GLOBALLY OPTIMAL NONLINEAR MODEL PREDICTIVE CONTROL

C. E. Long, P. K. Polisetty, and E. P. Gatzke [1]

Department of Chemical Engineering
University of South Carolina
Columbia, SC, USA 29208

Abstract
This paper presents a globally optimal nonlinear Model Predictive Control (NMPC) algorithm. Utilizing local techniques on nonlinear nonconvex problems leaves one susceptible to suboptimal solutions. In complex problems, local solver reliability is difficult to predict and often highly dependent upon the choice of initial guess. For the purpose of NMPC, local solvers can cause unexpected closed-loop results or failure of the algorithm. Stochastic attempts at global optimization of NMPC methods cannot provide rigorous bounds on the optimality of the resulting solution. Implementation of a global solution technique (Falk and Soland [1969], Horst and Tuy [1990]), which guarantees global optimality, restores the integrity of NMPC technology. Due to the combinatorial nature of nonconvex optimization, real-time considerations must be considered. The proposed algorithm's capabilities are demonstrated by the application of the controller on the benchmark control problem of the isothermal operation of a continuous stirred tank reactor (CSTR) with Van de Vusse reactions (Kremling and Allgöwer [1993]).

Keywords:
Constrained Nonlinear Control, Nonlinear Model Predictive Control, Global Optimization, Branch-and-Reduce, Convexification

INTRODUCTION

Given that a dynamic process can be approximated by a linear model near its nominal operating point, linear Model Predictive Control (MPC) methods typically provide reasonable control of a system (Morari and Lee [1991, 1997]). For these methods, a convex optimization problem is solved online at each time step for the optimal control sequence that will keep the system within a desired region of operation. However, in many processing examples, a process may not exhibit linear dynamics resulting in unstable or poor closed-loop performance using linear methods. Alternative formulations that consider nonlinear dynamics must be pursued. For constrained nonlinear systems, a nonlinear model can be used in nonlinear Model Predictive Control (NMPC) formulations (Henson [1998]). The nonlinearity in such formulations, does however, give rise to a more complex optimization problem to be solved online. In general, convexity of the resulting online optimization problem is lost. The nonconvex nonlinear program (NLP) formulation leaves one susceptible to determination of suboptimal solutions.

With a nonlinear discrete-time state-space representation of a process, the NMPC problem can be formulated at each sampling time. The controller output will depend upon online solutions of nonconvex optimization problems which will include nonconvex constraints representing a projection of the nonlinear model states and measurements into the

[1] Author to whom correspondence should be addressed: gatzke@sc.edu

future. This work presents a new NMPC application that attempts to find the guaranteed global optimum for problems posed by general NMPC algorithms.

Deterministic methods for the global solution of nonconvex problems typically rely on convex relaxations of nonconvex functions. Numerous methods (Adjiman, Dalliwig, Floudas, and Neumaier [1998], McCormick [1976], Tawarmalani and Sahinidis [2000], Gatzke, Tolsma, and Barton [2002]) have been proposed to carry out such relaxations. For this work, the nonconvex NLP is reformulated to a convex NLP via variable transformations. A linearization strategy of (Tawarmalani and Sahinidis [2000], Gatzke et al. [2002]) is then implemented to generate a Linear Program (LP). Calculating the lower-bounds for the problem then relies on solution of these LP's, for which various methods are available (Dantzig [1963], Karmarkar [1984], ILOG [2002]). Note that nearly any algebraic function can be reformulated for convexification.

This work expands on previous contributions which considered a relative small class of nonlinear dynamic systems (Sriniwas and Arkun [1997]). Once the convex relaxation is obtained, branch-and-bound (Falk and Soland [1969]) or branch-and-reduce (Ryoo and Sahinidis [1995]) techniques can be applied to solve the modified program providing a guaranteed global optimum (within tolerances). For faster convergence to this solution, the variable space can be reduced using standard techniques such as interval analysis (Moore [1979]). Although global optimization methods are combinatorial in nature, the NMPC formulation does have a relatively restricted variable branching space. In applying this solution technique to the NMPC, other issues arise regarding the algorithms use for real-time applications.

CONTROLLER FORMULATION

Assume that a given process with n_u inputs, n_x states, and n_y outputs can be represented by a nonlinear discrete-time state space model:

$$\begin{aligned} x(k+1) &= f(x(k), u(k)) \\ y(k) &= g(x(k), u(k)) \end{aligned} \tag{1}$$

where $x(k) \in \mathbb{R}^{n_x}$ is the state vector at sample time k, $u \in \mathbb{R}^{n_u}$ is the vector of inputs, and $y \in \mathbb{R}^{n_y}$ is a vector of the predicted outputs. Note that $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$ and $g : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$. Without loss of generality, it is assumed here that g is a linear map.

The NMPC is formulated to choose a sequence of input moves over the move horizon (m) that minimizes some cost function. This cost function typically quantifies the difference between the model predicted evolution of the system and the desired setpoints over some prediction horizon (p). A 1-norm objective function may take the form:

$$\Phi = \sum_{i=1}^{p} \Gamma_y(i) e(i) + \sum_{j=0}^{m-1} \Gamma_u(j) \Delta u(j) \tag{2}$$

where $e(i)$ is the absolute value of error predicted for the $(i)^{th}$ time step into the future. The error (e) is defined and constrained as:

$$|r_y(i) - y(i)| \leq e(i) \quad \forall i = 1...p \tag{3}$$

Δu is a vector defining changes in input movements. $\Gamma_y(i)$ and $\Gamma_u(j)$ are weighting factors used to define the relative importance of each objective function term.

The optimization problem to be solved at each time step includes constraints. The predicted state and output values are constrained by the model. The model predicted output is updated to account for any plant/model mismatch. This disturbance update is defined as:

$$d(i) = y_m(0) - y_p(0) \tag{4}$$

where $y_m(0)$ and $y_p(0)$ are the measurement at the current time and the predicted value of the output at the current time, respectively. This predicted output is based on state estimates from an open-loop observer run in parallel to the process. Constraints on the input movements are implemented as

$$|u(i-1) - u(i-2)| \leq \Delta u(i-1) \qquad \forall i = 1...m \tag{5}$$

Hard constraints on the actual inputs of the process are implemented as:

$$u^l \leq u \leq u^u \tag{6}$$

The program to be solved online by the NMPC algorithm has been completely defined and can be re-written in a more compact form:

$$\begin{aligned} \min_{\mathbf{z}} \quad & C^T z \\ s.t. \quad & A_1 z \leq b \\ & h_i(z) = 0 \;\; \forall i = 1...M \\ & z^l \leq z \leq z^u \end{aligned} \tag{7}$$

where the vector $z \in \mathbb{R}^N$ is a vector of N unknowns including the desired input trajectory, the resulting state, output, and error projections, as well as the resulting Δu terms. All linear inequality constraints are represented by $A_1 z \leq b$. The M nonlinear constraints that arise from the model and the objective function are written as $h_i(z) = 0$, where $h_i : z \rightarrow \mathbb{R}^M$, $M = n_x * p + 1$. Hard constraints on the inputs are incorporated into the bounds on the unknown vector (z^l and z^u).

GLOBAL SOLUTION

Deterministic methods for global optimization depend on the generation of convex relaxations of the original nonconvex nonlinear problems. Numerous methods have been proposed for constructing such relaxations. For this work, a reformulation method (McCormick [1976]) is used which converts the original factorable nonconvex nonlinear problem into an equivalent form by the introduction of new variables and new constraints. The reformulated problem contains only linear and simple nonlinear terms for which convex relaxations can be constructed using the convex envelopes already known for such simple algebraic functions. The reformulated Nonconvex NLP is of the form:

$$
\begin{aligned}
\min_{\mathbf{w},\mathbf{z}} \quad & C^T z \\
s.t. \quad & A_1 z \leq b \\
& A_2 \begin{bmatrix} w \\ z \end{bmatrix} \leq 0 \\
& w = h(w, x) \\
& z^l \leq z \leq z^u \\
& w^l \leq w \leq w^u
\end{aligned} \tag{8}
$$

where $A_2[w\ z]^T \leq 0$ defines the new linear constraints obtained from reformulation, while $w = h\,(w, z)$ provides the relationship between the new and original variables. With Q new variables, $w \in \mathbb{R}^Q$ and $h : w \times z \rightarrow \mathbb{R}^Q$. Bounds on w are determined from the bounds on z. Note that h consists of simple nonlinear terms relating 2or 3 variables.

Convex relaxations of this problem are then constructed using DAEPACK (Gatzke et al. [2002], Tolsma and Barton [2000]), an automated code generation tool. The advantage of using DAEPACK tool for generating convex relaxations is it can be applied to legacy models coded in standard FORTRAN. The convex relaxations can be denoted as:

$$
\check{h}(w, z, w^l, w^u, z^l, z^u) \leq w \leq \hat{h}(w, z, w^l, w^u, z^l, z^u) \tag{9}
$$

where $\check{h}$ and $\hat{h}$ are the convex under and over estimates of the reformulated problem.

The linearization strategy (Tawarmalani and Sahinidis [2000], Gatzke et al. [2002]) is then used to generate an LP relaxation of the convex NLP created using DAEPACK. The resulting LP is of the form:

$$
\begin{aligned}
\min_{\mathbf{w},\mathbf{z}} \quad & C^T z \\
s.t. \quad & A_1 z \leq b \\
& A_2 \begin{bmatrix} w \\ z \end{bmatrix} \leq 0 \\
& A_3 \begin{bmatrix} w \\ z \end{bmatrix} \leq b_3 \\
& z^l \leq z \leq z^u \\
& w^l \leq w \leq w^u
\end{aligned} \tag{10}
$$

where $A_3[w\ z]^T \leq b_3$ expresses the new linear constraints resulting from the linearization process. This linearization technique is ideal because it yields an LP for which robust solvers exist (e.g., ILOG CPLEX 8.0 ILOG [2002] and the IBM OSL library I. B. M. [1997]).

Upon creation of the linear (convex) underestimates for the nonconvex nonlinear problem, the branch-and-reduce method (Ryoo and Sahinidis [1995]) is implemented. This is an extension of the traditional branch-and-bound method with bound tightening techniques for accelerating the algorithm's convergence. Within this branch-and-reduce algorithm, infeasible or suboptimal parts of the feasible region can be eliminated using range reduction techniques such as optimality based and feasibility based range reduction tests (Ryoo and Sahinidis [1995]) or interval analysis techniques (Moore [1979]). These techniques help to derive tighter variable bounds for a given partition in the search tree. Finally, the algorithm terminates when the lower bounds for all partitions either exceed or are sufficiently close (within specified tolerances) to the best upper bound. At this point, a global optimum has been found.

In general, global optimization methods are combinatorial in nature. However, for this specific application, despite the NMPC formulation involving hundreds of variables, the problem has a relatively restricted variable branching space. Only the $m\,n_u$ true decision variables are branched on during the branch and reduce algorithm. The remainder of the variable bounds are defined by the selection of the optimal input sequence bounds. This is promising, especially with regards to the feasibility of the solution approach for real-time application in the NMPC context.

A modification of standard branch-and-reduce methods was required in order to allow for rapid global solution. Typically, a node is selected from the active node list. This partition is examined for a possible new improved upper bound to the overall problem by local solution of the corresponding nonconvex optimization problem. The node is then partitioned and lower bounds are derived for the new partitions based on the convex lower bounding problems. It was found that it may be necessary for rapid convergence in this real-time application to search for an upper bound for each partition using a local search as soon as any new partition is created. This amounts to a minor reordering of events in the typical branch-and-bound search algorithm. If this upper bound local solution search is not performed as early as possible for the partition, a partition containing the global solution may be added to the active node list without a local solution and this node may not be selected from the list early on in the branch-and-bound search. This is especially problematic for problems that contain partitions with highly degenerate lower bounds, as is the case in NMPC types of problems. The lower bounding solutions may return an objective of zero for many of the early partitions. Depending on the implementation, the partition containing the global solution may not be selected from this active node list immediately in typical implementations, delaying the determination of the global solution. Performing a local searching on every node as soon as the partition is created helps determine the global solution early on, which also leads to more rapid convergence as more of the parameter space can be fathomed using the branch-and-reduce methods.

CASE STUDY

Consider the benchmark control problem of the isothermal operation of a two state continuous stirred tank reactor (CSTR) with Van de Vusse reactions. In this reactor, the Van de Vusse reactions are:

$$A \rightarrow B \rightarrow C$$
$$2A \rightarrow D$$

Material balances dictate that the system can then be described by:

$$\frac{dC_a}{dt} = (F/V)(C_{ao} - C_a) - k_1 C_a - k_3 C_a^2 \qquad (11)$$

$$\frac{dC_b}{dt} = k_1 C_a - k_2 C_b - (F/V) C_b \qquad (12)$$

where F is the feed flow rate of A into the reactor, V is the constant reactor volume, C_a and C_b are the reactant concentrations in the reactor, and k_i are the reaction rate constants for the three reactions. For this work, let $k_1 = 50\,h^{-1}$, $k_2 = 100\,h^{-1}$, and $k_3 = 10\,L\,gmol^{-1}\,h^{-1}$. Assume that the volume of the reactor is constant, that the feed is pure A, and that the nominal concentration of A in the feed (C_{ao}) is $10\,gmol\,L^{-1}$.

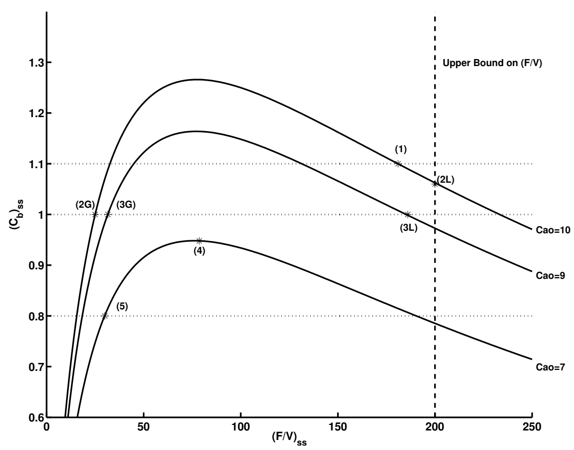

Figure 1. Steady state loci for various feed concentrations showing steady state operating points in the Local (L) and Global (G) solution methods.

For this single input single output (SISO) system, the input (u) is taken as (F/V) , the state vector (x) consists of the concentrations of A and B in the reactor, and the single measurement (y) is that of the concentration of B (i.e., $y = x_2$). An unmeasured disturbance (d) will be simulated through changes in C_{ao}. The discrete-time model necessary for the predictive aspects of this type of control is found by discretizing the nonlinear state-space model using a backward difference approximation with a sampling rate of 0.002 hours. It should be noted that this system happens to exhibit a steady-state input multiplicity. This is depicted in a plot of the steady-state loci presented in Figure 1. An upper bound on the input (F/V) is assumed to be at a value of 200 h^{-1}.

The predictive controller is tested for its abilities in both setpoint tracking and disturbance rejection. The controller is tuned with $m = 1$, $p = 30$, $\Gamma_y(p) = 100$, and $\Gamma_u = 0$. Note that $\Gamma_y(k) = 0\ \forall k \neq p$. By weighting only the p^{th} error term in the projection, a terminal error penalty is enforced. Assume that the process is initially operated at $u = 181\,h^{-1}$ and $y = 1.1\ gmol/L$. This operating point is indicated by (1) in Figure 1. At a time of 0.1 hours, the setpoint is stepped to $1\ gmol/L$. A series of unmeasured disturbances are then introduced. At a time of 0.5 hours, the setpoint is again stepped down, this time under disturbance, to a value of $0.8\ gmol/L$.

The closed-loop results for the controller using both the local solution techniques (MINOS version 5.51) and the proposed global technique are shown in Figure 2. At the

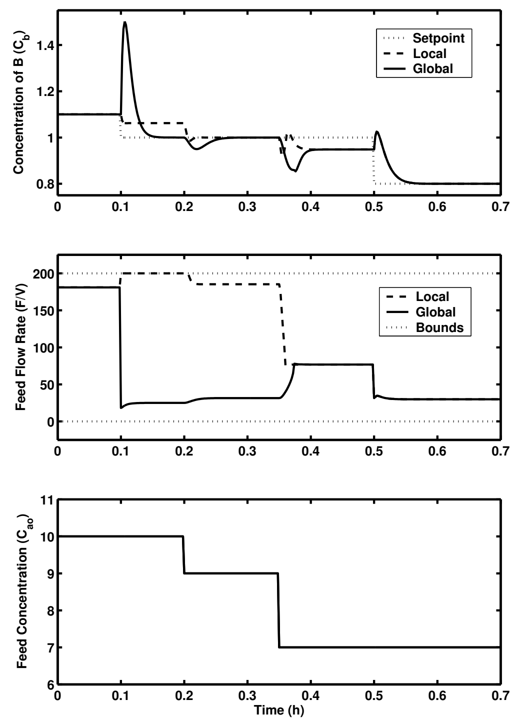

Figure 2. Closed-loop results (m=1) with setpoint changes at 0.1 and 0.5 hours and disturbance loads at 0.2 and 0.35 hours.

time of the first setpoint change, the local solver moves the system in an improving direction but ends up at a local minima (against the upper bound constraint). This is indicated by (2L) in Figure 1. The global solver is able to realize the full setpoint change by finding the global solution (2G). This example shows that the global NMPC can achieve superior performance. A sample objective function for the optimization problem encountered under this circumstance is provided in Figure 3.

At $t = 0.2$ hours, the first disturbance hits and both algorithms (using local and global solution techniques) are then able to track the setpoint under this disturbance. However, the presence of the input multiplicity allows them to do so at different operating conditions (denoted 3L and 3G in Figure 1).

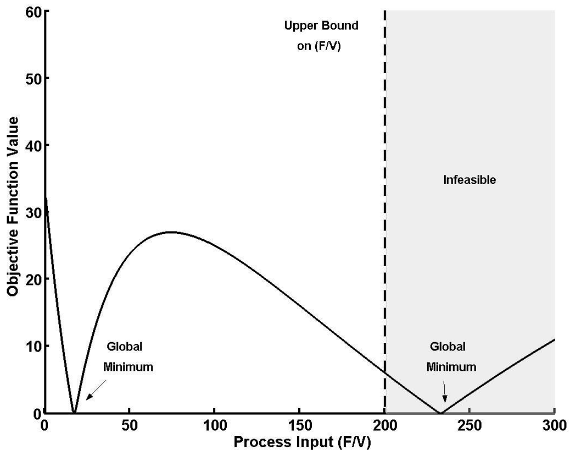

Figure 3. Sample objective function at the time of a setpoint change.

From this, it is obvious that the local solver is moving in the improving direction toward a minimum that is infeasible due to the upper bound on the input variable. At $t = 0.35$ hours, a second disturbance hits that moves the system to an operating regime in which the setpoint of $1.1\ gmol/L$ can no longer be achieved. Both the local and global solution techniques move the system to the optimal operating point denoted as (4) in Figure 1. Finally, at $t = 0.5$ hours, the second setpoint change is implemented. Both algorithms track this reference change without issue.

For a 0.7 hour simulation with a sampling rate of 0.002 hours, 350 optimization problems are solved online. The required time for finding the global solution for each problem is presented in Figure 4. In most cases using Redhat Linux 9.0 on a dual AMD 1900+ MP system, the global solver is able to guarantee global optimality sufficiently fast for real-time operation (i.e., the solver returns the global optimum within 7.2 seconds). However, at a time of 0.35 hours, the system is under a disturbance large enough that the desired setpoint can no longer be achieved. At this particular point, guaranteeing the global solution takes significantly more time than previous cases and the solver is no longer fast enough for real-time purposes. This can be attributed to a flat spot in the objective function (Figure 5). In order to account for this, the global solver is terminated at the real-time threshold and the best solution thus far is implemented. In this case, there is no degradation in the controller's performance by terminating the solve at the real-time threshold. However, the guarantee on global optimality is lost. It is likely that the global solution has indeed been found, however, some suboptimal regions of the solution space have not yet been fathomed. A measure of this optimality gap is provided in Figure 4. Convergence is dictated by bringing the lower bound within some tolerance of the upperbound. This plot shows the difference between the two upon termination.

Closed-loop results with a larger move horizon are presented in Figure 6. Again, the controller is choosing the

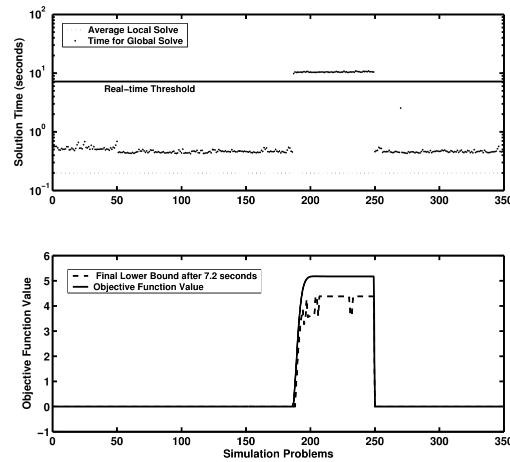

Figure 4. Solution times for the NMPC problems showing the time for the global solution, the global solution value, and the worst case lower bound upon termination for real-time application.

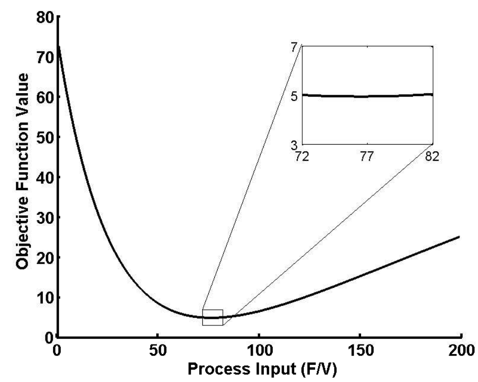

Figure 5. Sample objective function exhibiting a region of insensitivity to changes in the input.

optimal mn_u moves that minimize the objective function. The controller then implements only the first control move in that sequence. In an attempt to avoid the scenario in which the controller chooses not to move the system in this first calculated move, the objective function was modified from the previous case. An additional error term from the prediction horizon was included in the objective function to account for process dynamics. An input movement penalty was also implemented. The specific controller tunings were $m = 3$, $p = 30$, $\Gamma_y(15) = \Gamma_y(30) = 100$, and $\Gamma_u = 0.005$. Again note that $\Gamma_y(k) = 0 \quad \forall k \neq p,\ \frac{p}{2}$. Velocity constraints were also imposed on the system as:

$$\Delta u(i) \leqslant 70 \qquad \forall i = 1..m \tag{13}$$

The controller was subjected to the same setpoint tracking and disturbance rejection tests as in the $m = 1$ case. Interestingly enough, the NMPC using local solution techniques was able to track the reference changes and reject the disturbances without issue. In particular, when the initial setpoint change is applied, the controller based on local solution techniques is able to move the system to the setpoint instead of moving the input value to the upper bound as before. On the other hand, the NMPC algorithm using the global

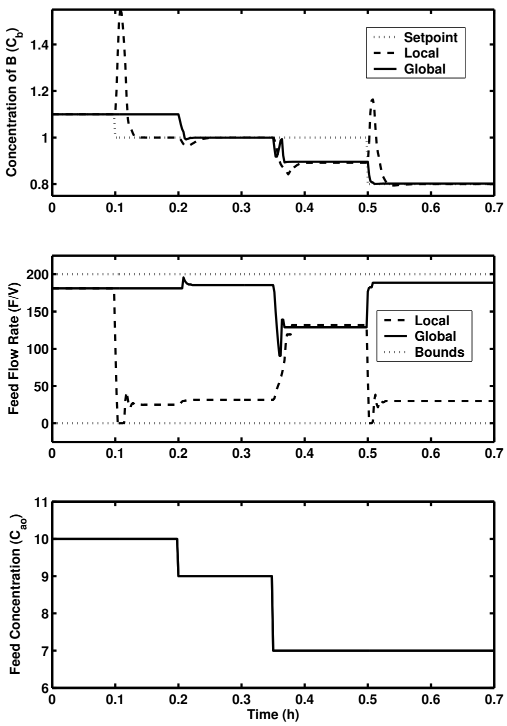

Figure 6. Closed-loop Results (m=3) with setpoint changes at 0.1 and 0.5 hours and disturbance loads at 0.2 and 0.35 hours

solution technique did not perform as well. At the time of this first setpoint transition, the controller chose not to move. The controller did indeed return the global solution of the optimization problem at this point, however, the solution dictated that it was not necessary for the controller to move the system immediately (i.e., in the first move of the chosen optimal input sequence). For the remainder of the simulation, the NMPC algorithm using the global solution technique was able to track the reference and reject the disturbances. It should be noted that it was able to do so at different input values than the local method, which is made possible by the presence of the input multiplicity.

CONCLUSIONS

A globally optimal NMPC algorithm has been proposed. A deterministic approach is used in finding the guaranteed global optimum to the nonconvex NLPs associated with the controller's operation. The global algorithm was shown to eliminate the poor performance in a simple CSTR example resulting from the suboptimal input trajectories supplied by a controller which uses local solution techniques, provided that the controller was properly tuned. In doing so, the globally optimal NMPC algorithm showed promise with regards to its real-time application. However, backup hybrid control methods should be considered to handle situations where the desired solution cannot be obtained in the allotted time.

Acknowledgments

The authors would like to acknowledge financial support from the American Chemical Society Petroleum Research Foundation grant # 38539-G9 and the National Science Foundation Early Career Development grant CTS-038663.

REFERENCES

C. S. Adjiman, S. Dalliwig, C. A. Floudas, and A. Neumaier. A Global Optimization Method, αBB, for General Twice-Differentiable Constrained NLPs - I Theoretical Advances. *Comput. Chem. Eng.*, 22(9):1137–1158, 1998.

G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.

J. E. Falk and R. M. Soland. An Algorithm for Separable Nonconvex Programming Problems. *Management Science*, 15(9):550–569, 1969.

E. P. Gatzke, J. E. Tolsma, and P. I. Barton. Construction of Convex Function Relaxations Using Automated Code Generation Techniques. *Optimization and Engineering*, 3(3):305–326, 2002.

M. A. Henson. Nonlinear Model Predictive Control: Current Status and Future Directions. *Computers and Chemical Engineering*, (23):187–202, 1998.

R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer, Berlin, 1990.

I. B. M. . IBM Optimization Solutions and Library Linear Programming Solutions. Technical report, I. B. M., 1997.

ILOG. *ILOG CPLEX 8.0: User's Manual*. Mountain View, CA, July 2002.

N. Karmarkar. A New Polynomial-Time Algorithm for Linear Programming. *Combinatorica*, 4:373–395, 1984.

A. Kremling and F. Allgöwer. Optimaler Betrieb eines kontinuierlichen Ruhrkesselreactors mit Parallel und Folgereaction. *preprint*, 1993.

G. P. McCormick. Computability of Global Solutions to Factorable Nonconvex Programs: Part I - Convex Underestimating Problems. *Mathematical Programming*, 10:147–175, 1976.

R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.

M. Morari and J. H. Lee. Model Predictive Control: The Good, the Bad, and the Ugly. In *Proceedings of the Fourth International Conference on Chemical Process Control*, pages 419–444, Padre Island, TX, 1991.

M. Morari and J. H. Lee. Model Predictive Control: Past, Present and Future. In *PSE ESCAPE-7 Symposium*, Trondheim, Norway, 1997.

H. S. Ryoo and N. V. Sahinidis. Global Optimization of Nonconvex NLPS and MINLPs with Application to Process Design. *Comput. Chem. Eng.*, 19(5):551–566, 1995.

G. R. Sriniwas and Y. Arkun. A Global Solution to the Nonlinear Model Predictive Control Algorthm Using Polynomial ARX Models. *Comput. Chem. Eng.*, 21(4):431–439, 1997.

M. Tawarmalani and N. V. Sahinidis. Global Optimization of Mixed Integer Nonlinear Programs: A Theoretical and Computational Study. Technical report, University of Illinois, 2000.

J. E. Tolsma and P. I. Barton. DAEPACK: An Open Modeling Environment for Legacy Code. *Ind. Eng. Chem. Res.*, 39(6):1826–1839, 2000.