times 10^5$]. In Figure 4 we illustrate the case that when the trial point lies near the solution point, the value of constraint (c) will be high. In this case, the trial point will not be accepted in the line search step and small step sizes are likely to be calculated even in the neighborhood of the optimum (that is, the Maratos effect). In contrast, the equality constraints are not included in SQP in the quasi-sequential approach and the merit function for line search consists solely of the objective function. The information of the convergence paths is shown in Figure 5 with respect to $b = 100$. Here SIM means the full space SQP. Figure 5 shows that the quasi-sequential method takes larger step sizes and this leads to fewer SQP iterations.

Finally, we examine the convergence properties of the two approaches. The convergence analysis given in Biegler et al.[24]

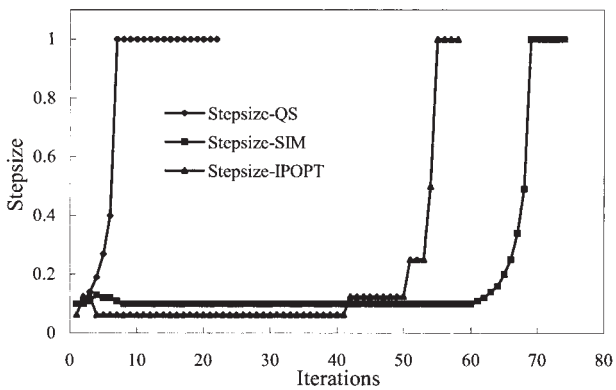

Figure 5. Comparison of convergence paths for solving the problem expressed in Eqs. 44 and 45.

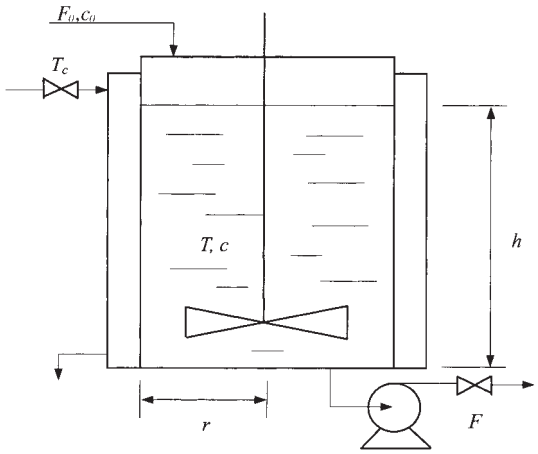

Figure 6. CSTR example.

shows that the reduced Hessian of the simultaneous approach is of a Q-superlinear rate of convergence for equality constrained problems. As for the quasi-sequential approach, it is equivalent to solving an unconstrained quadratic optimization problem. Furthermore, according to Nocedal and Wright,[34] the quasi-Newton SQP method normally is Q-superlinearly convergent. It can be concluded that these two approaches are well matched in terms of the convergence rate *as long as full steps are taken in the line search*. However, as seen above and also in the numerical experiments below, the quasi-sequential approach often leads to larger steps in the line search, although this comes as a trade-off with repeated solution of the DAEs in the quasi-sequential approach. Here the number of Newton iterations, *Num*, could be regarded as the comparison criterion between the quasi-sequential and the simultaneous methods. If *Num* is large, the quasi-sequential approach is not fit for solving such problems, and the simultaneous approach is more suitable. On the other hand, when the discretized DAE constraints are relatively easy to converge with Newton's method, as is often the case of optimal control problems, such problems can be solved more efficiently with the quasi-sequential approach.

Numerical Experiments

A FORTRAN package has been coded to carry out the implementation of this approach. With the discretization of the DAE system, the Newton algorithm is used to solve the nonlinear algebraic system. The SQP subroutine DNCONG in the IMSL Library[31] is used as the NLP solver for the quasi-sequential approach. In this section we present two process examples to show the effectiveness of the quasi-sequential approach. The first one is a continuous stirred-tank reactor (CSTR). The other one concerns the optimal operation for a heat-integrated distillation column system.

Table 2. Parameters of the CSTR

F_0	100 L/min	E/R	8750 K
T_0	350 K	U	915.6 W m^{-2} K^{-1}
c_0	1.0 mol/L	ρ	1 kg/L
R	0.219 m	C_p	0.239 J g^{-1} K^{-1}
K_0	7.2×10^{10} min^{-1}	ΔH	-5.0×10^4 J/mol

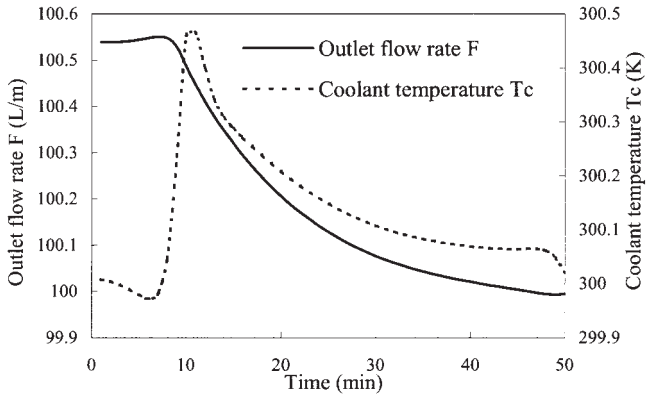

Figure 7. Optimal outlet flow and coolant temperature profiles.

Optimal control of a CSTR

A CSTR is considered as shown in Figure 6 An exothermic, irreversible, first-order reaction A → B occurs in the liquid phase and the temperature is regulated with external cooling. This example is taken from Henson and Seborg[35] or Pannocchia and Rawlings[36] with the assumption that the liquid level is not constant. Mass and energy balances lead to the following highly nonlinear state model

$$\frac{dh}{dt} = \frac{F_0 - F}{\pi r^2} \tag{46}$$

$$\frac{dc}{dt} = \frac{F_0(c_0 - c)}{\pi r^2 h} - k_0 c \exp\left(-\frac{E}{RT}\right) \tag{47}$$

$$\frac{dT}{dt} = \frac{F_0(T_0 - T)}{\pi r^2 h} + \frac{-\Delta H}{\rho C_p} k_0 c \exp\left(-\frac{E}{RT}\right) + \frac{2U}{r\rho C_p}(T_c - T) \tag{48}$$

The controlled variables are the level of the tank h and the product concentration c. The third state variable is the reactor temperature T. The manipulated variables are the outlet flow rate F and the coolant liquid temperature T_c. Moreover, it is assumed that the inlet flow F_0 or the inlet concentration c_0 acts as a disturbance. The model parameters in nominal conditions

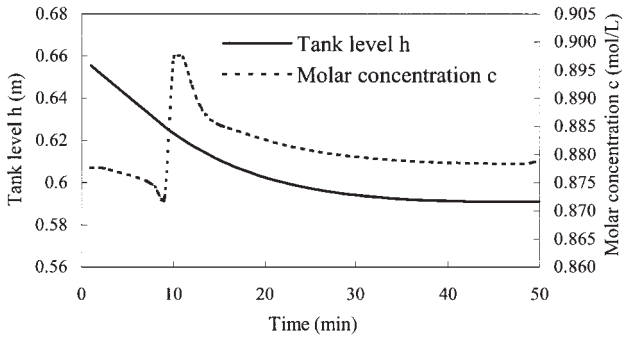

Figure 8. Profiles of controlled variables.

Tank level h and molar concentration c.

Table 3. Performance of Different Approaches to the CSTR Problem

Disturbance c_0 (mol/L)		Quasi-Sequential [Iter/CPU (s)]	Simultaneous (IPOPT) [Iter/CPU (s)]	Full Space SQP (IMSL) [Iter/CPU (s)]
No. 1	1.00–1.01	11/2.56	105/5.93	63/1481.50
No. 2	1.00–1.02	14/3.36	106/6.41	70/1628.74
No. 3	1.00–1.03	13/3.31	109/6.91	76/1732.02
No. 4	1.00–1.04	14/3.55	111/7.03	84/1962.58

are given in Table 2. The desired steady-state operating point is defined as follows: $h^s = 0.659$ m, $c^s = 0.877$ mol/L, $T^s = 324.5$ K, $F^s = 100$ L/min, and $T_c^s = 300$ K.

It is assumed that at $t = 10$ min a disturbance enters the plant, at a level of 0.05 mol/L on the inlet molar concentration c_0. Here the objective is to minimize the offset caused by the disturbance by controlling outlet flow rate F and the coolant liquid temperature T_c over a time horizon of $t_f = 50$ min. The problem is formulated as follows

$$\min \int_0^{t_f} [(h - h^s)^2 + 100(c - c^s)^2 + 0.1(F - F^s)^2 + 0.1(T_c - T_c^s)^2] \quad (49)$$

$$\text{s.t.} \quad \text{DAE model Eqs. 46–48} \quad (50)$$

$$0.5 \le h \le 2.50 \text{ m} \qquad 0.80 \le c \le 1.0 \text{ mol/L} \quad (51)$$

$$85 \le F \le 115 \text{ L/min} \qquad 299 \le T_c \le 301 \text{ K} \quad (52)$$

Here we divide the time horizon t_f into 50 elements and discretize the problem with three-point collocation in each element. The two control variables are represented as piecewise constants. Thus there are $2 \times 50 = 100$ control variables and $3 \times 3 \times 50 = 450$ state variables after the discretization. We initialize the controls with a constant outlet flow rate F value of 100 L/min and a constant coolant temperature T_c value of 300 K in each element. The algorithm converged in 16 iterations and 5.56 s of CPU time with a SUN Ultra 10 Station. The objective function value at the optimal point is 0.906112. The results are presented in Figures 7 and 8.

Now we compare the performance of both quasi-sequential and simultaneous approaches. Because there is a large parameter $K_0 = 7.2 \times 10^{10}$ in the model Eqs. 47 and 48, there exists a possibility that an infeasible path will cause a large violation of the equality constraints. To see the performance of the individual options introduced previously under this CSTR, the disturbance (that is, the inlet concentration c_0) was varied from $c_0 = 1.0$ to 1.01–1.04 mol/L at time $t = 10$ min. The computational results are listed in Table 3. The step sizes of the iterations by different approaches are shown in Figure 9, where the average number of Newton iterations required by the quasi-sequential approach in the simulation at each element is also given. It can be seen that the quasi-sequential approach takes full steps more often and therefore fewer iterations are required. Thus, although more than three Newton iterations are required for the simulation step, the total CPU time is much lower.

In implementing the quasi-sequential approach and the simultaneous approach (using the IPOPT package), the bounds (lower bounds = 0 and upper bounds = 10^6) are imposed only on control constraints. The results presented in Table 3 show that the quasi-sequential approach requires less computational cost. Also, because the full space SQP takes no advantage of sparsity, the CPU times are considerably higher. Nevertheless, the iteration numbers are still worth comparing.

To compare the performance of problem solutions with inequality constraints, we assume that the inlet flow rate F_0 acts as a disturbance and add all variable bounds as stated in Eqs. 51 and 52. As the disturbance is increased, the number of active constraints will change. Table 4 gives the results for number of active constraints in the first QP for different disturbances of F_0, in solving the problem with both approaches.

From Table 4 we see that as the number of active constraints is increased, the solution times by the quasi-sequential approach gradually increase, whereas the solution times by

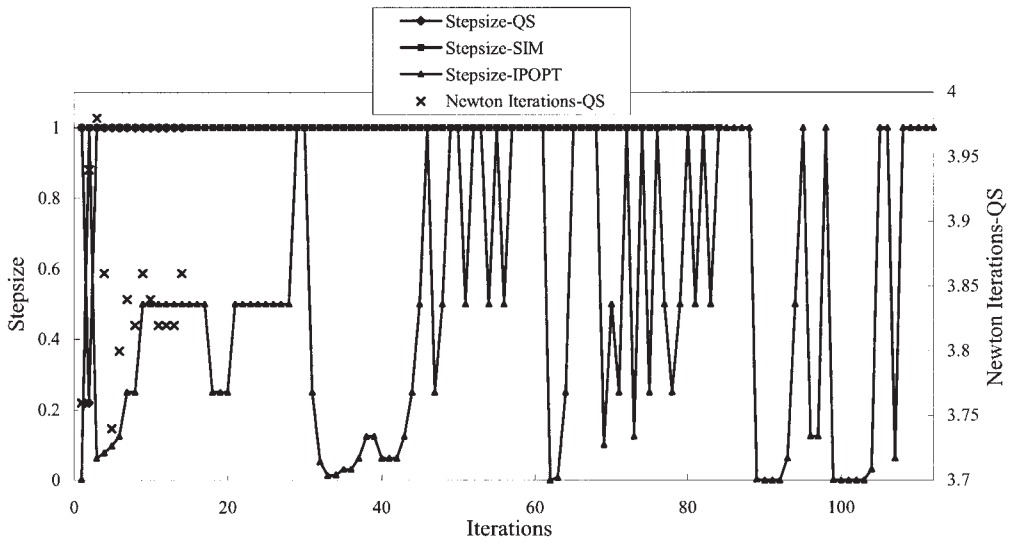

Figure 9. Comparison of convergence paths for solving the CSTR problem with disturbance c_0 = 1.00–1.04 mol/L.

Table 4. Comparison of Quasi-Sequential vs. IPOPT under Inequality Constraints

Disturbance F_0 (L/min)	Number of Active Constraints	Quasi-Sequential [Iter/CPU (s)]	IPOPT [Iter/CPU (s)]
100–106.5	0	13/3.03	107/10.82
100–107.5	13	14/3.22	106/11.29
100–108.5	25	16/3.75	106/10.68
100–109.5	35	15/3.65	105/11.00
100–110.5	44	24/5.88	102/10.45
100–111.5	51	18/4.87	99/10.15
100–112.5	59	Failure	101/10.13
100–113.5	65	26/8.57	101/10.11
100–114.5	65	24/8.34	101/9.93
100–115.5	70	32/11.03	101/9.97
100–116.5	73	35/12.01	103/10.17

IPOPT are almost unchanged. For the problem with less than 70 active constraints, from the perspective of CPU time, the quasi-sequential approach is faster than IPOPT. Yet, as the active constraints are increased, the IPOPT becomes faster than the quasi-sequential approach. This conclusion accords with the results presented in Ternet and Biegler.[37] In addition, when the disturbance of F_0 is 100–112.5, the quasi-sequential approach times out at intermediate step of DAEs solver, whereas IPOPT solves it. The line search step sizes with different approaches and the average Newton iteration numbers for simulation are shown in Figure 10.

We should point out that the comparison is also affected by a number of implementation details. In the quasi-sequential approach, we code the simulation algorithm ourselves. IPOPT uses sparse Harwell routines to solve linear algebraic systems. The convergence tolerance used to terminate the algorithm is thus different by different ways. In particular, quasi-Newton methods used with both approaches have a substantial impact on the efficiency of the algorithm. It should be noted that IPOPT can also exploit the use of exact second derivatives, thus leading to much better performance. However, this effect was not considered here. In conclusion, the results presented in this section indicate that the quasi-sequential approach is well suited for problems containing a moderate number of active constraints. For problems with many active constraints, the interior-point method with the simultaneous approach seems to be more effective.

Heat-integrated distillation column systems

In this example, we consider a pilot heat-integrated distillation system consisting of a high-pressure and a low-pressure column, both with a diameter of 100 mm, to separate a mixture of water and methanol. The columns have a central down-comer with 28 and 20 bubble-cap trays, respectively. The overhead vapor from the high-pressure column (HP) is introduced as the heating medium to the reboiler of the low-pressure column (LP). The reboiler duty of HP, reflux flow rate of both HP and LP, and feed flow rate of HP are the manipulated variables to be optimized for minimizing the reboiler duty consumption.

A rigorous tray-by-tray model is used to model the process. The equations of each tray consist of three parts: (1) mass and energy balances; (2) phase equilibrium equations; and (3) tray hydraulics relations, which lead to an index one DAE system. The heat capacity of the metal of the column wall and the tray elements are considered in the enthalpy balance. The vapor–liquid equilibrium is described by the Wilson and Antoine equations. The equilibrium state is corrected by Murphree tray efficiencies of stripping and rectifying sections. The tray hydraulics is modeled with the Francis–Weir formula to correlate the tray holdup with the liquid flow. The tray pressure drop is calculated by the gas and liquid fluid hydraulics based on the tray geometric size. As a result, a complex large-scale DAE system is formulated with 468 state variables and the same number of corresponding equations. A detailed model explanation can be found in Löwe[38] and Wendt et al.[11]

For such a continuous process we consider dynamic operation attributed to considerable disturbances from feed flow rate and composition. The aim of the optimization is to develop operation policies of the manipulated variables to minimize the reboiler duty, that is, the total energy consumption. Certainly, the compositions of distillate and bottom should satisfy the product specification. A large-scale dynamic optimization problem is formulated

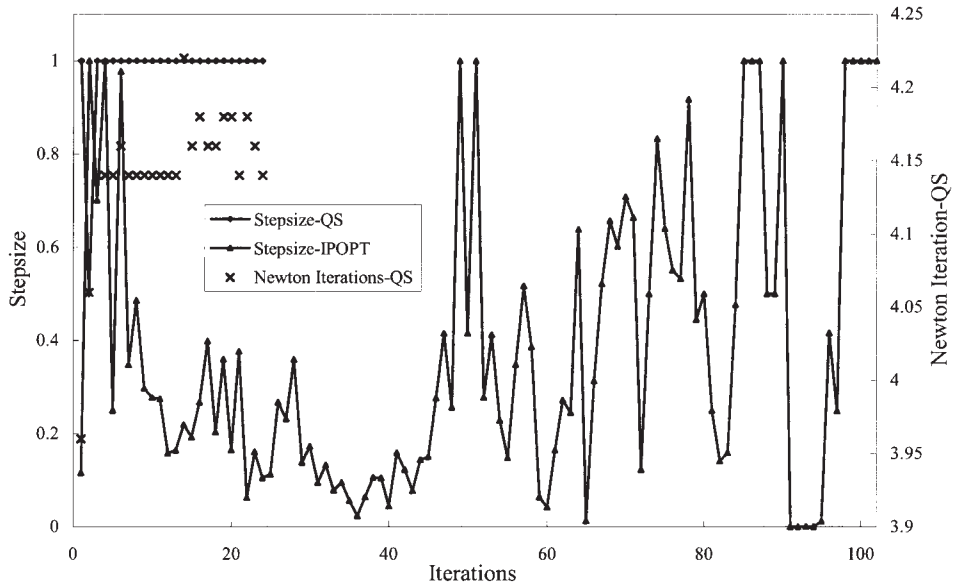

Figure 10. Comparison of convergence paths for solving the CSTR problem with disturbance F_0 = 100–110.5 L/min.

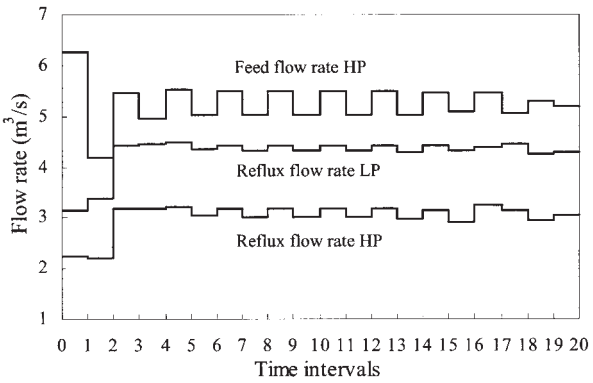

Figure 11. Dynamic feed composition disturbance and optimal control profiles.

$$\min \int_0^{t_f} Q^{HP} dt \tag{53}$$

$$\text{s.t.} \quad \text{DAE model equations.} \tag{54}$$

$$x_D^{HP} \geq x_D^* \qquad x_D^{LP} \geq x_D^* \qquad x_B^{HP} \geq x_B^* \qquad x_B^{HP} \geq x_B^* \tag{55}$$

$$L_{\min}^{HP} \leq L^{HP}(t) \leq L_{\max}^{HP} \qquad L_{\min}^{LP} \leq L^{LP}(t) \leq L_{\max}^{LP} \tag{56a}$$

$$F_{\min}^{HP} \leq F^{HP}(t) \leq F_{\max}^{HP} \qquad Q_{\min}^{HP} \leq Q^{HP}(t) \leq Q_{\max}^{HP} \tag{56b}$$

where Q^{HP} is the reboiler duty of HP and x_D^* and x_B^* are the distillation and bottom purity specifications, respectively. In this formulation, Eq. 53 represents the objective function, Eq. 54 is a large set of equality constraints, Eq. 55 consists of a set of path inequality constraints of state variables, and Eq. 56 describes the limitations of control variables.

For the pilot column system, we define the length of time horizon as $t_f = 5$ h, then divide the time horizon into 20 subintervals. The three-point collocation is used to approximate the state variables and the control variables are represented as piecewise constant in each element. After the discretization we have $(n - m) = 4 \times 20 = 80$ control variables and $m = 3 \times 468 \times 20 = 1044 \times 20 = 28{,}080$ state variables in the optimization problem. With the quasi-sequential framework, only the control variables and the inequality constraints are included in the optimization layer, whereas the state variables are solved through solution of the model equations in the simulation layer. For the dynamic optimization of the pilot plant, we consider large oscillating disturbances of the feed composition, as shown in Figure 11. With these disturbances, the product purity specification should be satisfied and, meanwhile, the reboiler duty should be minimized. Both the distillate and bottom specifications, x_D^* and x_B^*, are set to 0.99. The algorithm converged in 10 iterations with 21.50 min of CPU time. The optimization results are shown in Figures 11 and 12.

Conclusions and Future Work

We presented and analyzed a new approach, called a *quasi-sequential approach*, which is demonstrated to be efficient and well suited for large-scale problems with path constraints. In this approach, only the control variables have to be treated by the NLP solver and the state variables are eliminated in the simulation step by discretizing both the control and state variables. The quasi-sequential approach is well suited for solving large-scale optimal control problems in which the initial state usually is near the solution point, such as in disturbance rejection in process control. This is especially the case for highly nonlinear large-scale problems where the dynamic model equations may be violated by large values during the course of solution by the simultaneous approach. Moreover, unlike the simultaneous approach, the quasi-sequential approach does not require large-scale NLP algorithms. Instead, small-scale NLP solvers, which often include robust QR factorizations in the QP step, can be applied. We summarize the features of these different approaches in Table 5.

The quasi-sequential approach has its limitations in that the solution step might be infeasible at an intermediate trial point while solving an unstable system, just as in the sequential approach. Moreover, it is not effective for problems with many active constraints, such as the most active-set method for inequality constrained optimization. Finally, incorporating the

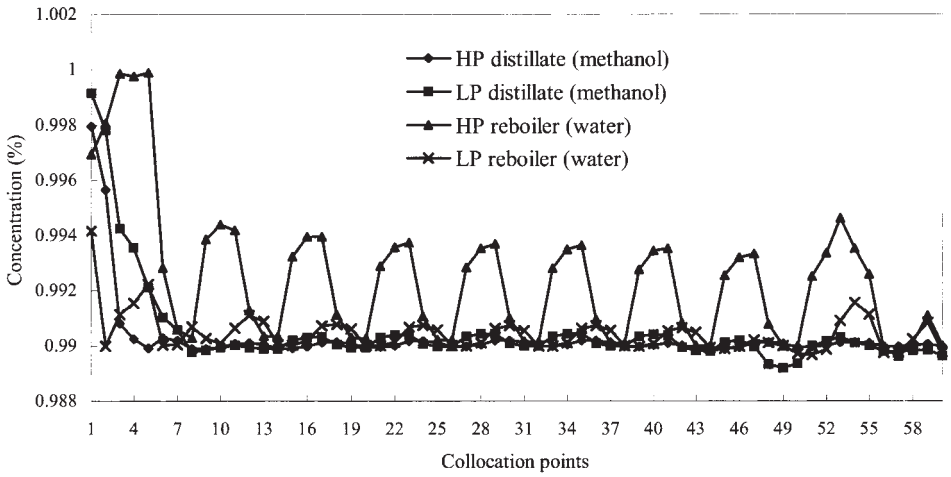

Figure 12. Optimal profiles of product compositions.

Table 5. Comparison of Properties of Dynamic Optimization Approaches

Simultaneous	Quasi-Sequential	Sequential
Full discretization of all the variables of DAE	Full discretization of all the variables of DAE	Only the control variables are discretized
Model equations are solved only once at the optimal point	Model equations are solved at each iteration	Model equations are solved at each iteration
Infeasible path method; the intermediate results are not useful	Feasible path method and the intermediate results are useful	Feasible path method and the intermediate results are useful
Requires large-scale NLP solvers	Allows small-scale NLP solvers	Allows small-scale NLP solvers
Easy-to-handle state variables path constraints	Easy-to-handle state variables path constraints	Difficult-to-handle states path constraints
Easy-to-handle unstable modes	Difficult-to-handle unstable modes	Difficult-to-handle unstable modes

quasi-sequential approach with the interior point method to handle inequality constraints remains an interesting topic for further research.

Acknowledgments

Financial support from the Sandwich scholarship program to Chinese researchers from the Technical University of Berlin is gratefully acknowledged.

Literature Cited

1. Biegler LT, Grossmann IE. Retrospective on optimization. *Comput Chem Eng.* 2004;28:1169-1192.
2. Logsdon JS, Biegler LT. Decomposition strategies for large-scale dynamic optimization problems. *Chem Eng Sci.* 1992;47:851-864.
3. Vassiliadis VS, Sargent RWH, Pantelides CC. Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints. *Ind Eng Chem Res.* 1994;33:2111-2122.
4. Barton PI, Allgor RJ, Feehery WF Galan S. Dynamic optimization in a discontinuous world. *Ind Eng Chem Res.* 1998;37:966-981.
5. Cervantes AM, Biegler LT. Large-scale DAE optimization using simultaneous nonlinear programming formulations. *AIChE J.* 1998;44:1038.
6. Cervantes AM, Biegler LT. A stable elemental decomposition for dynamic process optimization. *J Comput Appl Math*. 2000;120:41-57.
7. Biegler LT Nocedal J Schmid C, Ternet D. Numerical experience with a reduced Hessian method for large scale constrained optimization. *Comput Optim Appl.* 2000;15:45-67.
8. Biegler LT, Cervantes AM, Wächter A. Advances in simultaneous strategies for dynamic optimization. *Chem Eng Sci.* 2002;57:575-593.
9. Li P, Arellano-Garcia H, Wozny G. Optimization of a semibatch distillation process with model validation on the industrial site. *Ind Eng Chem Res.* 1998;37:1341-1350.
10. Li P, Wendt M, Arellano-Garcia H, Wozny G. Optimal operation of distillation processes under uncertain inflows accumulated in a feed tank. *AIChE J.* 2002;48:1198-1211.
11. Wendt M, Königseder R, Li P, Wozny G. Theoretical and experimental study on startup strategies for a heat-solved distillation column system. *Chem Eng Res Des Trans IChemE.* 2003;81a:153-161.
12. van Schijndel J, Pistikopoulos EN. Towards the solution of process design, process control and process operability—Current status and future trends. In: Malone M, Trainham J, Carnahan B, eds. *Foundations of Computer Aided Process Design 99* (AIChE Symp Ser 323); 2000;99-112.
13. Cervantes AM, Biegler LT. Optimization strategies for dynamic systems. In: Floudas C, Pardalos P, eds. *Encyclopedia of Optimization*. Dordrecht, The Netherlands: Kluwer Academic Publishers; 1999.
14. Storen S, Hertzberg T. The sequential linear quadratic programming algorithm for solving dynamic optimization problems—A review. *Comput Chem Eng.* 1995;19S:S495-S500.
15. Biegler LT. Advances in nonlinear programming concepts for process control. *J Proc Contr.* 1998;8:301-311.
16. Feehery WF. *Dynamic Optimization with Path Constraints*. PhD Dissertation. Cambridge, MA: Massachusetts Institute of Technology; 1998.
17. Feehery WF, Barton PI. Dynamic optimization with state variable path constraints. *Comput Chem Eng.* 1998;22:1241-1256.
18. Bock HG, Plitt KJ. A multiple shooting algorithm for direct solution of optimal control problem. Proc of 9th IFAC World Congress, Budapest, Hungary, July 2–6; 1984.
19. Leineweber DB, Schafer A, Bock HG, Schloder JP. An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. *Comput Chem Eng.* 2003;27:167.
20. Cervantes AM, Wächter A, Tütüncü R, Biegler LT. A reduced space interior point strategy for optimization of differential algebraic systems. *Comput Chem Eng.* 2000;24:39-51.
21. Biegler LT. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Comput Chem Eng.* 1984;8:243-248.
22. Floudas CA. *Deterministic Global Optimization: Theory, Algorithms and Applications.* Dordrecht, The Netherlands: Kluwer Academic; 2000.
23. Tawarmalani M, Sahinidis NV. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Dordrecht, The Netherlands: Kluwer Academic; 2002.
24. Biegler LT, Nocedal J, Schmid C. A reduced Hessian method for large scale constrained optimization. *SIAM J Optim.* 1995;5:314-347.
25. Bequette BW. Nonlinear control of chemical processes: A review. *Ind Eng Chem Res.* 1991;30:1391-1413.
26. Sistu PB, Gopinath RS, Bequette BW. Computational issues in nonlinear predictive control. *Comput Chem Eng.* 1993;17:361-366.
27. Pytlak R. *Numerical Methods for Optimal Control Problems With State Constraints* (Lecture Notes in Mathematics No. 1707). Berlin, Germany: Springer-Verlag; 1999.
28. Schittkowski K. NLPQL: A FORTRAN subroutine solving constrained nonlinear programming problems. *Ann Oper Res.* 1985;5:485-500.
29. Byrd R. An example of irregular convergence in some constrained optimization methods that use the projected Hessian. *Math Prog.* 1985;32:232-237.
30. Yuan Y. An only 2-step Q-superlinear convergence example for some algorithms that use reduced Hessian approximations. *Math Prog.* 1985;32:224-231.
31. International Mathematical Subroutine Library (IMSL) Visual Numerics web page. http://www.vni.com/products/imsl/alphabetized_functions.html. Bloomington, IN: IMSL, Indiana University.
32. Wächter A, Biegler LT. On the implementation of an interior point filter line search algorithm for large-scale nonlinear programming. *Math Prog.* 2005; published online 28 April.
33. Wächter A. *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. PhD Dissertation. Pittsburgh, PA: Carnegie Mellon University; 2002.
34. Nocedal J, Wright SJ. *Numerical Optimization.* New York, NY: Springer-Verlag; 1999.
35. Henson MA, Seborg DE. *Nonlinear Process Control*. Upper Saddle River, NJ: Prentice Hall; 1997.
36. Pannocchia G, Rawlings J. Disturbance models for offset-free model predictive control. *AIChE J.* 2003;49:426-437.
37. Ternet D, Biegler LT. Interior-point methods for reduced Hessian successive quadratic programming. *Comput Chem Eng.* 1999;23:859-873.
38. Löwe K. *Theoretical und experimental untersuchungen über das anfahren und die prozeβführung energetisch und stofflich gekoppelter destillationskolonnen*. PhD Dissertation. Berlin, Germany: Technishe Universität; 2001.
39. Papamichail I, Adjiman CS. Globe optimization of dynamic systems. *Comput Chem Eng.* 2004;28:403.

Manuscript received Jun. 9, 2004, and revision received Apr. 20, 2005.