Hard Constraints for Prioritized Objective Nonlinear MPC

Christopher E. Long and Edward P. Gatzke

University of South Carolina, Department of Chemical Engineering, Columbia, SC, USA 29208
gatzke@sc.edu

Summary. This paper presents a Nonlinear Model Predictive Control (NMPC) algorithm that uses hard variable constraints to allow for control objective prioritization. Traditional prioritized objective approaches can require the solution of a complex mixed-integer program. The formulation presented in this work relies on the feasibility and solution of a relatively small logical sequence of purely continuous nonlinear programs (NLP). The proposed solution method for accomodation of discrete control objectives is equivalent to solution of the overall mixed-integer nonlinear programming problem. The performance of the algorithm is demonstrated on a simulated multivariable network of air pressure tanks.

1 Introduction

Model Predictive Control (MPC) technology is most notable for its ability to control complex multivariable industrial systems. The model-based control scheme relies on the online solution of an optimization problem for the optimal control sequence that minimizes a cost function which evaluates the system over some prediction horizon. The typical cost function accounts for numerous control objectives spanning different levels of relative importance including those stemming from equipment limits and safety concerns, product quality specifications, as well as economic goals. Traditional formulations penalize violations of soft constraints to achieve the desired performance. However, they often rely on *ad hoc* tuning to determine the appropriate trade-off between the various control objectives. Moreover, the tuning becomes less intuitive for systems of increasing complexity.

Recent studies have focused on ensuring that control objective prioritization is handled effectively [3, 4, 7, 8, 13, 14, 18, 19]. Mixed integer methods utilize propositional logic [17] and binary variables to define whether discrete control objectives have been met and whether they are met in order of priority. Terms are included in the cost function to penalize failure to meet the prioritized objectives and the resulting mixed integer program is solved to determine the appropriate input move(s). Nevertheless, the mixed-integer programs are inherently combinatorial in nature and can prove to be computationally demanding, making real-time application difficult. This is of particular concern in nonlinear

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 217–228, 2007.
springerlink.com

formulations with logic constraints that require the solution of the nonconvex Mixed-Integer Nonlinear Program (MINLP).

Here, a nonlinear MPC formulation is presented that utilizes hard constraints to allow for control objective prioiritization. This avoids the need to solve the complex mixed-integer programming problem. By imposing hard variable constraints corresponding to process control objectives such as variable upper and lower bounds, the feasibility of the resulting nonlinear program (NLP) can be evaluated to determine whether a solution exists that will satisfy the control objective. Beginning with the highest priority objective, the hard constraints can be appended to the traditional MPC formulation and the feasibility can be tested in an orderly fashion to determine the optimal control sequence while addressing objective prioritization. This framework is equivalent to the mixed-integer formulation with geometrically weighted control objectives, in which a given control objective is infinitely more important that subsequent objectives. Instead of solving the combinatorial MINLP, the hard constraint approach provides an efficient and logical progression through the binary tree structure. This requires solution of a minimal number of NLP's.

The nonconvex nature of each NLP poses interesting problems in this NMPC formulation with regard to the determination of both global feasibility and global optimality. Use of local solution techniques lacks the ability to provide any indication of problem feasibility and leaves one susceptible to suboptimal solutions. A globally convergence stochastic approach is used to provide indication of global infeasibility and to pursue the global optimum. Deterministic approaches to guarantee global optimality are also considered, as they have been used previously in NMPC formulations[10].

2 Mixed Integer Control Formulation

In traditional Model Predictive Control approaches, control objectives are managed through penalizing violations of soft constraints. The typical objective function to be minimized at each time step is of the form:

$$\Phi = \sum_{i=1}^{p} e(i)^T \Gamma_e e(i) + \sum_{i=1}^{m} \Delta u(i)^T \Gamma_{\Delta u} \Delta u(i) \tag{1}$$

Here m and p are the move and prediction horizon. The error vector, e, represents the difference in the model predicted value and the desired reference for each of the controlled variables, while Δu is a vector describing the level of actuator movement. A vector of weights, or penalties (Γ_e), consists of constant scalar elements that represent the cost of violating each control objective. It is here that control objectives are assigned a relative importance or priority. Likewise, the elements of $\Gamma_{\Delta u}$ are utilized to suppress unnecessary control moves. The downfall of this approach is that it relies on *ad hoc* tuning to determine the appropriate trade-off between meeting various control objectives. For example, it is often difficult to infer how a controller will choose between violating a given constraint by a large amount for a short period of time, or violating a given constraint by a small amount for a more substantial period of time. This becomes particularly troublesome as the complexity of the system increases.

Mixed integer formulations have been used in the MPC framework in an attempt to make controller tuning more intuitive and to insure that higher priority control objectives are accommodated before the less important objectives [8, 9]. This is accomplished by adding additional terms to the traditional objective function and by introducing binary variables into the optimization problem. The premise of this approach is that errors associated with each control objective are explicitly defined and the traditional control objectives can be discretized using propositional logic and a "big M" constraint of the form:

$$e(i) \leq M(1 - O_j) \quad \forall j = 1..p \tag{2}$$

Here the O_j is a binary flag defining whether a given control objective can be met absolutely and M is a large value. Thus, if the error associated with a given control objective is zero, implying the control objective can be met at each point across the prediction horizon, the binary flag can take on a value of 1. Otherwise, O_j is forced to a value of zero in order to relax the constraint. Additional constraints can then be used to indicate whether a control objective is met in order of its priority. A binary flag, P_j is used to indicate if this is indeed the case. Constraints of the form $P_j \leq O_j$ are required to insure that the objectives are met before being flagged as met in order according to their relative priority. A set of constraints of the form $P_{j+1} \leq P_j$ are used to force higher priority objectives to be met first. The objective function in such a formulation is of the form:

$$\Phi = \Gamma_O O + \Gamma_P P + \sum_{i=1}^{p} e(i)^T \Gamma_e e(i) + \sum_{i=1}^{m} \Delta u(i)^T \Gamma_{\Delta u} \Delta u(i) \tag{3}$$

Here O is a vector of the binary flags defining if the discrete objectives have been met and P is a vector of binary flags defining whether or not the control objectives have been met in order of priority. Γ_O and Γ_P are vectors of weights that reward meeting the discretized objectives in order or priority. Typically the values of Γ_P are chosen to be much larger than the elements of Γ_O, which are orders of magnitude larger than the traditional MPC penalties. This approach has been demonstrated in a number of cases, including in the inferential control of unmeasured states in which a state space model is employed to explicitly define the unmeasured states to be constrained [8].

One drawback of the mixed integer MPC approaches is the combinatorial nature of the resulting mixed integer optimization problem. The resulting mixed-integer linear programming (MILP) problem can require the solution of up to 2^N LPs where N is the number of binary variables incorporated into the problem. Fortunately, the computational demand is often mitigated by the relatively few number of true decision variables associated with the MPC problem. The controller need only specify the m moves for each of the n_u process inputs, and the remaining variables (modeled process states, errors, etc) are then subsequently defined based on their relationship to the chosen input sequences, the model, and process data. From a practical standpoint, the approach can also be limited by problem dimensionality as penalties spanning numerous orders of magnitudes raise issues with solver tolerances.

To date, these mixed integer approaches for prioritized objective control have focused on linear formulations. With the availability of efficient MILP solvers [6], the methods prove to be viable for real-time control. This is particularly true for control of chemical processes, which typically have time constants on the order of minutes. However, many industrial processes are sufficiently nonlinear to motivate the consideration of nonlinear formulations. The use of a nonlinear model provides improved closed-loop performance but at the expense of increased computational demand as the nonlinear formulation inherently relies on the solution of a more difficult nonconvex nonlinear problem (NLP). As the prioritization of control objectives requires the solution of an MILP in the linear case instead of an LP, the nonlinear mixed integer formulation requires the solution of a difficult mixed integer nonlinear program (MINLP) instead of a single NLP.

3 Hard Constraints Formulation for Objective Prioritization

Assume that for a particular control problem n control objectives are to be handled appropriately based on their perceived relative priority. In the mixed integer formulation, this would require in the worst case the solution of $2^{2n+1}-1$ LP relaxation nodes in a traditional branch-and-bound search. This stems from the $2n$ binary variables. Efficient MILP solvers exist [6] and have been shown to be viable for real-time implementation in linear MPC formulations. However, for nonlinear dynamic formulations that consider solution of nonconvex MINLP's, the optimization problem can prove to be too computationally demanding. The motivation of the hard constraint formulation is a reduction in the computational demand to make prioritized objective NMPC possible for real-time control.

This algorithm uses hard variable constraints as a means to avoid the need to solve the complex MINLP for control objective prioritization in nonlinear MPC formulations. Consider the nonlinear control problem with n prioritized control objectives. Provided that each control objective is infinitely more important than subsequent objectives, these constraints can be handled using a logical progression through a reduced binary tree structure. Initially, a purely continuous constrained NLP is formulated. A traditional objective function (as in Equation 1) is used with explicitly defined errors for violations associated with each control objective constraint at each point in the prediction horizon. Soft constraint penalty weights are defined as in typical MPC methods. These dictate the controller performance in cases in which a control objective cannot be met for all points in p. Note that as with all soft constraint formulations, this NLP is inherently feasible.

Starting with the highest priority objective, the ability to meet each individual control objective is considered. First, hard constraints forcing the errors associated with the highest priority objective to zero are incorporated into the optimization problem. The constraints are of the form: $e_i(k) = 0 \; \forall k = 1..p$ where i corresponds to the objective. The feasibility of the modified problem is then examined, but the actual ***global solution is not required***. If feasible, it is the known that the control objective can be met and the constraints are left in the problem. However, if infeasible, the corresponding hard constraints are removed (relaxed). Note that at this point, violations of this unachievable control objective will be ultimately minimized based on their appearance in the traditional objective function as soft penalties. The next highest priority objective is then considered. Appropriate hard constraints are again added to the problem and the feasibility is again tested, with two possible results: The second problem considered will involve hard constraints corresponding to the two highest priority control objectives if the initial problem was feasible. However, if the initial problem was not feasible, the second problem will only consider the hard constraints associated with the second control objective. All subsequent control objectives are considered one at a time according to this procedure. This will define a single NLP that represents the final node from the binary tree that would have yielded the optimal solution. Again, the traditional weights associated with each error variable are still necessary. These values will define how the controller will handle cases in which the hard constraints associated with particular prioritized control objectives cannot be met absolutely over the whole prediction horizon. When it has been determined that a hard constraint cannot be met without sacrificing higher priority objectives the control algorithm will fall back to the traditional weights as it minimizes soft constraint violation. This NLP is then solved for the optimal input sequence which can then be implemented. Pseudo-code of this algorithm is presented in Algorithm 1.

Ultimately, this approach requires only that the feasibility of a maximum of n problems be assessed and then only the solution of a single NLP. Effectively, each individual feasibility check represents the binary flag from the mixed integer formulation that defines whether or not the discretized control objective can be met. Checking the numerous control objectives individually in the order of priority replaces the need for the binary variables and additional propositional logic constraints associated with meeting the objectives in order. Note that this framework is exactly equivalent to the mixed-integer implementation with geometrically weighted control objectives in which a given control objective is infinitely more important than subsequent objectives.

As presented, this NMPC formulation judiciously handles n prioritized control objectives through the consideration of a maximum of n NLPs. The feasibility of up to n problems must be determined and the appropriate NLP is then solved to global optimality. This maximum number of problems is encountered in the instance where the feasibility of the problem associated with each control objective is considered individually beginning with the highest priority objective. However, a number of heuristics can be utilized to further reduce the computational demands. For example, under relatively normal operating conditions, it is to be expected that a large number of the control objectives associated with

```
Formulate traditional soft constraint MPC problem with explicit errors for each control objective.
(Eq.1)
Rank control objectives in order from highest to lowest priority.
FOR i from 1 to # of Prioritized Objectives
    Impose hard constraints with highest priority control objective i not yet considered:
        (e_i(k) = 0 ∀k = 1..p)
    Check problem feasibility (Solve deterministically until infeasible or feasible)
    IF Problem is feasible
        Retain corresponding hard constraints.
    ELSE
        Remove corresponding hard constraints.
    END
END For all objectives
Solve resulting NLP with hard constraints corresponding to achievable objectives to global optimality.
Implement optimal control sequence.
```

Algorithm 1. Pseudo-Code for the Hard Constraint Formulation for Prioritized Objective Control

a given process can be met, particularly those of high priority. Under this assumption, it could be beneficial in the optimal case where all objective can be met. If feasible, the remaining problems (feasibility checks) can be ignored. If all control objectives are indeed feasible, the algorithm needs only to consider and solve a single NLP.

4 Nonconvex Optimization

An integral piece of any MPC algorithm is the optimization problem solution method. For nonlinear formulations this is of particular importance. Local solution methods can be used. This can leave the strategy susceptible to suboptimal solutions. Gradient-based methods can become trapped in local minima, thus the optimality of the solution may be dependent on the initial value. In an attempt to address the nonconvex problem, globally convergent stochastic methods can be employed. In this work, the feasible solution space is randomly searched and probabilistic arguments support the convergence of the algorithm to solution.

NLP solution serves two purposes in this NMPC formulation. The first function of this tool is to check the feasibility of a given NLP to determine whether a control objective can be met in order of priority. In this instance, an optimal solution is not needed. Here, the search need only provide any feasible solution (upper bound) or sufficiently search the solution space and exhaust the possibility that a feasible solution exists. For the stochastic search, as the number of points considered approaches infinity, the global feasibility can be guaranteed. This semblance of global optimality provided by the random search is important, as local methods can only be relied upon to indicate that a problem is feasible. Indication of local infeasibility fails to provide information of global feasibility.

The last step in the algorithm is to determine the global optimum of the final NLP that best accounts for the control objective prioritization. In this case, the stochastic approach is used to solve this single problem to “global optimality”. For this particular NLP, the solution space is again randomly searched. The best solution found in the random search is used as the starting point for a local gradient based solution. This effectively determines the global solution, provided that the solution space is adequately sampled. However, no guarantee can be made for samples of finite size. In this context, the solution provides an optimal control sequence that assures that all control objectives are logically handled in order of their priority and those that cannot be met absolutely have their violations minimized based on their appearance in the traditional objective function.

Alternatively, established deterministic methods that provide a rigorous guarantee on global optimality can be considered. Deterministic methods for global optimization typically rely on the generation of convex relaxations of the original nonconvex problem. These convex relaxations are constructed in a number of ways. One such approach, the αBB method [2], handles general twice-differentiable nonconvex functions. This method relies on the determination of the minimum eigenvalue for the Hessian of the nonconvex function over the region of interest, however does not require additional variables or constraints in the formulation. An alternative approach [11, 16] generates convex functions using the known convex envelopes of simple nonlinear functions. The original nonconvex problem is reformulated to a standard form with constraints involving simple nonlinear functions by the introduction of new variables and new constraints. This analysis is achieved by recursively simplifying terms in the function tree expression by introduction of new simple nonlinear expressions. The new simple nonlinear functions explicitly define new variables in terms of other variables. A resulting nonconvex equality constraint can then be replaced by convex inequality constraints. A detailed explanation of these methods and some comparison of the methods is given in [5].

Upon creation of the linear relaxation for the nonconvex nonlinear problem, the branch-and-reduce method [15] can be implemented. This is an extension of the traditional branch-and-bound method with bound tightening techniques for accelerating the convergence of the algorithm. 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 [1, 15, 16] or interval analysis techniques [12]. These techniques help to derive tighter variable bounds for a given partition in the search tree. 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. This approach was applied in a NMPC framework [10].

5 Case Study

The proposed prioritized objective nonlinear model predictive control algorithm is demonstrated on a simulated multivariable network of air pressure tanks. A complete description of the system and closed-loop results are presented below. Consider a simulated multivariable network of air pressure tanks.

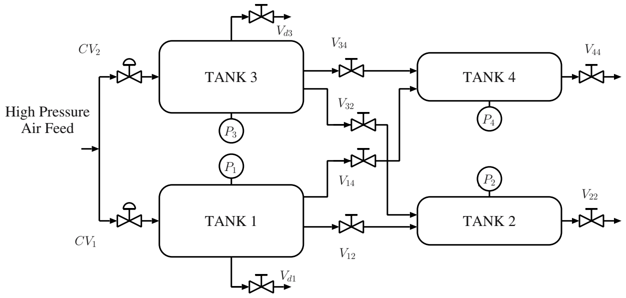

Fig. 1. Schematic of the Network of Pressure Tanks

Supply air is fed to the system at 60 psig through two control valves which act as the manipulated variables (u_1 and u_2) in the control problem. Pressure gradients drive the flow of air through the system. The air flows through the four tanks that are interconnected, while the numerous valves throughout the system dictate the direction of the flow. After traveling through the system, the air ultimately exits the downstream tanks to the atmosphere. It is assumed that the pressure in each of the four tanks can be measured. These will act as the process outputs to be controlled. The non-square nature of this configuration (2×4) lends itself well to demonstrating the ability of this specific controller as all four measurements cannot be maintained at setpoint using only two manipulated variables, thus forcing the controller to decide the appropriate trade-off based on the prioritized objectives.

For this work, it is assumed that the flow of air across a given valve (v_i) can be defined as:

$$f_i(k) = c_i\sqrt{\Delta P_i(k)} \tag{4}$$

where f_i is a molar flow, k is the sample time, c_i is a proportionality constant related to the valve coefficient, and ΔP_i is the pressure drop across the valve. For this study, it is assumed that there is no reverse flow across a valve, forcing ΔP_i non-negative. Under ideal conditions, the discrete time governing equations defining the pressures in each tank are taken as:

$$\begin{aligned}
P_1(k+1) &= \tfrac{hP_1(k)}{V_1}\left(u_1 f_{CV1} - \gamma_1 f_{12} - (1-\gamma_1) f_{14}\right) + P_1(k)\\
P_2(k+1) &= \tfrac{hP_2(k)}{V_2}\left(\gamma_1 f_{12} + (1-\gamma_2) f_{32} - f_{22}\right) + P_2(k)\\
P_3(k+1) &= \tfrac{hP_3(k)}{V_3}\left(u_2 f_{CV2} - \gamma_2 f_{34} - (1-\gamma_2) f_{32}\right) + P_3(k)\\
P_4(k+1) &= \tfrac{hP_4(k)}{V_4}\left(\gamma_2 f_{34} + (1-\gamma_1) f_{14} - f_{44}\right) + P_4(k)
\end{aligned} \tag{5}$$

where γ_1 and γ_2 define the fractional split of air leaving the upstream tanks. Here, this set of equations represents both the nonlinear process and the model used for control purposes. The sampling period (h) used was 3 minutes. This is important as it defines the time limit in which each optimization problem must be solved for real-time operation. The parameter values used in this study are summarized in Table 1.

Table 1. Model Parameters for the Simulated Network of Pressure Tanks

$V_1 = 8$	$V_3 = 8$	$c_{CV1} = 0.25$	$c_{12} = 0.02$	$c_{22} = 0.06$	$c_{44} = 0.06$	$\gamma_1 = 0.5$
$V_2 = 5$	$V_4 = 5$	$c_{CV2} = 0.25$	$c_{14} = 0.05$	$c_{34} = 0.02$	$c_{32} = 0.05$	$\gamma_2 = 0.3$

6 Closed-Loop Results

The performance of the proposed control algorithm is tested on the simulated pressure tank network. Its ability to appropriately handle control objective prioritization through a number of reference transitions and in the presence of disturbance loads is demonstrated. A number of control objectives are defined and assigned a relative priority. These are summarized in Table 2. Assume that for safety concerns, it is important that the pressure in the upstream tanks are kept below a pressure of 60 psig. These constraints are given highest priority. A secondary goal is the regulation of the pressure in second tank (P_2). It is desirable for this tank pressure to closely track setpoint, and thus a setpoint constraint as well as tight upper and lower bounds (± 2 psig from setpoint) are imposed. Note that the lower and upper bounds are assigned to be priority 3 and 4 respectively, while the setpoint constraint is not considered for objective prioritization and use of hard constraints. Subsequent control objectives include a lower bound on the pressure in tank 1 and bounds on the pressure in tank 4. The pressure in tank 3 is left unconstrained. All constraints include a 15 minute delay for enforcement.

Table 2. Summary of Prioritized Control Objectives (* Note that a hard constraint corresponding to the setpoint control objective is not used.)

Relative Priority	Variable Constrained	Constraint Type	Constraint Value	Relative Priority	Variable Constrained	Constraint Type	Constraint Value
1	P_1	UB	60	6	P_4	UB	30
2	P_3	UB	60	7	P_4	LB	20
3	P_2	LB	25/20/25	8	P_4	LB	23
4	P_2	UB	29/24/29	9*	P_2	SP	27/22/27
5	P_1	LB	55				

The controller is tuned to with $m = 2$ and $p = 20$. Each control objective is assigned a weight of $\Gamma_e = 100$ and the input movements are not penalized, $\Gamma_u = 0$. These weights are used to determine the tradeoffs between soft constraint violations of the various control objectives for which the hard constrained problem cannot be solved. For this example, at each time step, the appropriate NLP is solved using a stochastic approach followed by a local gradient based search. Specifically, 1000 points in the solution space are considered, the best of which is used as the starting point for the gradient-based solution. This stochastic and gradient-based solution process is repeated 3 times and the best solution is taken as the optimal control sequence to be implemented.

At $t = 150$ minutes, a setpoint change for the pressure in tank 2 steps from its initial value of 27 psig to 22 psig. The controller recognizes the change and

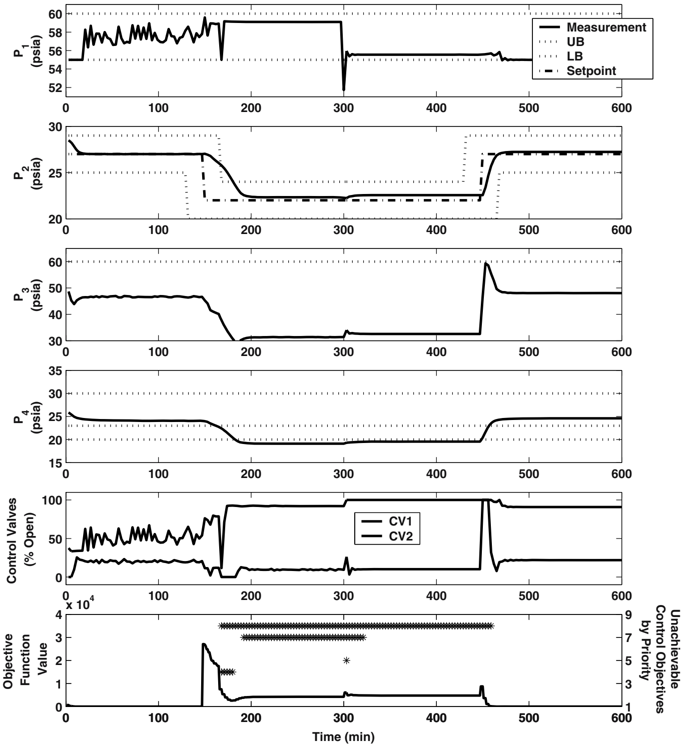

Fig. 2. Closed-loop Results of the Simulated Network of Pressure Tanks being Controlled by the Hard Constraint formulation for Prioritized Objective NMPC, Objective Function Values, and Unachievable Control Objectives

begins moving the system to accommodate it (Figure 2). Note that the bounds associated with P_2 are of high priority (priority 3 and 4). The move to satisfy these constraints requires the controller to violate the lower bounds on the pressure in tank 4 (priority 7 and 8). This means that imposing hard constraints corresponding to these control objectives renders the associated NLP infeasible. The control objectives that lead to NLP infeasibility and the objective function of the solution at each time step is seen in Figure 2. During this transition, the controller is also unable to move the system fast enough to avoid briefly violating the upper bound on the pressure in tank 2 (priority 4). The controller does however, track the new setpoint.

At $t = 300$ minutes, a disturbance is imposed on the system by simulating a leak in the first tank. The controller cannot respond to the disturbance fast enough to avoid the lower bound on the pressure in this tank (priority 5). However the controller is able to quickly return the pressure within the limiting values. This action saturates one of the inputs. Again, P_2 is maintained within its limits but is unable to track the setpoint in an offset free position. At $t = 450$ minutes, while still operating under a disturbance, P_2 is returned to its original setpoint value of 27 psig. The controller recovers the ability to meet the all control objectives associated with P_4. However, the controller does ride the lower bound constraint associated with P_1.

Note that the indication of feasibility in this formulation is based on the ability for the controller to find control moves that can maintain the model predicted outputs within their constraint limits by driving the error to zero. However, the model predicted values are not always necessarily equivalent to the true process values (measurements). This plant-model mismatch is inherent in the accuracy of models developed through the identification process and is often exaggerated in the presence of an unmeasured disturbance. For this reason, more conservative bounds should be used to insure that the desired limits are enforced.

7 Conclusions

A Nonlinear Model Predictive Control (NMPC) algorithm that utilizes hard variable constraints for control objective prioritization has been proposed. The formulation requires the solution of only a minimal number of NLP's as opposed to a complex MINLP. A stochastic approach is utilized to check problem feasibility and to find the optimum of the resulting nonconvex NLP's. This alleviates the shortcomings of purely local gradient based methods as it better searches the solution space for the global optimum. However, optimality can only be rigorously guaranteed using existing deterministic methods. The controller was shown to be effective in appropriately handling control objectives of varying importance in a simulated multivariable network of pressure tanks.

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

[1] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas. Global Optimization of Mixed-Integer Nonlinear Problems. *AIChE J.*, 46(9):1769–1797, 2000.

[2] 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.

[3] A. Bemporad and M. Morari. Control of Systems Integrating Logic, Dynamics, and Constraints. *Automatica*, 35:407–427, 1999.

[4] E. P. Gatzke and F. J. Doyle III. Multi-Objective Control of a Granulation System. *Journal of Powder Technology*, 121(2):149–158, 2001.

[5] 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.

[6] ILOG. *ILOG CPLEX 9.1: User's Manual*. Mountain View, CA, April 2005.
[7] E. C. Kerrigan, A. Bemporad, D. Mignone, M. Morari, and J. M. Maciejowski. Multi-objective Prioritisation and Reconfiguration for the Control of Constrained Hybrid Systems. In *Proceedings of the American Controls Conference*, Chicago, Illinois, 2000.
[8] C. E. Long and E. P. Gatzke. Globally Optimal Nonlinear Model Predictive Control. In *DYCOPS 7, the 7th International Symposium on Dynamics and Control of Process Systems*, Boston, MA, 2004.
[9] C. E. Long and E. P. Gatzke. Model Predictive Control Algorithm for Prioritized Objective Inferential Control of Unmeasured States Using Propositional Logic. *Ind. Eng. Chem. Res.*, 44(10):3575–3584, 2005.
[10] C. E. Long, P. K. Polisetty, and E. P. Gatzke. Nonlinear Model Predictive Control Using Deterministic Global Optimization. *J. Proc. Cont.*, accepted, 2005.
[11] G. P. McCormick. Computability of Global Solutions to Factorable Nonconvex Programs: Part I - Convex Underestimating Problems. *Mathematical Programming*, 10:147–175, 1976.
[12] R. E. Moore. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.
[13] R. S. Parker, E. P. Gatzke, and F. J. Doyle III. Advanced Model Predictive Control (MPC) for Type I Diabetic Patient Blood Glucose Control. In *Proc. American Control Conf.*, Chicago, IL, 2000.
[14] A. N. Reyes, C. S. Dutra, and C. B. Alba. Comparison of Different Predictive Controllers with Multi-objective Optimization: Application to An Olive Oil Mill. In *Proceedings of the 2002 IEEE International Conference on Control Applications and International Symposium on Computer Aided Control Systems Designs*, pages 1242–1247, Glasgow, UK, 2002.
[15] 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.
[16] E. M. B. Smith. *On the Optimal Design of Continuous Processes*. PhD thesis, Imperial College, London, 1996.
[17] M. L. Tyler and M. Morari. Propositional Logic in Control and Monitoring Problems. *Automatica*, 35:565–582, 1999.
[18] J. Vada, O. Slupphaug, and T. A. Johansen. Optimal Prioritized Infeasibility Handling in Model Predictive Control: Parameteric Preemptive Multiobjective Linear Programming Approach. *Journal of Optimization Theory and Applications*, 109(2):385–413, 2001.
[19] J. Vada, O. Slupphaug, T. A. Johansen., and B. A. Foss. Linear MPC with Optimal Prioritized Infeasibility Handling: Application, Computational Issues, and Stability. *Automatica*, 37:1835–1843, 2001.