Simultaneous mixed-integer dynamic optimization for integrated design and control

Antonio Flores-Tlacuahuac∗, Lorenz T. Biegler

Department of Chemical Engineering, Carnegie-Mellon University, Pittsburgh 15213, PA, United States

Received 17 November 2005; received in revised form 25 July 2006; accepted 17 August 2006
Available online 20 September 2006

Abstract

We consider strategies for integrated design and control through the robust and efficient solution of a mixed-integer dynamic optimization (MIDO) problem. The algorithm is based on the transformation of the MIDO problem into a mixed-integer nonlinear programming (MINLP) program. In this approach, both the manipulated and controlled variables are discretized using a simultaneous dynamic optimization approach. We also develop three MINLP formulations based on a nonconvex formulation, the conventional Big-M formulation and generalized disjunctive programming (GDP). In addition, we compare the outer approximation and NLP branch and bound algorithms on these formulations. This problem is applied to a system of two series connected continuous stirred tank reactors where a first-order reaction takes place. Our results demonstrate that the simultaneous MIDO approach is able to efficiently address the solution of the integrated design and control problem in a systematic way.

© 2006 Elsevier Ltd. All rights reserved.

Keywords: Design and control; Dynamic optimization; Mixed-integer nonlinear programming

1. Introduction

Design and control are highly interdependent engineering activities and it has long been recognized that they should be performed simultaneously (Ziegler & Nichols, 1942). However, in practice both activities are performed independently of each other. Normally, process design is often tackled by chemical and process engineers, while process control is often done by electrical and instrumentation engineers. It is only when the process design is complete that control problems are addressed. Chemical processes therefore tend to be highly constrained with few degrees of freedom left for process control purposes.

Over the last 20 years, important efforts have been aimed at providing methodologies for tackling process design and control in an integrated frame-work (Bansal, Perkins, & Pistikopoulos, 2002; Sakizlis, Perkins, & Pistikopoulos, 2004; Schweiger & Floudas, 1998). Initial efforts were directed towards steady-state indicators and control indices that address potential control problems (Grossmann & Morari, 1984; Pistikopoulos & Grossmann, 1989; Straub & Grossmann, 1990; Vinson & Georgakis, 2000). However, as the operability problem features strong dynamic variations, its assessment requires that the problem be approached using a dynamic frame-work. On the other hand, the integrated design and control problem can be cast as an optimization problem. As both discrete and continuous variables are embedded into this problem, it can be formulated naturally as a mixed-integer dynamic optimization (MIDO) problem. The integer variables take care of discrete decisions (i.e. flowsheet structure, number of control loops, number of distillation columns, etc.), while the continuous variables are normally related to design variables (i.e. flows, temperatures, composition, etc.). As many real chemical processes feature strong nonlinear behavior around optimal design regions, it is likely that the MIDO problem gives rise to a highly nonlinear optimization formulation.

From an optimization point of view, two major approaches have been proposed for tackling the solution of the MIDO problem. In sequential approaches (Chatzidoukas, Perkins, Pistikopoulos, & Kiparis-sides, 2003; Sakizlis et al., 2004) propose to solve the MIDO problem as a sequence of primal and master problems using Benders decomposition or the outer approximation algorithm. Here, the primal problem corresponds to a dynamic optimization problem, solved in a sequential manner, and the master problem constructs a relaxation of the original problem based on the solutions of the primal problems. In contrast, simultaneous strategies (e.g. Biegler, Grossmann, & Westerberg, 1997) attempt to find an exact solution of the MIDO problem by combining the design and control objectives into a single optimization problem. Simultaneous approaches have the advantage of providing a unified framework for design and control, but they can be computationally expensive due to the nonlinear nature of the problem.

In this work, we propose a simultaneous approach for solving the MIDO problem. The key idea is to transform the MIDO problem into a mixed-integer nonlinear programming (MINLP) problem, which can be solved using standard optimization techniques. We also develop three MINLP formulations based on a nonconvex formulation, the conventional Big-M formulation and generalized disjunctive programming (GDP). In addition, we compare the outer approximation and NLP branch and bound algorithms on these formulations. This problem is applied to a system of two series connected continuous stirred tank reactors where a first-order reaction takes place. Our results demonstrate that the simultaneous MIDO approach is able to efficiently address the solution of the integrated design and control problem in a systematic way.

∗ Corresponding author. Tel.: +52 55 5950 4074; fax: +52 55 5950 4279.
E-mail address: antonio.flores@uia.mx (A. Flores-Tlacuahuac).
manner, while the master problem is formulated as an MILP. On the other hand, full discretization of the dynamic model has also been suggested to address the solution of the MIDO problem (Biegler, Cervantes, & Wächter, 2002). Here both state and control variables are discretized and the DAE model is solved simultaneously with the optimization problem. In this simultaneous MIDO approach, the design and control problem is cast as an MINLP for which efficient solution strategies are required.

Simultaneous MIDO studies include Balakrishna and Biegler (1993) to tackle the simultaneous reaction, energy and separation problem. They employed a dynamic optimization formulation leading to an MINLP after problem discretization. Avraam, Shah, and Pantelides (1998) employed a full discretization approach to address the dynamic optimization of hybrid systems. One of their conclusions was that the full discretization approach presented deficiencies related to the size of the discretized NLP. Bahri, Bandoni, and Rogmanoli (1997) also use the full discretization approach to address the flexibility and controllability analysis of chemical processes. The authors proposed a two-stage procedure. In the outer loop, a dynamic MINLP is solved by applying the full discretization approach. The aim of the outer loop is to solve a structural design problem. In the inner loop, the maximum constraint violation is computed for a given set of disturbances. Mohideen, Perkins, and Pistikopoulos (1996) considered a fully discretized approach for design and control, along with constraints for closed-loop stability. Fraga, Hagemann, Estrada-Villagrana, and Bogle (2000) propose a discretization approach which tries to match the dynamic behavior to some selected a priori geometric patterns. Finally, MIDO approaches are applied for batch distillation in Oldenburg, Marquardt, Heinz, and Leineweber (2003).

In this work, we propose a simultaneous approach to tackle integrated design and control cast as a MIDO problem. Because the selection of controller parameters often requires treatment of unstable dynamic responses, simultaneous formulations are essential to avoid convergence failures of the DAE model and to obtain robust performance of the optimization algorithm (Biegler et al., 2002; Flores-Tlacuahuac, Biegler, & Saldivar-Guerra, 2005). We also emphasize that the resulting MINLP problem has special characteristics, i.e., relatively few integer variables and large NLP problems due to full discretization of DAE models. As a result, the MINLP strategy needs to handle NLPs that are often large and difficult to solve. Also, MINLP formulations are required that give good lower bounds, whether in the relaxed MILP problem or branch and bound tree.

Grossmann (2002) has reviewed numerical algorithms for solving MINLPs; an overview of industrial applications of MINLP for solving planning and scheduling problems is provided by Kallrath (2000). In this study, we compare three MINLP formulations including the popular Big-M and generalized disjunctive programming (GDP) formulations (Grossmann & Biegler, 2004). We also consider the outer approximation (OA) and NLP branch and bound MINLP algorithms. All formulations and algorithms will be demonstrated with a case study for the design and control of a two CSTR system. This comparison illustrates the need for careful consideration of the characteristics of MIDO problems.

2. Simultaneous approach for design and control

Starting from the differential–algebraic equations (DAEs) of the dynamic system, we discretize both the state and control variables. Because fast variations can arise in these profiles, the whole solution space is commonly divided into time intervals called finite elements. Inside each finite element, the differential–algebraic equations are satisfied at Radau collocation points. This approach corresponds to a fully implicit Runge–Kutta method with high order accuracy and stability properties. Other methods based on different collocation discretizations (Betts, 2001) and backward difference formulae (Jockenhövel, Biegler, & Wächter, 2003) have also been used. With this discretization the dynamic optimization problem is therefore cast as an NLP. With the addition of binary decision variables, the MIDO problem is then transformed into an MINLP. Formulations and algorithms for the solution of MINLP problems have been surveyed and discussed by Grossmann (2002). Most of these strategies work well for MINLP where the combinatorial features (e.g., solution of the MILP subproblem) dominate the cost of the algorithm. However, in face of highly nonlinear behavior these approaches have not been fully analyzed. Moreover, while spatial branch and bound codes (Sahinidis, 1996) have been advanced for the global optimal solution of MINLPs, we will consider only local algorithms here because of the large size of the NLP subproblems.

2.1. MIDO problem

Disturbance rejection is a key requirement for the closed-loop control of chemical processes. In order to get minimum time closed-loop disturbance rejection, the following optimization problem can be formulated:

\[
\min \int_0^t ||z(t) - \hat{z}(t)||^2 dt
\]

s.t. Semi-explicit DAE model:

\[
\frac{d\mathbf{z}(t)}{dt} = \mathbf{F}(\mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t), t, \mathbf{p} )
\]

\[0 = \mathbf{G}(\mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t), t, \mathbf{p} )\]

Initial conditions:

\[\mathbf{z}(0) = \mathbf{z}^0\]

Bounds:

\[\mathbf{z}^L \leq \mathbf{z}(t) \leq \mathbf{z}^U\]

\[\mathbf{x}^L \leq \mathbf{x}(t) \leq \mathbf{x}^U\]

\[\mathbf{u}^L \leq \mathbf{u}(t) \leq \mathbf{u}^U\]

\[\mathbf{p}^L \leq \mathbf{p} \leq \mathbf{p}^U\]

Disjunctions:

\[\forall j \in \mathcal{I} \{ \mathbf{a}_j \leq \mathbf{g}_j(\mathbf{w}) \leq \mathbf{b}_j \} \]
where \( \mathbf{F} \) is the vector of right-hand sides of differential equations in the DAE model, \( \mathbf{G} \) the vector of algebraic equations, assumed to be index one, \( t \in [0,f] \) the time, \( \mathbf{z} \) the differential state vector, \( \mathbf{z}^0 \) the initial values of \( \mathbf{z} \), \( \dot{z} \) the set-point vector, \( \mathbf{x} \) the algebraic state vector, \( \mathbf{u} \) the control profile vector and \( \mathbf{p} \) is a time-independent parameter vector. Also, we define \( \mathbf{w} = [\mathbf{z}^T, \mathbf{x}^T, \mathbf{u}^T, \mathbf{p}^T]^T \) and \( \Omega \) is the set of disjunctions with the inequality constraints having the property \( g_i(\mathbf{0}) = 0 \) for the \( i \)th disjunction. These disjunctions can be derived systematically using the logical expressions presented in Biegler, Grossmann, and Westerberg (1997).

2.2. Discretization of DAE system

The MIDO problem is converted into an MINLP by first approximating state and control profiles by a family of polynomials on finite elements \((t_0 < t_1 < \cdots < t_N = 0)\). Here, we use a monomial basis representation for the differential profiles, as follows:

\[
\mathbf{z}(t) = \mathbf{z}_{i-1} + h_i \sum_{q=1}^{n_{col}} \Omega_q \left( \frac{t - t_{i-1}}{h_i} \right) \frac{d\mathbf{z}}{dt_{i,q}}
\]

where \( \mathbf{z}_{i-1} \) is the value of the differential variable at the beginning of element \( i \), \( h_i \) the length of element \( i \), \( d\mathbf{z}/dt_{i,q} \) the value of its first derivative in element \( i \) at the collocation point \( q \) and \( \Omega_q \) is the polynomial of order \( n_{col} \), satisfying

\[
\Omega_q(0) = 0 \quad \text{for} \quad q = 1, \ldots, n_{col}
\]

\[
\Omega_q(\rho_r) = \delta_{q,r} \quad \text{for} \quad q, r = 1, \ldots, n_{col}
\]

where \( \rho_r \) is the location of the \( r \)th collocation point within each element. Continuity of the differential profiles is enforced by

\[
\mathbf{z}_i = \mathbf{z}_{i-1} + h_i \sum_{q=1}^{n_{col}} \Omega_q \left( \frac{t - t_{i-1}}{h_i} \right) \frac{d\mathbf{z}}{dt_{i,q}}.
\]

We use Radau collocation points because they allow constraints to be set easily at the end of each element and they help stabilize the system when high index DAEs are present. In addition, the control and algebraic profiles are approximated using a similar monomial basis representation, which takes the form:

\[
\mathbf{x}(t) = \sum_{q=1}^{n_{col}} \psi_q \left( \frac{t - t_{i-1}}{h_i} \right) x_{i,q}
\]

\[
\mathbf{u}(t) = \sum_{q=1}^{n_{col}} \psi_q \left( \frac{t - t_{i-1}}{h_i} \right) u_{i,q}.
\]

Here, \( x_{i,q} \) and \( u_{i,q} \) represent the values of the algebraic and control variables, respectively, in element \( i \) at collocation point \( q \). \( \psi_q \) is the Lagrange polynomial of order \( n_{col} \) satisfying

\[
\psi_q(\rho_r) = \delta_{q,r} \quad \text{for} \quad q, r = 1, \ldots, n_{col}.
\]

From (7), the differential variables are required to be continuous throughout the time horizon, while the control and algebraic variables are allowed to have discontinuities at the boundaries of the elements. It should be mentioned that with (7), bounds on the differential variables are enforced directly at element boundaries; they can be enforced at all collocation points by writing additional point constraints.

The integral objective function is approximated with Radau quadrature with \( n_e \) finite elements and \( n_{col} \) quadrature points in each element. This leads to the following objective function:

\[
\min \Phi = \sum_{i=1}^{n_e} h_i \sum_{j=1}^{ncol} \omega_j ||z(t_{i,j}) - \tilde{z}||^2.
\]

2.3. Constraint formulation of disjunctions

As described in Grossmann (2002), disjunctions can be formulated in a number of ways through the addition of a vector of binary variables, \( \mathbf{y} \). The simplest approach is to represent the disjunctions as:

\[
\sum_j y_j (g_j(\mathbf{w}) - b_j) \leq 0
\]

\[
\sum_j y_j (a_j - g_j(\mathbf{w})) \leq 0
\]

\[
\sum_j y_j = 1
\]

\[
y_j \in [0, 1], j \in D.
\]

However, this formulation introduces nonconvex constraints. The second approach is the traditional Big-M method where the disjunction are represented as:

\[
a_j + M(1 - y_j) \leq g_j(\mathbf{w}) \leq b_j + M(1 - y_j)
\]

\[
\sum_j y_j = 1
\]

\[
y_j \in [0, 1], j \in D.
\]

Finally, we consider a variable disaggregation approach for generalized disjunctive programming (GDP). Here, we define separate variables \( \mathbf{w}_j \) for each disjunction and write the constraints:

\[
a_j y_j \leq g_j(\mathbf{w}_j) \leq b_j y_j
\]

\[
\sum_j \mathbf{w}_j = \mathbf{w}
\]

\[
\sum_j y_j = 1
\]

\[
y_j \in [0, 1], j \in D.
\]

The effectiveness of these formulations is usually evaluated by comparing feasible regions that are generated when the binary variables are relaxed to continuous variables. Smaller regions that envelop the disjunctions lead to tighter lower bounds and better performance in solving the MINLP. The first formulation is usually discarded as it immediately leads to a nonconvex region; linearization about a point in this region leads to linear constraints that exclude part of the region that may contain the solution. Assuming that \( g(\mathbf{w}) \) is linear, relaxation of the Big-M formulation leads to a convex region that is, however, arbitrarily
large based on the choice of the parameter \( M \). Finally, relaxation of the GDP formulation leads to a tight, convex region, but at the cost of increasing the problem size with the disaggregated variables.

To illustrate the effect of these formulations, consider the disjunction:

\[ w \in [20, 40] \lor w \in [100, 200]. \]

For both the nonconvex and the GDP formulations the relaxed feasible region can be shown to be:

\[
\begin{align*}
100 - 80y & \leq w \leq 200 - 160y \\
0 & \leq y \leq 1
\end{align*}
\]

while the Big-M formulation yields:

\[
\begin{align*}
My + 20 - M & \leq w \leq 40 + M - My \\
100 - My & \leq w \leq 200 + My \\
0 & \leq y \leq 1
\end{align*}
\]

From these constraints, it is clear that for \( M > 160 \) the feasible region for Big-M is larger than the region for the other two formulations. Note that the region for the nonconvex formulation is convex here, but if we consider the equivalent disjunction:

\[ 5w \in [100, 200] \lor w \in [100, 200], \]

we obtain the alternate nonconvex region:

\[
\begin{align*}
100 & \leq w + 4wy \leq 200 \\
0 & \leq y \leq 1.
\end{align*}
\]

**Fig. 1** presents the feasible regions for these formulations.

Substitution of equations (7)–(10) into (1)–(5) along with one of the representations of the disjunctions (6) leads to the following MINLP:

\[
\begin{align*}
\min_{x' \in \mathbb{R}^n} & \quad f(x', y) \\
\text{s.t.} & \quad c_i \leq c(x', y) \leq c_{i\alpha} \\
& \quad x'_L \leq x' \leq x'_U, \; y \in \{0, 1\}^{ny}
\end{align*}
\]

where \( x' = \left( \frac{dx}{dq}, z_i, y_{i,q}, u_{i,q}, t, p \right)^T, \quad f : \mathbb{R}^n \rightarrow \mathbb{R} \) and \( c : \mathbb{R}^n \rightarrow \mathbb{R}^m \). We next provide a brief description of algorithms for the solution of this MINLP.

### 2.4. MINLP algorithms

For the solution of (15)–(17), we consider SBB, an NLP branch and bound algorithm and DICOPT, based on outer approximation (Viswanathan & Grossmann, 1990). Both are implemented in the GAMS modeling environment (Brooke, Kendrick, & Meeraus, 1992); CONOPT is used as the NLP solver and XPRESS is the MILP solver.

The SBB algorithm is based on standard branch and bound methods combined with selected NLP solvers incorporated within GAMS. During the solution process SBB navigates a binary tree and at each node solves a number NLP models, with a subset of binary decision variables fixed. Solutions to these NLP models at the leaf nodes (i.e., with all binary decisions fixed) provide upper bounds to the objective function. Otherwise, solutions at intermediate nodes with a subset of binary decisions relaxed leads either to valid lower bounds or to objectives that exceed current upper bounds; the latter can be used to fathom branches of the tree.

Discrete and continuous optimizer (DICOPT) applies the outer approximation algorithm with equality relaxation. The MINLP is first solved as a relaxed NLP to initialize the variables and fix a likely set of binary decisions. For this fixed set of decisions, an NLP problem is solved to yield an upper bound on the MINLP solution. Next, the constraints and objective functions are linearized about the solution and this information is used to update a master MILP problem. Solution of the MILP problem yields new values of the binary decisions and a lower bound to the objective function. A new NLP is solved with the binary decisions and a new upper bound is obtained. This process repeats until the upper and lower bounds cross.

Both methods have been proved to converge to the global MINLP solution as long as the MINLP problem is convex. In the absence of convexity, these algorithms can, at best, converge to local solutions; this is characteristic of the MIDO problems in this study. Moreover, simultaneous MINLP formulations of the MIDO problem tend to have many continuous variables and constraints and relatively few binary decisions. These lead to the following algorithmic considerations:

- MINLPs derived from MIDO problems tend to have expensive NLP solutions. This makes algorithms favorable that require fewer NLP solutions. As a result, the SBB algorithm is at a disadvantage here.
- Lower bounds determined in all three algorithms rely on tight relaxations of the binary variables. This has a direct effect on fast convergence of the combinatorial part of the MINLP algorithm. As a result, Big-M formulations are likely to perform poorly.
- For nonconvex constraint and objective functions, linearizations can lead to infeasible MILP problems in DICOPT; recovery from these failures is needed.
- NLP subproblems need to be handled reliably for the full range of binary decisions. If failure of the NLP solver occurs even when feasible solutions exist, performance of both MINLP algorithms can be severely impeded. As a result, robust NLP initialization strategies are needed.
In the next section, we see how these characteristics influence the solution of a MIDO case study problem.

3. Case study

MINLP methods for the simultaneous design and control problem first require development of a superstructure of processing and control alternatives, in which the optimal solution is embedded. Special representation schemes have been proposed for addressing the efficient generation of processing alternatives (Biegler et al., 1997). Here, we choose process alternatives that relate to the structure of the flowsheet. For control alternatives we choose among output (i.e., controlled) variables and input (i.e., manipulated) variables, and then determine the control laws between them.

3.1. Problem formulation

The simultaneous MIDO approach is applied to a system of two series-connected CSTRs systems. Design parameters values and notation are shown in Table 1 for the following DAE model.

- First reactor

\[
\frac{dC_1}{dt} = \frac{C_f - C_1}{\theta} + r_{A1} \tag{18}
\]

\[
\frac{dT_1}{dt} = \frac{T_f - T_1}{\theta} + \beta r_{A1} - \alpha(T_1 - T_{c1}) \tag{19}
\]

- Second reactor

\[
\frac{dC_2}{dt} = \frac{C_1 - C_2}{\theta} + r_{A2} \tag{20}
\]

\[
\frac{dT_2}{dt} = \frac{T_1 - T_2}{\theta} + \beta r_{A2} - \alpha(T_2 - T_{c2}) \tag{21}
\]

The form of the equation describing the jacket energy balance depends on whether the cooling water flows either in co-current or counter-current direction. For the jacket equations:

\[
\frac{dT_{c1}}{dt} = \frac{Q_c(T_{c1}^\text{in} - T_{c1})}{V_c} + \alpha_c(T_1 - T_{c1}) \tag{22}
\]

\[
\frac{dT_{c2}}{dt} = \frac{Q_c(T_{c2}^\text{in} - T_{c2})}{V_c} + \alpha_c(T_2 - T_{c2}) \tag{23}
\]

the co-current direction has \(T_{c1}^\text{in} = T_f\) and \(T_{c2}^\text{in} = T_{c1}\), while the counter-current flow direction has \(T_{c1}^\text{in} = T_{c2}\) and \(T_{c2}^\text{in} = T_{c1}\). Here, we also define:

\[
\theta = \frac{V}{Q_c}, \quad \beta = \frac{\Delta H_r}{\rho C_p}, \quad \alpha = \frac{UA}{\rho C_p \cdot \rho C_p}, \quad \alpha_c = \frac{UA}{\rho C_p \cdot \rho C_p},
\]

\[
r_{A1} = -K_0 e^{-E/RT_1} C_1, \quad r_{A2} = -K_0 e^{-E/RT_2} C_2.
\]

Under these operating conditions, the system of two series-connected reaction system exhibits up to five steady-state solutions at the outlet of the second reactor. Fig. 3 depicts the continuation diagrams for both co-current and counter-current cooling systems using the feedstream temperature as the main continuation parameter.

3.2. MIDO formulation

The objective of the simultaneous design and control problem is to design the series-connected CSTR system, and its associated control system, so that the conversion at the second reactor outlet is maximized. Since, normally conversion is not measured on-line, maximum conversion will be achieved through the closed-loop control of the second reactor temperature \(T_2\).

The cooling system of the reaction train may be operated either in a co-current or counter-current direction. Candidates for manipulated variables are the cooling water flow rate \(Q_c\) and the feedstream temperature \(T_f\). Either \(T_1\) or \(T_2\) can be chosen as controlled variables. For simplicity, we choose three binary decision variables to represent selections. Table 2 contains the binary variables \(y_i, i = 1, \ldots, 3\) defining the integer decisions, while the superstructure of processing alternatives is shown in Fig. 2.

As seen below, we assume that the controllers are just PI controllers. Therefore, the aims of the MIDO problem solution are: (a) to find the best cooling water configuration, (b) to compute reactor and cooling water residence times, (c) to select the control structure and (d) to select the PI controller tuning parameters so that the disturbance is rejected in minimum time, while keeping the second reactor temperature at its set-point. We now consider the three MINLP formulation for this problem.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Q)</td>
<td>2.5</td>
<td>l/s</td>
</tr>
<tr>
<td>(T_f)</td>
<td>29</td>
<td>°C</td>
</tr>
<tr>
<td>(C_f)</td>
<td>0.6</td>
<td>mol/l</td>
</tr>
<tr>
<td>(V)</td>
<td>900</td>
<td>l</td>
</tr>
<tr>
<td>(Q_c)</td>
<td>2</td>
<td>l/s</td>
</tr>
<tr>
<td>(T_{c1})</td>
<td>25</td>
<td>°C</td>
</tr>
<tr>
<td>(V_{c})</td>
<td>100</td>
<td>l</td>
</tr>
<tr>
<td>(E)</td>
<td>10.1</td>
<td>kcal/mol</td>
</tr>
<tr>
<td>(K_o)</td>
<td>2 \times 10^4</td>
<td>l/s</td>
</tr>
<tr>
<td>(R)</td>
<td>1.98 \times 10^{-3}</td>
<td>kcal/mol</td>
</tr>
<tr>
<td>(\rho)</td>
<td>850</td>
<td>g/l</td>
</tr>
<tr>
<td>(C_p)</td>
<td>1.35 \times 10^{-4}</td>
<td>kcal/g °C</td>
</tr>
<tr>
<td>(\Delta H_r)</td>
<td>-35</td>
<td>kcal/mol</td>
</tr>
<tr>
<td>(\rho_c)</td>
<td>1000</td>
<td>g/l</td>
</tr>
<tr>
<td>(C_{p,c})</td>
<td>1 \times 10^{-3}</td>
<td>kcal/g °C</td>
</tr>
<tr>
<td>(A)</td>
<td>900</td>
<td>cm²</td>
</tr>
<tr>
<td>(U)</td>
<td>4 \times 10^{-5}</td>
<td>kcal/cm² °C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Definition of the binary variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>(y_i)</td>
</tr>
<tr>
<td>(y_i)</td>
</tr>
<tr>
<td>(y_i)</td>
</tr>
<tr>
<td>(y_i)</td>
</tr>
<tr>
<td>(y_i)</td>
</tr>
<tr>
<td>(y_i)</td>
</tr>
</tbody>
</table>
Fig. 2. Superstructure of processing alternatives.

Fig. 3. Continuation diagrams at the outlet of the first and second reactor using the feedstream temperature as the main continuation parameter. Dashed arcs show unstable solutions. (a and b) Co-current and (c and d) counter-current.
**Nonconvex formulation:** Here, we introduce binary variables as nonconvex switching variables for the cooling and control selection.

\[
\min_{x,u,y,t_f} \frac{1}{T_f} \int_0^{T_f} (T_2^{sp} - T_2)^2 \, dt
\]

CSTR equations: (18)–(23)

**Disturbances**

\[ C_f = C_f^{\text{nominal}} + \alpha_d (e^{-\lambda t} - 1) \]  

Binary variables

\[ y_c \in \{0, 1\}, \quad y_i \in \{0, 1\}, \quad y_o \in \{0, 1\} \]

**Cooling direction**

\[ T_{c1}^{\text{in}} = T_f y_c + (1 - y_c) T_{c2} \]
\[ T_{c2}^{\text{in}} = T_{c1} y_c + (1 - y_c) T_{cf} \]

**Control system**

\[
T_f = T_f^{\text{bias}} + y_i (K_P \times P(t) + K_I I(t))
\]
\[
Q_c = Q_c^{\text{bias}} - (K_P \times P(t) + K_I I(t))
\]

\[\begin{align*}
0 &\leq K_P^T \leq M y_i \\
0 &\leq K_P^Q \leq M y_i \\
0 &\leq K_I^T \leq M y_i \\
0 &\leq K_I^Q \leq M y_i
\end{align*}\]

Control system—inputs

\[
T_f = T_f^{\text{bias}} + (K_P^T \times P(t) + K_I^T I(t))
\]
\[
Q_c = Q_c^{\text{bias}} - (K_P^Q \times P(t) + K_I^Q I(t))
\]

\[\begin{align*}
0 &\leq K_P^T \leq M y_i \\
0 &\leq K_P^Q \leq M y_i \\
0 &\leq K_I^T \leq M y_i \\
0 &\leq K_I^Q \leq M y_i
\end{align*}\]

Control system—outputs

\[
\begin{align*}
(T_1^{sp} - T_1) - M (1 - y_o) &\leq T_1^{\text{diff}} \leq (T_1^{sp} - T_1) + M (1 - y_o) \\
-M y_o &\leq T_1^{\text{diff}} \leq M y_o \\
(T_2^{sp} - T_2) - M y_o &\leq T_2^{\text{diff}} \leq (T_2^{sp} - T_2) + M y_o \\
-M (1 - y_o) &\leq T_2^{\text{diff}} \leq M (1 - y_o) \\
P(t) &\equiv T_1^{\text{diff}} + T_2^{\text{diff}} \\
\frac{dI}{dt} &= P(t), \quad I(0) = 0
\end{align*}\]

**GDP formulation:** Additional continuous variables are introduced as disagggregations of the terms in the disjunctions. Note that all of the added constraints are linear. Finally, lower and upper bounds are added for the disagggregated variables. Since these are not known a priori, we set them to \(a_j = -M\) and \(b_j = M\).

\[
\min_{x,u,y,t_f} \frac{1}{T_f} \int_0^{T_f} (T_2^{sp} - T_2)^2 \, dt
\]

Equations: (18)–(23), (25), (26)

**Cooling direction**

\[
T_{c1}^{\text{in}} = t_{c11} + t_{c12}
\]
\[
\begin{align*}
a_{11} y_c &\leq t_{c1} \leq b_{11} y_c \\
a_{12} (1 - y_c) &\leq t_{c1} \leq b_{12} (1 - y_c)
\end{align*}
\]
\[
T_{c2}^{\text{in}} = t_{c21} + t_{c22}
\]
\[
\begin{align*}
a_{21} y_c &\leq t_{c2} \leq b_{21} y_c \\
a_{22} (1 - y_c) &\leq t_{c2} \leq b_{22} (1 - y_c)
\end{align*}
\]
\[
T_{cf} = t_{c11} + t_{c22}
\]
\[
T_{c1} = t_{c11} + s_1
\]
\[
T_{c2} = t_{c12} + s_2
\]
\[
\begin{align*}
a_{31} (1 - y_c) &\leq s_1 \leq b_{31} (1 - y_c) \\
a_{32} y_c &\leq s_2 \leq b_{32} y_c
\end{align*}
\]

**Control system—inputs**

\[
\begin{align*}
T_f &= T_f^{\text{bias}} + (K_P^T \times P(t) + K_I^T I(t)) \\
Q_c &= Q_c^{\text{bias}} - (K_P^Q \times P(t) + K_I^Q I(t)) \\
0 &\leq K_P^T \leq M y_i \\
0 &\leq K_P^Q \leq M y_i \\
0 &\leq K_I^T \leq M y_i \\
0 &\leq K_I^Q \leq M y_i
\end{align*}
\]

**Jacket equations**

\[
\frac{dT_{c1}}{dt} = \frac{Q_c (\Delta T_{1,co} + \Delta T_{1,\text{count}})}{V_c} + \alpha_c (T_1 - T_{c1})
\]
\[
\frac{dT_{c2}}{dt} = \frac{Q_c (\Delta T_{2,co} + \Delta T_{2,\text{count}})}{V_c} + \alpha_c (T_2 - T_{c2})
\]

**Cooling direction**

\[
\begin{align*}
-M (1 - y_c) + T_f - T_{c1} &\leq \Delta T_{1,co} \leq T_f - T_{c1} + M (1 - y_c) \\
-M y_c + T_{c2} - T_{c1} &\leq \Delta T_{1,\text{count}} \leq T_{c2} - T_{c1} + M y_c \\
-M (1 - y_c) + T_{cf} - T_{c2} &\leq \Delta T_{2,co} \leq T_{cf} - T_{c2} + M (1 - y_c) \\
-M y_c &\leq \Delta T_{1,co} \leq M y_c \\
-M (1 - y_c) &\leq \Delta T_{1,\text{count}} \leq M (1 - y_c) \\
-M y_c &\leq \Delta T_{2,co} \leq M y_c \\
-M (1 - y_c) &\leq \Delta T_{2,\text{count}} \leq M (1 - y_c)
\end{align*}
\]

Equations: (18)–(21), (25), (26)
Control system—outputs

\[
\begin{align*}
(T_1^\text{sp} - T_1) &= r_1 + q_1 \\
(T_1^\text{sp} - T_2) &= r_2 + q_2 \\
a_1 y_o &\leq r_1 \leq b_1 \\
a_2 (1 - y_o) &\leq r_2 \leq b_2 (1 - y_o) \\
a_0 q_1 (1 - y_o) &\leq q_1 \leq b_0 q_1 (1 - y_o) \\
a_0 q_2 y_o &\leq q_2 \leq b_0 q_2 y_o \\
P(t) &= r_1 + r_2 \\
\frac{dI}{dt} &= P(t), \quad I(0) = 0
\end{align*}
\]

4. Results and discussion

Several cases are analyzed to evaluate the above MINLP approaches. In all the scenarios, it was assumed that the coupled reaction system always operates around the region of high concentration and temperature shown in Fig. 3. Also, note the high sensitivities, multiplicity and instability in the phase diagrams. As seen in (25), feedstream composition disturbances were injected into the system with an amplitude of \( \alpha \) grams. As seen in (25), feedstream composition disturbances were injected into the system with an amplitude of \( \alpha \) grams. As seen in (25), feedstream composition disturbance increases as well, as the heat released by the reaction diminishes. Therefore, the effect of the closed-loop control action will be to increase the second reactor temperature until it reaches its set-point temperature. Here, we considered all three MINLP formulations and both MINLP algorithms. For this problem, the nonconvex formulation has 1925 continuous variables, 3 binary variables and 1921 constraint equations. On the other hand, the Big-M formulation has 2285 continuous variables, 3 binary variables and 3905 constraints, most of which are inequalities, while the GDP formulation has 2647 continuous variables, 3 binary variables and 3730 constraints. The last two formulations are considerably larger than the nonconvex formulation. Moreover, only the nonconvex formulation could be solved reliably as stated; the other two formulations often terminated with infeasible NLP solutions. To avoid infeasibilities in the Big-M and GDP formulations, we added artificial variables to the right-hand side of each nonlinear equation, \( j \), i.e., \( a_j^+ - a_j^- \), with \( a_j^+ \), \( a_j^- \) \( \geq 0 \); we also added the penalty term, \( M_P \Sigma_j (a_j^+ + a_j^-) \), to the objective function (with \( M_P = 10^4 \)).

This successfully overcame the infeasibility problems, but an additional 1440 variables were added in each formulation. Results of the MINLP formulations are given in Table 3. Here, we see that DICOPT is able to obtain solutions for all three formulations and the nonconvex formulation requires the least amount of effort. The SBB algorithm fails on the Big-M and GDP formulations, but does quite well on the nonconvex formulation. Also, because only three binary variables are present, we present an enumeration of all eight combinations in the MINLP in Table 4; this enumeration also allowed us to verify the consistency of the three MINLP formulations. Table 4 shows that the best control loop pairing is \( T_f \rightarrow T_2 \) and that the difference between the co-current and counter-current cooling direction is negligible. Moreover, from Table 3, the best computational performance is obtained with the nonconvex formulation and the SBB algorithm.

### Table 3

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Algorithm</th>
<th>CPU (s)</th>
<th>Objective function</th>
<th>( y_i/y_o )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonconvex</td>
<td>DICOPT</td>
<td>101.76</td>
<td>0.001</td>
<td>0/1/0</td>
</tr>
<tr>
<td>Nonconvex</td>
<td>SBB</td>
<td>23.45</td>
<td>0.001</td>
<td>1/1/0</td>
</tr>
<tr>
<td>Big-M</td>
<td>DICOPT</td>
<td>198.96</td>
<td>0.0</td>
<td>1/1/0</td>
</tr>
<tr>
<td>Big-M</td>
<td>SBB</td>
<td>122.44</td>
<td>0.0</td>
<td>0.153/0.21</td>
</tr>
<tr>
<td>GDP</td>
<td>DICOPT</td>
<td>117.53</td>
<td>0.0577</td>
<td>0/1/0</td>
</tr>
<tr>
<td>GDP</td>
<td>SBB</td>
<td>211.86</td>
<td>0.0898</td>
<td>0/0/1</td>
</tr>
</tbody>
</table>

* a Failed to solve.
* b Small penalty/constraint violation.

### Table 4

<table>
<thead>
<tr>
<th>Direction</th>
<th>Pairing</th>
<th>( y_i/y_o )</th>
<th>Objective function</th>
<th>Solution profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-current</td>
<td>( T_f \rightarrow T_1 )</td>
<td>1/1/1</td>
<td>0.988</td>
<td>Exponentially damped</td>
</tr>
<tr>
<td>Counter-current</td>
<td>( T_f \rightarrow T_1 )</td>
<td>0/1/1</td>
<td>1.456</td>
<td>Exponentially damped</td>
</tr>
<tr>
<td>Co-current</td>
<td>( Q_i \rightarrow T_1 )</td>
<td>1/0/1</td>
<td>40.123</td>
<td>Exponentially damped</td>
</tr>
<tr>
<td>Counter-current</td>
<td>( Q_i \rightarrow T_1 )</td>
<td>0/0/1</td>
<td>40.447</td>
<td>Exponentially damped</td>
</tr>
<tr>
<td>Co-current</td>
<td>( T_f \rightarrow T_2 )</td>
<td>1/1/0</td>
<td>0.0010</td>
<td>Oscillatory increasing</td>
</tr>
<tr>
<td>Counter-current</td>
<td>( T_f \rightarrow T_2 )</td>
<td>0/1/0</td>
<td>0.0011</td>
<td>Oscillatory increasing</td>
</tr>
<tr>
<td>Co-current</td>
<td>( Q_i \rightarrow T_2 )</td>
<td>1/0/0</td>
<td>43.53</td>
<td>Exponentially damped</td>
</tr>
<tr>
<td>Counter-current</td>
<td>( Q_i \rightarrow T_2 )</td>
<td>0/0/1</td>
<td>44.61</td>
<td>Exponentially damped</td>
</tr>
</tbody>
</table>

### Table 4

<table>
<thead>
<tr>
<th>Direction</th>
<th>Pairing</th>
<th>( y_i/y_o )</th>
<th>Objective function</th>
<th>Solution profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Co-current</td>
<td>( T_f \rightarrow T_1 )</td>
<td>1/1/1</td>
<td>0.988</td>
<td>Exponentially damped</td>
</tr>
<tr>
<td>Counter-current</td>
<td>( T_f \rightarrow T_1 )</td>
<td>0/1/1</td>
<td>1.456</td>
<td>Exponentially damped</td>
</tr>
<tr>
<td>Co-current</td>
<td>( Q_i \rightarrow T_1 )</td>
<td>1/0/1</td>
<td>40.123</td>
<td>Exponentially damped</td>
</tr>
<tr>
<td>Counter-current</td>
<td>( Q_i \rightarrow T_1 )</td>
<td>0/0/1</td>
<td>40.447</td>
<td>Exponentially damped</td>
</tr>
<tr>
<td>Co-current</td>
<td>( T_f \rightarrow T_2 )</td>
<td>1/1/0</td>
<td>0.0010</td>
<td>Oscillatory increasing</td>
</tr>
<tr>
<td>Counter-current</td>
<td>( T_f \rightarrow T_2 )</td>
<td>0/1/0</td>
<td>0.0011</td>
<td>Oscillatory increasing</td>
</tr>
<tr>
<td>Co-current</td>
<td>( Q_i \rightarrow T_2 )</td>
<td>1/0/0</td>
<td>43.53</td>
<td>Exponentially damped</td>
</tr>
<tr>
<td>Counter-current</td>
<td>( Q_i \rightarrow T_2 )</td>
<td>0/0/1</td>
<td>44.61</td>
<td>Exponentially damped</td>
</tr>
</tbody>
</table>
Table 5
Nonconvex MINLP formulation results for all cases

<table>
<thead>
<tr>
<th>Case</th>
<th>Method</th>
<th>$c_{f, \text{final}}$</th>
<th>$\theta_d$</th>
<th>CPU (s)</th>
<th>Objective function</th>
<th>$y_i/y_i^o$</th>
<th>$K_p$</th>
<th>$K_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DICOPT</td>
<td>0.55</td>
<td>0.1</td>
<td>101.76</td>
<td>0.001</td>
<td>0/1/0</td>
<td>500</td>
<td>0.357</td>
</tr>
<tr>
<td>1</td>
<td>SBB</td>
<td>0.55</td>
<td>0.1</td>
<td>23.45</td>
<td>0.001</td>
<td>1/1/0</td>
<td>500</td>
<td>0.368</td>
</tr>
<tr>
<td>2</td>
<td>DICOPT</td>
<td>0.55</td>
<td>6.0</td>
<td>31.04</td>
<td>0.0009</td>
<td>0/1/0</td>
<td>500</td>
<td>0.445</td>
</tr>
<tr>
<td>2</td>
<td>SBB</td>
<td>0.55</td>
<td>6.0</td>
<td>21.25</td>
<td>0.0009</td>
<td>0/1/0</td>
<td>500</td>
<td>0.445</td>
</tr>
<tr>
<td>3</td>
<td>DICOPT$^a$</td>
<td>0.65</td>
<td>0.1</td>
<td>32.65</td>
<td>0.955</td>
<td>1/1/1</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>3</td>
<td>SBB</td>
<td>0.65</td>
<td>0.1</td>
<td>49.11</td>
<td>0.583</td>
<td>0/1/1</td>
<td>0</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>DICOPT$^a$</td>
<td>0.65</td>
<td>6.0</td>
<td>21.0</td>
<td>0.0</td>
<td>0.531/0.459</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>4</td>
<td>SBB</td>
<td>0.65</td>
<td>6.0</td>
<td>34.61</td>
<td>0.575</td>
<td>0/1/1</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>5</td>
<td>DICOPT$^a$</td>
<td>0.55</td>
<td>0.1</td>
<td>17.38</td>
<td>0.0098</td>
<td>0/0/0</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>5</td>
<td>SBB</td>
<td>0.55</td>
<td>0.1</td>
<td>18.80</td>
<td>0.0009</td>
<td>1/1/1</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>6</td>
<td>DICOPT$^a$</td>
<td>0.65</td>
<td>0.1</td>
<td>41.22</td>
<td>14.87</td>
<td>0/0/0</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>6</td>
<td>SBB</td>
<td>0.65</td>
<td>0.1</td>
<td>69.33</td>
<td>0.0025</td>
<td>0/1/1</td>
<td>356.0</td>
<td>500</td>
</tr>
</tbody>
</table>

$^a$ Failed to solve.

The MINLP solution in Table 3 also dictates that $T_2$ must be controlled by the first reactor feedstream temperature $T_f$, and that both counter-current and co-current solutions are detected. Table 5 shows the PI controller settings; proportional gain is at its upper bound with a small integral component added. The reason why the $T_f \rightarrow T_2$ control pairing was selected is because the $T_2(s)/T_f(s)$ gain is higher than the $T_2(s)/Q_c(s)$ gain.

![Fig. 4. Case 1 profiles: co-current solution.](image1)

![Fig. 5. Case 1 profiles: counter-current solution.](image2)

Fig. 4 shows the state and control profiles for the co-current case while Fig. 5 shows the counter-current case. In both cases $T_f$ and $T_1$ have oscillating profiles with increasing amplitude, characteristic of an unstable solution. Note that the simultaneous approach is able to capture these unstable solutions. Despite the instability, this solution rejects disturbances very well. The integral squared error is about 1.0 and $T_2$ is effectively maintained at the set-point.
As formulated, the MIDO approach does not avoid unstable solutions but allows them to be determined in order to detect best performance limits. Of course, stability constraints could be imposed in the MIDO problem (as in Blanco, Bandoni, & Biegler, 2004; Mohideen et al., 1996) to avoid unstable implementations, but these constraints generally lead to reduced performance.

While this unstable solution is not implementable, it demonstrates the limits of the control/design problem and suggests three alternatives for improvement: detune the controller, redesign the process, adopt a multivariable controller. A rough idea of the consequences of detuning can be seen in Table 4, where all the other scenarios are stable, but have integral squared errors that are higher by three or four orders of magnitude. On the other hand, process redesign is considered in Case 5, where a stable system results with excellent disturbance rejection. Finally, advanced controllers can also be included within simultaneous MIDO formulations; we leave this as a topic for future work.

4.2. Case 2: decreasing first-order disturbance

This case resembles Case 1, but instead of a step disturbance, we assume that the feedstream composition disturbance has a first-order transfer function shape with a 6.0 s time constant. Based on the experience from Case 1, we obtain solutions only for the nonconvex formulation, using DICOPT and SBB. Both algorithms obtain the same solution with excellent disturbance rejection (integral squared error less than 1.0, but with unstable characteristics). As noted from Table 5, the solution of the MIDO problem again selects the $T_f \rightarrow T_2$ pairing and counter-current cooling with similar PI settings as in Case 1. Although the feedstream disturbance is slower, the observed response is very similar to Case 1. From Fig. 6, we note that $T_2$ is maintained very well at its set-point (within 0.05 °C).

As in Case 1, this unstable solution is not implementable but it demonstrates the limits of the control/design problem and allows for areas of improvement. In particular, we can consider a redesign, similar to Case 5, where a stable system leads to excellent disturbance rejection.

4.3. Case 3: increasing step disturbance

In this case, the feedstream composition disturbances increases from 0.6 to 0.65 as a step input. Again we consider only the nonconvex formulation. Here, only SBB is able to obtain a reasonable solution with an integral squared error of 583,
although we believe this may be a local result. When the disturbance hits the system, the second reactor temperature \( T_2 \) starts rising. Here, the disturbance is not rejected as well as in the first two cases; there is an offset of about 1 °C in \( T_2 \). However, as seen from Fig. 7, the profiles are stable without oscillations. The solution of the MIDO problem selects the \( T_f \rightarrow T_1 \) controller pairing, along with counter-current cooling. Fig. 7 displays the profiles for the state and control variables. We believe that poorer disturbance rejection occurs because the reaction system exhibits stronger nonlinearities in the second reactor when the feedstream composition is increased (see Fig. 3).

4.4. Case 4: increasing first-order disturbance

This case is similar to Case 3, except that the feedstream composition disturbance has a first-order transfer function shape with a 6.0 s time constant. The solution of the MIDO problem formulation again selects the \( T_f \rightarrow T_1 \) control pairing and counter-current cooling. Again we consider only the nonconvex formulation. Moreover, only SBB is able to obtain a reasonable solution, although we believe this may be only a local result, with an integral squared error of 575. Here, the disturbance is not rejected as well as in the first two cases and there is an offset in \( T_2 \) of about 1 °C. Nevertheless, profiles for the state and manipulated variables are stable. Constants for the proportional and integral elements are set to their upper bounds. Fig. 8 displays the profiles for the state and control variables; these are very similar to the ones in Case 3. As in Case 3, we believe that the worsened disturbance rejection occurs because the reaction system exhibits stronger nonlinearities when the feedstream composition is increased.

4.5. Case 5: redesign with decreasing step disturbance

This case has the same disturbance as in Case 1. Here, we extend the MIDO problem by adding two design variables, the reactor residence time and the volume of the cooling jacket; these are allowed to vary between 50% and 200% of their nominal values (see Table 1). Again, we apply only the nonconvex formulation and initialize the problem from the solution of Case 1. As seen in Table 5, only SBB leads to a satisfactory solution. From Fig. 9, this case reveals a very interesting result. The integral squared error is increased from Case 1, but still has very good disturbance rejection; \( T_2 \) is maintained very well at its set-point (within 0.02 °C). Moreover, with the adjustment of the equipment parameters, the profiles are now stable and the oscillations from Case 1 have disappeared. Here, the reactor
residence time is reduced by 11% while the jacket volume is reduced by 50% (its lower bound). Counter-current cooling is chosen along with proportional and integral gains at their upper bounds. Also, the controller pairing is $T_f \rightarrow T_1$. Note that this differs from the pairing in Case 1, but leads to a lower integral squared error. This case shows the benefits that can be obtained with integrated design and control, as better controller performance is obtained with equipment redesign. Moreover, for this problem we obtain the interesting result that the performance is now stabilized.

### 4.6. Case 6: redesign with increasing step disturbance

This case has the same disturbance as in Case 3. Again, we extend the MIDO formulation by adding two design variables. The reactor residence time and the volume of the cooling jacket are allowed to vary between 50% and 200% of their nominal values (see Table 1). Applying the nonconvex formulation, we initialize the problem from the solution of Case 3. As seen in Table 5 only SBB leads to a satisfactory solution. Here, the reactor residence time is increased by 8.5% while the jacket volume is reduced by 50% (its lower bound). Counter-current cooling is chosen along with proportional and integral gains nearly at their upper bounds. Also, the controller pairing is $T_f \rightarrow T_1$. This is the same structure as in Case 3.

Note that the adjustment of the equipment parameters leads to a much better solution than in Case 3, with an integral squared error that is decreased more than 200 times. Also, as in Case 3, the state and manipulated variable profiles are stable without oscillations. This solution allows much better rejection of the disturbance than in Case 3, as seen in Fig. 10. Here, $T_2$ is maintained at its set-point within about 0.1 °C. As with Case 5, this case shows the benefits of integrated design and control. Here, disturbance rejection is greatly improved through the redesign of the process equipment.

### 5. Conclusions

Dynamic optimization with unstable models can only be tackled reliably with a simultaneous NLP formulation, where both the state and control variables are discretized. Because such instabilities occur in the integrated design and control problem, systematic strategies for this task need to consider these large-scale optimization formulations. Moreover, formulated as a MIDO problem, integrated design and control problems have relatively few binary variables and many continuous variables. In solving the resulting MINLP problem, NLP subproblems require most of the computational effort. This is in contrast to MINLPs formulated for planning and scheduling. In addition, the NLP subproblems are often beyond the limits of emerging global optimization solvers. As a result, attention must be paid to reliable NLP solvers and good initialization strategies.

This study considers the integrated design and control of a two CSTR sequence with a single control loop and two possible cooling directions. Although there are few discrete decisions, this problem is highly nonlinear and has a number of unstable regions. Here, we considered six cases for disturbance rejection. The results of the MIDO formulation showed that excellent performance can be obtained with the selection of control structures and tuning parameters. An interesting observation is that the simultaneous MIDO approach also finds unstable solutions that may represent the best controller performance. While not implementable, such cases represent targets, which can be compared against alternative problem formulations. Moreover, in this study, we show that unstable solutions, obtained with fixed equipment designs, could be stabilized through redesign of the equipment. This is a key motivation for integrated design and control.

To solve the MINLPs in this case study, we compared three MINLP formulations. Both the Big-M and GDP formulations add only linear constraints to define relations about the discrete variables; these are very useful for convex problems. Moreover, the GDP formulation leads to tighter relaxations of the discrete variables and better lower bounds to the MINLP solution, but at the expense of larger problem sizes. We also considered a simple and small nonconvex formulation, which also has a tight relaxation, but linearizations of this formulation may cut off the MINLP solution. These characteristics are observed with two MINLP algorithms: DICOPT and SBB. Because of the difficulties in initializing solving larger NLP problems, neither the GDP nor the Big-M formulations worked well; only the nonconvex formulation gave reliable solutions to all of the cases. Moreover, the SBB method was superior to DICOPT in this study and was the only method that could solve all problems. This can be justi-
fied because SBB solves relaxed NLPs at the nodes of the branch and bound tree, and does not rely on linearizations to develop lower bounds. As a result, it is less likely to suffer convergence failures associated with poor linearizations of nonconvex functions.

With up to 2647 continuous variables, 3905 constraints and a high degree of nonlinearity the case study addressed in this work is much more demanding than many MINLP cases addressed in the literature. As a result the challenges are not the complexity of dealing with many discrete decisions, but rather how aspects of the MINLP algorithm address nonconvexities and large size for the continuous variable problem. Handling these characteristics is independent of the number of binary decisions. Moreover, another advantage of keeping the number of binary variables small is that it allows an enumeration of all possibilities in order to assess the performance of the MINLP algorithms. On the other hand, using the MIDO approach proposed in this paper, a polymer scheduling case study with 96 binary decisions has been recently addressed in Flores-Tlacuahuac and Grossmann (2006). Finally, these results are useful to guide our future work for the formulation and solution of challenging integrated design and control problems. Future MINLP strategies will deal with specialized MINLP formulations for problems with challenging NLP subproblems. More reliable NLP solvers are being developed with additional features that allow specialized initialization strategies. Of great interest are more efficient global solvers that deal systematically with nonconvexity. Finally, we intend to consider integrated design and control problems for larger nonlinear systems with challenging dynamics and embedded stability considerations.

Acknowledgements

Antonio Flores-Tlacuahuac wishes to acknowledge the financial support from the Fulbright Commission, Universidad Iberoamericana-Santa Fe and to the Department of Chemical Engineering at Carnegie-Mellon University.

References