Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Predictive Control of Nonlinear Hybrid Systems Using Generalized Outer Approximation

Naresh N. Nandola and Sharad Bhartiya

Department of Chemical Engineering, Indian Institute of Technology Bombay
Mumbai 400 076. India (Tel: +91-22-2576-7225;e-mail:naresh_nandola@iitb.ac.in; bhartiya@ che.iitb.ac.in)

Abstract: This paper presents an efficient optimization algorithm for mixed integer nonlinear programming (MINLP) problem resulting from multiple partially linearized (MPL) model based control of nonlinear hybrid dynamical system (NHDS). The algorithm uses structural information of the canonical MPL framework and derives comparatively easier quadratic programming (QP) primal problem as well as an MILP master problem for generalized outer approximation (GOA) algorithm, a decomposition based solution strategy for MINLP. Computational efficiency of the algorithm over the branch and bound strategy is demonstrated using a simulated benchmark three-spherical tank system.

1. INTRODUCTION

Hybrid systems are used to describe processes that involve continuous dynamics in addition to discrete (logical) decisions (Branicky *et al.*, 1998; Bemporad and Morari, 1999) and have found applications in manufacturing systems, automobile control, and computer disk drive control among others. Although the use of a hybrid system framework in modeling and control of chemical processes has emerged only recently, large continuous plants have always used logic controllers to implement safety features such as the triggering of a coolant pump and the various safety interlocks. However, current trends in the chemical process industry emphasize the need for flexible processing, which invariably necessitates a greater degree of logical decision-making along with the continuous control laws.

A number of modeling formalisms that represent nonlinear hybrid dynamical systems (NHDS) have been proposed in literature (Branicky *et al.*, 1998; Engell, 1998; van der Schaft and Schumacher, 2000; Buss *et al.*, 2002). These formalisms can be broadly assigned to the following three categories (Kowalewski, 2002): (i) a discrete formalism, such as finite automata, that can be extended with continuous variables resulting in hybrid frameworks such as timed automata and hybrid Petri Nets, (ii) a continuous formalism that can accommodate discrete variables or logical conditions by appropriately switching between system dynamics, and (iii) an approach that directly combines the continuous subsystem with its discrete counterpart through an interface. These models play a key role in various aspects of hybrid systems such as simulation, verification, identification, optimization and control. Optimal feedback control, such as model predictive control, of NHDS is challenging since this typically requires an online solution of a mixed integer nonlinear/quadratic program (MINLP/MIQP) within a small fraction of the sampling period. This impediment to control of NHDS can be addressed along three paths: (i) efficient representation of the NHDS, (ii) efficient algorithms for solution of MINLP/MIQP (iii) enhanced computer speed.

Our earlier work (Nandola and Bhartiya, 2008) addressed the first aspect by modeling the NHDS using a multiple, partially linearized (MPL) scheme. Each linearized model is a local representation of all locations of the hybrid system. These models are then combined using Bayes theorem to describe the nonlinear hybrid system. The multiple models, which consist of continuous as well as discrete variables, are used for synthesis of a model predictive control (MPC) law. The MPC formulation takes on a similar form as that used for discrete-time control of a continuous variable system. Although implementation of the control law requires an online solution of an MINLP, the optimization problem has a fixed structure with certain computational advantages. These advantages of the MPL model over the mixed logical dynamical (MLD) model (Bemporad and Morari, 1999) were demonstrated for servo control of a benchmark three tank system, where all integer programs were solved using a branch and bound (BB) strategy.

The current work examines the second impediment to control of NHDS by exploiting the structure of the MINLP resulting from the MPL model based predictive control law. We show that this MINLP is particularly suited for decomposition approaches such as generalized outer approximation (GOA) and generalized bender decomposition (GBD). Specifically, we show that for the MPL based control law the primal problem of the GOA algorithm is reduced to a quadratic program (QP) while the master problem retains its mixed integer linear program (MILP) form. Thus the GOA method reduces to solving a series of QPs and MILPs to obtain the solution of the original MINLP. Both of these subproblems are comparatively less expensive than the NLP solutions needed when using the BB strategy. Moreover, the fixed structure of the MINLP enables us to derive analytical gradients for the nonlinear objective function and constraints, which can be used in the master problem to further speed up the solution. The benefits of this algorithm relative to the BB strategy are demonstrated on a three-spherical tank system. The results confirm significant computational advantage relative to previous results in Nandola and Bhartiya (Nandola and Bhartiya, 2008) that used a BB strategy. The current paper is organized as follows: Section 2 reviews MPL modeling and control for nonlinear hybrid dynamic (NHDS) system. Generalized outer approximation (GOA) algorithm in perspective of our MPL model based predictive control is discussed in Section 3. Section 4 demonstrates the applicability of GOA algorithm for MPL model based control of NHDS on three spherical tank system. Finally, the work is concluded in Section 5.

2. MODELING AND CONTROL USING MULTIPLE PARTIALLY LINEARIZED (MPL) MODEL

Hybrid systems may involve both continuous and discrete states as well as continuous and discrete inputs. Typically, the flow-field describing the evolution of continuous states is dependent on discrete phenomena characterized by discrete state events as well as control events due to discrete inputs. Buss *et al.* (Buss *et al.*, 2002) modelled the nonlinear hybrid system by introducing discrete states ($\boldsymbol{x}^d$) as well as discrete control inputs in addition to the continuous states ($\boldsymbol{x}^c$) and control inputs ($\boldsymbol{u}^c$). Their model forms the starting point for multiple partially linearized (MPL) model for NHDS. The continuous states of the hybrid system evolve based on the flow-field f_l , which is dependent on the location l of the system. Upon occurrence of an event, the system jumps to a new location l' which results in a change in the flow-field to $f_{l'}$. To enable identification of the different locations and the transitions between them, suitable event generating functions s_j, j = 1, 2,… ,n_s are defined. When one or more of these functions take on a value of zero, that is $s_j = 0$, an event is said to occur. State events as well as control events are considered, both of which are identified by event generating functions s_j.

In order to obtain a control relevant model by representing all locations of NHDS by a global flow-field, Nandola and Bhartiya (Nandola and Bhartiya, 2008) modified the HSM for switched hybrid system by assigning binary indicator variables $\delta_j \in \{0,1\}, j = 1,...,n_s$ to each event generating function s_j . The resulting logical expression was then converted into linear inequalities using equivalence with propositional logic expressions (such as Big-*M* constraints; see, for example, Williams (Williams, 1993) and Raman and Grossmann (Raman and Grossmann, 1991)). Thus, the modified HSM is written as follows,

$$\dot{\boldsymbol{x}}^c = f_g(\boldsymbol{x}^c, \boldsymbol{u}^c, \boldsymbol{\delta}) \tag{1}$$

$$\boldsymbol{x}^d(t) = \boldsymbol{b}_d\left(\boldsymbol{\delta}(t)\right) \tag{2}$$

$$\boldsymbol{E}_1\boldsymbol{u}^c(t) + \boldsymbol{E}_2\boldsymbol{\delta}(t) + \boldsymbol{E}_3\boldsymbol{x}(t) \le \boldsymbol{E}_4 \tag{3}$$

$$\boldsymbol{x}^c(t^+) = \boldsymbol{x}^c(t^-) \tag{4}$$

$$\boldsymbol{x}^d(t^+) = \boldsymbol{b}_d\left(\boldsymbol{\delta}(t^+)\right) \tag{5}$$

where f_g is a global flow-field that subsumes all location-dependent flow-fields f_l and is parameterized by the indicator vector $\boldsymbol{\delta}$, whose elements are determined by inequality (3). Superscripts + and - indicate values of states just after and before occurrence of an event, respectively. Matrices $\boldsymbol{E}_i$ (i=1,2,3,4) are constant coefficient matrices and vector $\boldsymbol{b}_d$ is a function of binary variables $\boldsymbol{\delta}$. Note that the vector $\boldsymbol{\delta}$ with n_s binary elements can describe 2^{ns} locations. A change in the status of one or more elements of $\boldsymbol{\delta}$ corresponds to an event that may be triggered by discontinuity in states (that is, a State Event (SE)) and/or discontinuity in inputs (that is, a Control Event (CE)). If the model described by (1)-(5) is used for model predictive control, an online solution of an MINLP is required. In the remainder of this section, we describe an approximation of the model by using a multiple partially linearized (MPL) model scheme. The chief advantage of the MPL model is that although the controller law continues to be an MINLP, it takes on a canonical form for which efficient optimization algorithms can be tailored.

Remark 1: Elements of $\boldsymbol{\delta}$ describe discrete control inputs in addition to discrete states (see (2)) of the NHDS. Thus, a change in the status of discrete control inputs triggers a control event which can be identified from the status of binary indicator variables $\delta_j \in \{0,1\}, j = 1,...,n_s$.

The MPL modeling begins with performing a Taylor series expansion on (1) at a point of continuous variable ($\boldsymbol{x}^c$, $\boldsymbol{u}^c$) and retaining binary vector $\boldsymbol{\delta}$ as a parameter. Thus, one can obtained linearized model whose *system matrices are a function of the binary variables* $\boldsymbol{\delta}$*.* Next, the linearized model is discretized in the time domain, which in turn will enable writing the prediction equations needed in MPC. Discretization of the linearized model starts with fixing the value of binary variables $\boldsymbol{\delta}$ in the system matrices of the linearized model, thereby obtaining a model for a system at a fixed location. Since n_s binary variables result in 2^{n_s} possible locations, one can obtain 2^{n_s} discrete time linear models from (1) and (2). These 2^{n_s} models are then combined using a corresponding scalar logical multiplier (Colmenares *et al.*, 2001) ℓ_i to produce a linearized discrete-time representation of (1)-(3) as follows,

$$\boldsymbol{x}_{k+1} = \left(\sum_{i=1}^{2^{n_s}} \ell_{i,k}\boldsymbol{\Phi}_i\right)\boldsymbol{x}_k + \left(\sum_{i=1}^{2^{n_s}} \ell_{i,k}\boldsymbol{\Gamma}_i\right)\boldsymbol{u}_k^c + \left(\sum_{i=1}^{2^{n_s}} \ell_{i,k}\boldsymbol{f}_{di}\right) \tag{6}$$

$$\boldsymbol{E}_1\boldsymbol{u}_k^c + \boldsymbol{E}_2\boldsymbol{\delta}_k + \boldsymbol{E}_3\boldsymbol{x}_k \le \boldsymbol{E}_4 \tag{7}$$

where $\boldsymbol{x}_k = \left[\boldsymbol{x}_k^c \quad \boldsymbol{x}_k^d\right]^T$. The logical multiplier ℓ_i is defined using the indicator variables $\boldsymbol{\delta}_k$, and is designed to take on a value 1 if and only if the i^{th} combination of the binary variables is encountered and zero, otherwise.

Remark 2: The deviation form of variables has not been used as this allows representation of non-equilibrium operation, a common feature in hybrid systems applications, resulting in the affine representation.

Note that the RHS of (6) consists multiplicative terms between binary variable $\boldsymbol{\delta}_k$, state variables $\boldsymbol{x}_k^c$ and inputs $\boldsymbol{u}_k^c$ and thus the model is nonlinear. These multiplicative terms can be masked by introducing auxiliary binary and auxiliary continuous variables and its corresponding constraints to recast (6) and (7) into MLD model (Bemporad and Morari, 1999). However, the increased size of MLD imposes large computational burden in its use. On the other hand retaining it in the nonlinear form as shown in (6) and (7) is computationally efficient as it requires fewer number of variables and constraints. In addition, (6) can be represented in a compact vector form. Thus, (6) and (7) take the form,

$$\boldsymbol{x}_{k+1} = (\bar{\boldsymbol{L}}_k \bar{\Phi})\boldsymbol{x}_k + (\bar{\boldsymbol{L}}_k \bar{\Gamma})\boldsymbol{u}_k^c + \bar{\boldsymbol{L}}_k \bar{\boldsymbol{f}} \quad (8)$$

$$\boldsymbol{E}_1 \boldsymbol{u}_k^c + \boldsymbol{E}_2 \boldsymbol{\delta}_k + \boldsymbol{E}_3 \boldsymbol{x}_k \leq \boldsymbol{E}_4 \quad (9)$$

where $\bar{\boldsymbol{L}}_k, \bar{\Phi}, \bar{\Gamma}$ and $\bar{\boldsymbol{f}}$ are constituted from $\ell_{i,k}$, Φ_i, Γ_i and $\boldsymbol{f}_{di}$, respectively. The outputs of the linearized model may be written as follows,

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k \quad (10)$$

Note that this model describes all locations of the nonlinear hybrid system in the vicinity of a single operating point. Similar linearized discrete-time models may obtained at different operating points (Ozkan *et al.*, 2000) characterized by the continuous states and continuous inputs ($\boldsymbol{x}^c$, $\boldsymbol{u}^c$). These models are then combined using a weighting scheme such as Bayesian weighting (Schott and Bequette, 1997; Nandola and Bhartiya, 2008) to reconstitute the original nonlinear model. Thus, the overall weighted model may be written as follows,

$$\boldsymbol{x}_{k+1} = (\bar{\boldsymbol{L}}_k \bar{\Phi}_{avg})\boldsymbol{x}_k + (\bar{\boldsymbol{L}}_k \bar{\Gamma}_{avg})\boldsymbol{u}_k^c + \bar{\boldsymbol{L}}_k \bar{\boldsymbol{f}}_{avg} \quad (11)$$

$$\boldsymbol{E}_1 \boldsymbol{u}_k^c + \boldsymbol{E}_2 \boldsymbol{\delta}_k + \boldsymbol{E}_3 \boldsymbol{x}_k \leq \boldsymbol{E}_4 \quad (12)$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k \quad (13)$$

where $\bar{\Phi}_{avg}, \bar{\Gamma}_{avg}, \bar{\boldsymbol{f}}_{avg}$ are the blended system matrices that depend on weighting of the different models. Thus, (11)-(13) approximate the nonlinear operating range as well as all locations of the hybrid system. In addition, this model becomes linear at a particular location. Note that the structure of the MPL model remains unchanged for any arbitrary NHDS. Nandola and Bhartiya (Nandola and Bhartiya, 2008) derived the MPC law for NHDS which can be stated as follows,

$$\min_{\boldsymbol{\mu}_k^c, \bar{\boldsymbol{\delta}}_k} J = \begin{pmatrix} (\bar{\boldsymbol{H}}_{1k}\boldsymbol{x}_k + \bar{\boldsymbol{H}}_{2k}\boldsymbol{\mu}_k^c + \bar{\boldsymbol{H}}_{3k} - \boldsymbol{\psi}_{ref})^T \\ \boldsymbol{W}_y(\bar{\boldsymbol{H}}_{1k}\boldsymbol{x}_k + \bar{\boldsymbol{H}}_{2k}\boldsymbol{\mu}_k^c + \bar{\boldsymbol{H}}_{3k} - \boldsymbol{\psi}_{ref}) \\ +(\boldsymbol{R}\boldsymbol{\mu}_k^c - \boldsymbol{R}_0\boldsymbol{u}_{k-1}^c)^T \boldsymbol{W}_u (\boldsymbol{R}\boldsymbol{\mu}_k^c - \boldsymbol{R}_0\boldsymbol{u}_{k-1}^c) \end{pmatrix} \quad (14)$$

such that,

$$\boldsymbol{g}_1 : \bar{\boldsymbol{E}}_1\boldsymbol{\mu}_k^c + \bar{\boldsymbol{E}}_2\bar{\boldsymbol{\delta}}_k + \bar{\boldsymbol{E}}_3\left(\boldsymbol{H}_{1k\text{-}1}\boldsymbol{x}_k + \boldsymbol{H}_{2k\text{-}1}\boldsymbol{\mu}_k^c + \boldsymbol{H}_{3k\text{-}1}\right) \leq \bar{\boldsymbol{E}}_4 \quad (15)$$

$$\boldsymbol{g}_2 : \boldsymbol{\psi}_{\min} \leq \bar{\boldsymbol{H}}_{1k}\boldsymbol{x}_k + \bar{\boldsymbol{H}}_{2k}\boldsymbol{\mu}_k^c + \bar{\boldsymbol{H}}_{3k} \leq \boldsymbol{\psi}_{\max} \quad (16)$$

$$\begin{aligned} &\boldsymbol{\mu}_{\min}^c \leq \boldsymbol{\mu}_k^c \in R^{nc} \leq \boldsymbol{\mu}_{\max}^c \\ &\bar{\boldsymbol{\delta}}_k \in \{0,1\}^{nb} \end{aligned} \quad (17)$$

where $\bar{\boldsymbol{H}}_{1k}, \bar{\boldsymbol{H}}_{2k}, \bar{\boldsymbol{H}}_{3k}, \boldsymbol{H}_{1k\text{-}1}, \boldsymbol{H}_{2k\text{-}1}, \boldsymbol{H}_{3k\text{-}1}$ are variable coefficient matrices which depends on the discrete decision variables $\bar{\boldsymbol{\delta}}_k$, $\boldsymbol{\mu}_k^c$ is a controlled vectors of continuous inputs (for *m* control horizon) and $\bar{\boldsymbol{\delta}}_k$ is appropriate vector of binary variables resulting from *p* step ahead prediction for *m* control moves using (11)-(13). Vector ψ_{ref} stands for the setpoint trajectory. Matrices $\bar{\boldsymbol{E}}_1, \bar{\boldsymbol{E}}_2, \bar{\boldsymbol{E}}_3, \bar{\boldsymbol{E}}_4$ are constant coefficient matrices made up of matrices $\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3, \boldsymbol{E}_4$, respectively. Note that the objective function (14) and constraints, $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$, make the above optimization problem an MINLP. For details, the reader is referred to Nandola and Bhartiya (Nandola and Bhartiya, 2008).

Nandola and Bhartiya (Nandola and Bhartiya, 2008) used a BB algorithm with an NLP solver to solve the MINLP (14)-(17). They showed that their formulation has superior computational aspects when compared with the MLD framework. The BB algorithm performs a tree search in the space of integer variables and solves a relaxed NLP at each node where a subset of binary variables is fixed. The relaxed NLP provides a lower bound of original MINLP and this information is used to fathom the BB nodes. Thus, it solves a sequence of NLPs until all the binary variables take on value of 0-1. This method is computationally expensive for high dimensional binary variables and can be of use only for fewer binary variables or when the relaxed NLP is easy to solve (Grossmann and Kravanja, 1995). On the other hand, decomposition based algorithms such as generalized bender decomposition (GBD) (Geoffrion, 1972; Floudas, 1995) and generalized outer approximation (GOA) (Fletcher and Leyffer, 1994) have proved computationally efficient for a certain class of MINLP such as mixed-integer dynamic optimization (MIDO) (Bansal *et al.*, 2003). In the next section, we discuss the GOA method in perspective of the MINLP resulting from formulation of the MPL model based control law.

3. GENERALIZED OUTER APPROXIMATION (GOA) FOR MPL MODEL BASED CONTROL

The key idea of GOA algorithms is generation of a non-increasing upper bound and a non- decreasing lower bound by solving a series of primal problems and master problems. The primal problem is obtained by fixing the binary variables. Hence it results in an NLP whose solution represents an upper bound to the MINLP. The master problem is obtained via generation of support functions of the nonlinear objective functions as well as nonlinear constraints, using their linear approximation at current value of continuous variables (that is solution of primal problem) and binary variables. The master problem results in an MILP whose solution is a lower bound of the MINLP. The master problem also produces the value of the binary variables for next iteration. When the difference of lower and upper bounds lies within a user-defined tolerance, the algorithm terminates and the current solution of the primal problem and master problem correspond to optimal values for the continuous variables and binary variables, respectively. For detailed theoretical development and issues related to infeasibility see (Fletcher and Leyffer, 1994; Floudas, 1995). However, for highly non-convex problems, the linear approximation needed in the master problem may not result in an outer approximation of the original MINLP and hence may sometimes result in a suboptimal solution. In such problems, convexification methods for MINLP (Porn *et al.*, 1999) or augmented penalty based methods may be used prior to application of this algorithm .

3.1 Primal problem

In this work, we consider the MINLP of (14)-(17). This MINLP has a canonical structure and remains unchanged for arbitrary nonlinear hybrid system. The objective function (14) of this MINLP is nonlinear in binary variables $\bar{\boldsymbol{\delta}}_k$ and quadratic in continuous variables $\boldsymbol{\mu}_k^c$. The constraints represented by (15) and (16) are nonlinear in binary variables $\bar{\boldsymbol{\delta}}_k$ and linear with respect to continuous variables $\boldsymbol{\mu}_k^c$. Therefore, on fixing the binary variables (for example, $\bar{\boldsymbol{\delta}}_k = \bar{\boldsymbol{\delta}}_{k,i}$), the primal problem reduces to the following quadratic programming (QP) optimization problem,

$$\min_{\boldsymbol{\mu}_k^c} J_{p,i} = \left(\frac{1}{2} \left(\boldsymbol{\mu}_k^c\right)^T \boldsymbol{Q}_i \left(\boldsymbol{\mu}_k^c\right) + \boldsymbol{F}_i^T \boldsymbol{\mu}_k^c \right) \tag{18}$$

such that,

$$\boldsymbol{A}_i \boldsymbol{\mu}_k^c \le \boldsymbol{b}_i \tag{19}$$

$$\boldsymbol{\mu}_{\min}^c \le \boldsymbol{\mu}_k^c \le \boldsymbol{\mu}_{\max}^c \tag{20}$$

where $\boldsymbol{Q}_i$, $\boldsymbol{A}_i$, $\boldsymbol{F}_i$ and $\boldsymbol{b}_i$ are constant matrices and can be easily derived by fixing the value of binary variables $\bar{\boldsymbol{\delta}}_{k,i}$ in (14) -(16) as follows,

$$\boldsymbol{Q}_i = \left(\bar{\boldsymbol{H}}_{2k,i}^T \boldsymbol{W}_y \bar{\boldsymbol{H}}_{2k,i} + \boldsymbol{R}^T \boldsymbol{W}_u \boldsymbol{R} \right), \; \boldsymbol{A}_i = \begin{bmatrix} \bar{\boldsymbol{E}}_1 + \bar{\boldsymbol{E}}_3 \boldsymbol{H}_{2k-1,i} \\ \bar{\boldsymbol{H}}_{2k,i} \\ -\bar{\boldsymbol{H}}_{2k,i} \end{bmatrix}$$

$$\boldsymbol{F}_i^T = \boldsymbol{2} \begin{pmatrix} \boldsymbol{x}_k^T \bar{\boldsymbol{H}}_{1k,i}^T \boldsymbol{W}_y \bar{\boldsymbol{H}}_{2k,i} + \bar{\boldsymbol{H}}_{3k,i}^T \boldsymbol{W}_y \bar{\boldsymbol{H}}_{2k,i} + \\ \boldsymbol{d}_{k,i}^T \boldsymbol{W}_y \bar{\boldsymbol{H}}_{2k,i} - \boldsymbol{\psi}_{ref}^T \boldsymbol{W}_y \bar{\boldsymbol{H}}_{2k,i} - (\boldsymbol{u}_{k-1}^c)^T \boldsymbol{R}_0^T \boldsymbol{W}_u \boldsymbol{R} \end{pmatrix}$$

$$\boldsymbol{b}_i = \begin{bmatrix} \left(\bar{\boldsymbol{E}}_4 - \bar{\boldsymbol{E}}_2 \bar{\boldsymbol{\delta}}_{k,i} - \bar{\boldsymbol{E}}_3 \boldsymbol{H}_{1k-1,i} \boldsymbol{x}_k - \boldsymbol{H}_{3k-1,i} \right) \\ \left(\boldsymbol{\psi}_{\max} - \bar{\boldsymbol{H}}_{1k,i} \boldsymbol{x}_k - \bar{\boldsymbol{H}}_{3k,i} - \boldsymbol{d}_k \right) \\ \left(-\boldsymbol{\psi}_{\min} + \bar{\boldsymbol{H}}_{1k,i} \boldsymbol{x}_k + \bar{\boldsymbol{H}}_{3k,i} + \boldsymbol{d}_k \right) \end{bmatrix}$$

In the above QP, suffix *i* denotes GOA iteration and suffix *k* indicates time instant of the MPL based model predictive control. The QP optimization problem is easier to solve as compared to the general primal problem which is a NLP. Note that the solution of this problem results in the current best value of continuous variables (that is $\boldsymbol{\mu}_{k,i}^c$) and current upper bound (that is $J_{UB,i} = J_{p,i}$) of MINLP.

3.2 Master problem

The master problem relaxation is obtained via outer approximation of the original MINLP. This relaxation varies with the feasibility of the primal problem. For the case of a feasible primal problem, the master problem is obtained by outer approximation of nonlinear objective function (14) as well as the nonlinear constraints (15) and (16) at $\left(\boldsymbol{\mu}_{k,i}^c, \bar{\boldsymbol{\delta}}_{k,i}\right)$. On the other hand, for the case of infeasible primal problem, the master problem is obtained by considering the outer approximation of the nonlinear constraints alone without considering outer approximation of nonlinear objective function (Floudas, 1995). Thus, the master problem reduces to the following mixed integer linear programming (MILP)

$$\min_{\boldsymbol{\mu}_k^c, \bar{\boldsymbol{\delta}}_k, \alpha_{GOA}} \quad J_{m,i} = \alpha_{GOA} \tag{21}$$

$$\left.\begin{aligned} J\left(\boldsymbol{\mu}_{k,i}^c, \bar{\boldsymbol{\delta}}_{k,i}\right) + \nabla \boldsymbol{J}^T \begin{bmatrix} \boldsymbol{\mu}_k^c - \boldsymbol{\mu}_{k,i}^c \\ \bar{\boldsymbol{\delta}}_k - \bar{\boldsymbol{\delta}}_{k,i} \end{bmatrix} \le \alpha_{GOA} \\ \boldsymbol{g}\left(\boldsymbol{\mu}_{k,i}^c, \bar{\boldsymbol{\delta}}_{k,i}\right) + \nabla \boldsymbol{g}^T \begin{bmatrix} \boldsymbol{\mu}_k^c - \boldsymbol{\mu}_{k,i}^c \\ \bar{\boldsymbol{\delta}}_k - \bar{\boldsymbol{\delta}}_{k,i} \end{bmatrix} \le 0 \end{aligned}\right\} \forall i \in F.P. \tag{22}$$

$$\left.\boldsymbol{g}\left(\boldsymbol{\mu}_{k,i}^c, \bar{\boldsymbol{\delta}}_{k,i}\right) + \nabla \boldsymbol{g}^T \begin{bmatrix} \boldsymbol{\mu}_k^c - \boldsymbol{\mu}_{k,i}^c \\ \bar{\boldsymbol{\delta}}_k - \bar{\boldsymbol{\delta}}_{k,i} \end{bmatrix} \le 0 \right\} \forall i \in N.F.P. \tag{23}$$

where variable $\boldsymbol{\alpha_{GOA}}$ is a decision variable, $\boldsymbol{F.P.}$ and $\boldsymbol{N.F.P.}$ stands for feasible and infeasible primal problem, respectively. Vector $\boldsymbol{g}$ represents all nonlinear constraints shown in (15) and (16) while $\nabla \boldsymbol{J}$, $\nabla \boldsymbol{g}$ are gradients of objective function (14) and nonlinear constraints, $\boldsymbol{g}$, respectively. In general, these gradients are calculated numerically. However, the canonical structure of the MINLP enables one to write analytical gradients, which can be used for enhanced accuracy as well as efficient computation. In the simulation example presented next, we use the analytical gradients but are not presented here for brevity.

4. APPLICATION

We use the GOA algorithm for control of the three-tank benchmark system using the MPL model based predictive control strategy. To evaluate the computational advantage we compare the average time needed to solve the control problem with the BB strategy.

4.1. Spherical three-Tank system

The system consists of two independent pumps that deliver the liquid flowrates Q_1 and Q_2 to Tank-1 and Tank-2, respectively through the two control valves. Six independent solenoid (on/off) valves (V_1, V_2, V_{13}, V_{23}, V_{L1} and V_{N3}) can be manipulated to interrupt the flows into or out of the three tanks. Tank-1 and Tank-3 as well as Tank-2 and Tank-3 are connected through upper and lower pipes. In order to enhance the nonlinear behavior, we replaced the cylindrical tanks in the benchmark problem (Villa *et al.*, 2004) with spherical tanks (Nandola and Bhartiya, 2008), as shown in Figure 1.

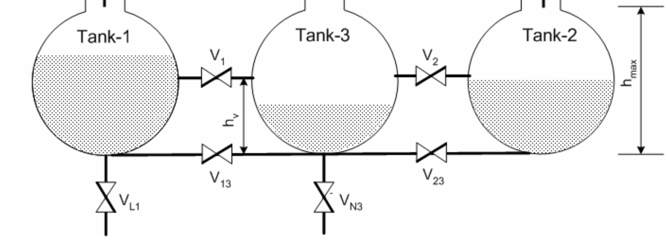

Fig. 1. Schematic of the 3-Tank benchmark problem.

The first principles model is briefly described below,

$$\pi h_1(h_{\max} - h_1)\frac{dh_1}{dt} = (Q_1 - V_{13}Q_{13V_{13}} - V_1 Q_{13V_1} - V_{L1}Q_{L_1}) \quad (24)$$

$$\pi h_2(h_{\max} - h_2)\frac{dh_2}{dt} = (Q_2 - V_{23}Q_{23V_{23}} - V_2 Q_{23V_2}) \quad (25)$$

$$\pi h_3(h_{\max} - h_3)\frac{dh_3}{dt} = \begin{pmatrix} V_{13}Q_{13V_{13}} + V_{23}Q_{23V_{23}} + V_1 Q_{13V_1} \\ + V_2 Q_{23V_2} - V_{N3}Q_N \end{pmatrix} \quad (26)$$

where $V_1, V_2, V_{13}, V_{23}, V_{L1}$ and V_{N3} represents binary indicator variables for corresponding valves, and h_{max} represents tank diameter (0.6*m*). Variables Q_i represent flowrates through valves V_i and may be evaluated using the following constitutive equations,

$$Q_{i3V_{i3}} = a_z S_{i3} sign(h_i - h_3)\sqrt{\left|2g\left(h_i - h_3\right)\right|}, \quad i = 1,2 \quad (27)$$

$$Q_{L1} = a_z S_{L1}\sqrt{2gh_1} \quad (28)$$

$$Q_N = a_z S_N \sqrt{2gh_3} \quad (29)$$

$$Q_{i3V_i} = \begin{pmatrix} V_i a_z S_i sign\left(\max\{h_i, h_v\} - \max\{h_3, h_v\}\right) \\ \times\sqrt{2g\left|\left(\max\{h_i, h_v\} - \max\{h_3, h_v\}\right)\right|} \end{pmatrix}, i = 1,2 \quad (30)$$

where S_i, S_{i3}, S_{L1} and S_N are cross sectional areas of valves and assumed identical for all valves (0.95 cm^2), h_v (see Figure 1) is height of upper pipe from bottom (0.3 m), and a_z is the discharge coefficient, which is assumed to be unity. Next, we considered three partially linearized models for the three spherical tank system. The points of linearization are listed below:

(i) Model-I: $h_1 = h_2 = 0.15m, h_3 = 0.14m, Q_1 = Q_2 = 0$

(ii) Model-II: $h_1 = h_2 = 0.25m, h_3 = 0.24m, Q_1 = Q_2 = 0$

(iii)Model-III: $h_1 = h_2 = 0.35m, h_3 = 0.34m, Q_1 = Q_2 = 0$

Model-I and Model-II correspond to levels below the upper pipe connections while Model-III corresponds to a level above the upper pipe. The three points correspond to low, medium and high levels in the three tanks. Also, the points of linearization were chosen such that the continuity of the max function is maintained. Alternatively, smooth approximation of the max function may be used.

We use the MPL model (11)-(13) based MPC for a setpoint tracking control problem, which involves filling of empty tanks to desired levels, followed by multiple setpoint changes. To study the computational efficiency of GOA algorithm, discussed in previous section, we compare the computation time with the results obtained using BB. Both cases use a sampling time t_s of 3s, prediction horizon, *p*=5 (15s) and control horizon, m=2 (6s). We consider the first principles nonlinear model (24)-(30) as the *plant* model and the MPL framework (11)-(13) as the *controller* model.

The NLP needed for solution of the MINLP using BB used subroutine *fmincon* in MATLAB 6.5 (Mathworks Inc, Natick, MA, USA). The QP and LP solution needed for the primal and master problems in GOA used subroutines *quadprog* an *linprog* in MATLAB 6.5, respectively. All of the above algorithms were based on the activeset method. All simulations have been performed on a 3.0 GHz P-IV machine with 1 GB RAM.

Figure 2 documents the results of the MPL model based control of the levels in the three tanks at different levels using both BB algorithm (dotted line) as well as GOA algorithm (solid line). Note that control results for both the cases are

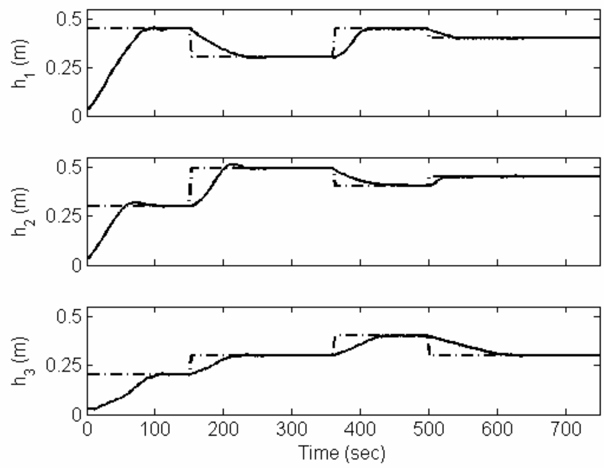

Fig. 2. MPL model based predictive control of levels h_1, h_2, and h_3 in the 3-Spherical Tank system using using GOA (solid line) and using BB (dotted line).

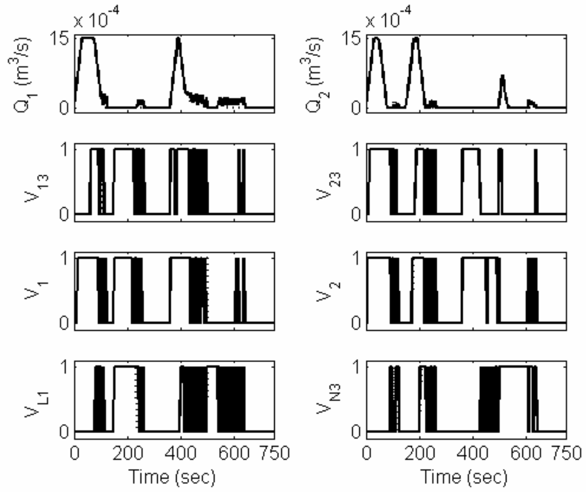

Fig. 3. Control moves for the level control problem in the 3-Spherical Tank system using GOA (solid line) and using BB (dotted line).

found to be similar and hence almost indistinguishable in the figure. Figure 3 shows the corresponding control moves for both the cases. Both the algorithms produce similar results when used to solve MINLP resulting from MPL model based control. However, the main advantage of GOA algorithm lies in its computational efficiency. BB algorithm solves a relaxed NLP at each node while GOA solves QPs and MILPs (that is, a number of relaxed LPs) at each iteration. Thus, the GOA algorithm solves comparatively easier optimization problems than BB but it requires additional computation time for linearization of the nonlinear objective function as well as constraints to obtaine the master problem. However, availability of analytical gradients reduces the computation effort. Figure 4 documents cpu time required to solve the control problem at each sampling time using BB and GOA algorithms. It is evident that the BB algorithm is unable to solve the control problem within one sampling period for about 16 % of the optimization problem. On the other hand, the GOA algorithm takes less than one sampling time for all instants. Additionally, in case of GOA, the average computation time and standard deviation are noted as 0.45s and 0.36s, respectively for this control problem and for BB, they are 1.99s and 2.8s, respectively. Ideally, values of the manipulated variables should be injected immediately upon availability of the measurement. This requires calculating control problem within a very small fraction of the sampling time. Since the GOA algorithm takes only about 15% of the sampling period on an average, it is likely to show better practical behavior relative to the BB algorithm. We have also solved the same control problem with different control horizon to check the effect of number of binary variables and observed consistent superior performance of GOA. Thus, GOA algorithm is especially suitable for real time implementation of MPL model based control of NHDS.

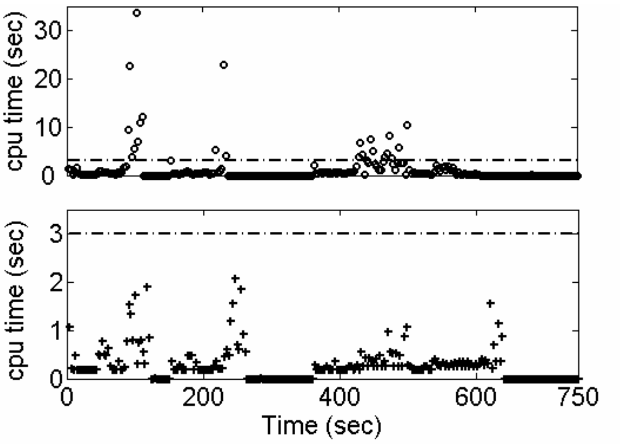

Fig. 4. Comparison of computation time to solve MINLP control problem at each sampled using GOA (+), BB (o) with respect to sampling time (dashed-dotted line)

5. CONCLUSIONS

Applications of hybrid systems are becoming increasingly common in the process industry. The main hurdle in the optimal control of hybrid systems is the requirement of an online solution to an MINLP/MIQP. The authors have shown computational advantage of the MPL model based control over multiple MLD model based control of NHDS previously. In this work, we have exploited the fixed structure of MPL framework to further augment the computational efficiency. We have used GOA algorithm and derived the primal problem of MINLP resulting from MPL model based control of NHDS. Due to the particular structure of MPL model, this primal problem reduces to a comparatively simpler QP optimization problem. We have also used analytical gradients of objective function and nonlinear constraints to derive master problem of GOA. Computational efficiency for online control of NHDS is demonstrated with this algorithm using the three-spherical tank benchmark system.

REFERENCES

Bansal, V., V. Sakizlis, R. Ross, J. D. Perkins and E. N. Pistikopoulos (2003). New algorithms for mixed-integer dynamic optimization. *Computers & Chemical Engineering* **27**(5): 647-668.

Bemporad, A. and M. Morari (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica* **35**(3): 407-427.

Branicky, M. S., V. S. Borkar and S. K. Mitter (1998). A unified framework for hybrid control: model and optimal control theory. *IEEE Transactions on Automatic Control* **43**(1): 31-45.

Buss, M., M. Glocker, M. Hardt, O. von Stryk, R. Bulirsch and G. Schmidt (2002). Nonlinear hybrid dynamical systems: modeling, optimal control, and applications. *Modelling, Analysis, and Design of Hybrid Systems*. E. Sebastian, F. Goran and E. Schnieder, Springer-Verlag**:** 311-35 BN - 3 540 43812 2.

Colmenares, W., S. Cristea, C. De Prada and T. Villegas (2001). MLD systems: modeling and control. Experience with a pilot process. *Proceedings of the 2001 IEEE International Conference on Control Applications (CCA'01), 5-7 Sept. 2001*, Mexico City, Mexico, IEEE.

Engell, S. (1998). Modelling and analysis of hybrid systems. *Mathematics and Computers in Simulation* **46**(5-6): 445-464.

Fletcher, R. and S. Leyffer (1994). Solving Mixed Integer Nonlinear Programs by Outer Approximation. *Mathematical Programming* **66**: 327.

Floudas, C. A. (1995). *Nonlinear and mixed-integer optimization : fundamentals and applications*, New York : Oxford University Press.

Geoffrion, A. M. (1972). Generalized benders decomposition. *J. Optim. Theory and Appl.* **10**(4): 237.

Grossmann, I. E. and Z. Kravanja (1995). Mixed-integer nonlinear programming techniques for process systems engineering. *Computers & Chemical Engineering* **19**(Suppl.1): 189-204.

Kowalewski, S. (2002). Introduction to the analysis and verification of hybrid systems. *Modelling, Analysis, and Design of Hybrid Systems*. E. Sebastian, F. Goran and E. Schnieder, Springer-Verlag**:** 153-71 BN - 3 540 43812 2.

Nandola, N. N. and S. Bhartiya (2008). A Multiple Model Approach for Predictive Control of Nonlinear Hybrid Systems. *Journal of Process Control* **18**(2): 131-148.

Ozkan, L., M. V. Kothare and C. Georgakis (2000). Model predictive control of nonlinear systems using piecewise linear models. *Computers and Chemical Engineering* **24**(2): 793-799.

Porn, R., I. Harjunkoski and T. Westerlund (1999). Convexification of different classes of non-convex MINLP problems. *Computers & Chemical Engineering* **23**(3): 439-448.

Raman, R. and I. E. Grossmann (1991). Relation between MILP modelling and logical inference for chemical process synthesis. *Computers & Chemical Engineering* **15**(2): 73-84.

Schott, K. D. and B. W. Bequette (1997). Multiple model adaptive control. *Multiple model approaches to modelling and control*, Taylor & Francis**:** 269-91 BN - 0 7484 0595 X.

van der Schaft, A. and H. Schumacher (2000). *An Introduction to Hybrid Dynamical Systems*. London, Springer-Verlag.

Villa, J. L., M. Duque, A. Gauthier and N. Rakoto-Ravalontsalama (2004). A new algorithm for translating MLD systems into PWA systems. *Proceedings of the 2004 American Control Conference (AAC), Jun 30-Jul 2 2004*, Boston, MA, United States, Institute of Electrical and Electronics Engineers Inc., Piscataway, NJ 08855-1331, United States.

Williams, H. P. (1993). *Model building in mathematical programming*. New York, John Wiley & Sons.