A path constrained method for integration of process design and control

Jigar Patel, Korkut Uygun, Yinlun Huang *

Abstract

Integration of process design and control (IPDC) has been the holy grail of process systems engineering since the introduction of heat and mass integration. A proper combination of these separate yet connected tasks carries the promise of achieving superior designs that cannot be realized with conventional procedures. In this work, a bi-level dynamic optimization approach is introduced for achieving IPDC in its true sense. The principal idea proposed here is to utilize an optimal controller (a modified linear quadratic regulator) to practically evaluate the best achievable control performance for each candidate design during process design. The evaluation of complete, closed-loop system dynamics can then be meshed with a superstructure-based process design algorithm, thus enabling considering both cost and controllability in design of a process. The practicality of the introduced approach enables a solution of this complex dynamic optimization problem within reasonable computational requirements, as demonstrated in an evaporator case study.

Keywords: Integration of process design and control; Bi-level optimization; Linear quadratic regulator

1. Introduction

Process integration has been an ideal method for reducing energy and raw materials consumption of process industries (El-Halwagi, 1997; El-Halwagi & Manousiouthakis, 1989; Ho & Keller, 1987; Linnhoff et al., 1982; Rossiter, Spiggles, & Klee, 1993). As the level of integration in process plants did increase, industrial practice made it clear that process controllability should be considered during process synthesis (Elliott & Luyben, 1995; Fisher, Doherty, & Douglas, 1988; McAvoy, 1987; Morari, 1992; Perkins & Walsh, 1994; Sheffield, 1992). This realization led to the introduction of another type of integration—the integration of process design and control (IPDC).

Process controllability problems of a plant usually have their sources originated in the early stages of process synthesis and design. Naturally, they should be prevented in the same stage. Over the past decades, a primary focus has been on process alternative screening, mostly based on steady-state models; these methods include system flexibility (Floudas & Grossmann, 1986; Swaney & Grossmann, 1985a,b), disturbance propagation and rejection (Huang & Fan, 1992; Yan, Yang, & Huang, 2001; Yang, Gong, & Huang, 1996; Yang, Lou, & Huang, 2005). There are also a number of dynamic indices, such as relative gain array (Bristol, 1966; Shinskey, 1995), the resiliency index (Morari, 1983) and condition number (Grosdidier, Morari, & Holt, 1985; Morari & Zafiropoulo, 1989). While clearly desirable for IPDC purposes, advances in methods for incorporating dynamic models into the design problem have been slow. Luyben and Floudas (1994a,b) and Walsh and Perkins (1996) presented two of the earlier efforts in this direction; operability (Georgakis, Uztkur, Subramanian, & Vinson, 2003; Subramanian & Georgakis, 2005) and the dynamic flexibility index (Dimitriadis & Pistikopoulos, 1995) are also well worth mentioning.

1.1. Current state of IPDC and motivation

As summarized above, for most of the existing algorithms controllability is evaluated utilizing controllability indices. However, the available indices are all indirect indicators of achievable controllability because closed-loop system dynamic performance is measured indirectly. These indices are unable to check if the system violates any operational constraints during the transition process, where the effects of incoming disturbances to the system are most significant. Therefore, what is
necessary is the development of an IPDC methodology that examines system closed-loop dynamic performance via solving system dynamic equations in the process structure determination and design stage. This is a necessity in order to claim complete integration of design and control.

In parallel to the argument above, a recent trend in the field of IPDC is to involve dynamic simulation of the process within the design procedure (Bansal, Perkins, Pistikopoulos, Ross, & Van schijndel, 2000; Kookos & Perkins, 2001; Sakizlis, Perkins, & Pistikopoulos, 2003), in order to ensure that dynamic constraints and closed-loop controllability measures are satisfied. The simulation of system dynamics in the presence of controllers can, therefore, be performed, which achieves a level of integration completely unattainable by usual IPDC techniques. All the system variables (i.e., state and manipulated variables) can be observed at the dynamic level by this approach.

The main obstacle with this approach is the size of control system design problem that has to be augmented to the already challenging process design problem. A major obstacle in IPDC is that without a control system, it is not possible to evaluate system dynamic performance, i.e., evaluate the input–output controllability (i.e., the ability to satisfy all the steady-state and dynamic constraints under incoming disturbances and set-point changes both at static and dynamic levels). However, due to the fact that possible control configurations, including cascade loops, signal transformation techniques and combinations of control devices grow ad infinitum, control system design is more an art than science. The number of options available for selecting control system design increases the problem size drastically and thus will not be manageable in process synthesis stage. A usual practice is to enable a limited number of PI loops with a minimal number of observed manipulated variable pairing options, and use the observed dynamic performance as an underestimate of the achievable control performance (e.g., Kookos & Perkins, 2001).

As a better method for sidestepping the “curse of dimensionality” problem involved in controller synthesis decisions, and yet preserving the quality of control performance estimates, efforts have been made to incorporate centralized, on-line optimization-based controllers in process design (Brennel & Seider, 1992; Loeblein & Perkins, 1999; Swartz, Perkins, & Pistikopoulos, 2000), as well as control structure selection (Zhu & Henson, 2002). One such approach utilizes model predictive control (MPC). The design problem is reduced in size significantly by using MPC as control scheme, as there is no control structure to be designed. However, MPC requires the solution of an optimization problem at frequent control steps, and therefore still displays high computational requirements. This issue has rendered IPDC with MPC practicable for problems of only small sizes (Uygun, 2004).

The latest iteration of modern process design methods seeks to remove this final obstacle by finding a computationally feasible centralized control scheme with provable control performance. The basic idea of these studies is to derive a simple state-feedback controller, which should demonstrate the control performance similar to that of a centralized controller with online-optimization (such as MPC). The works of Johansen, Petersen, and Slupphaug (2002) and Grancharova and Johansen (2002) are notable in terms of developing such explicit model-based controllers. But these approaches are based on trading control performance for computational advantage, which usually leads to significantly sub-optimal control. On the other hand, the use of model-based parametric controllers (Bemporad, Morari, Dua, & Pistikopoulos, 2002; Pistikopoulos, Dua, Bozinis, Bemporad, & Morari, 2002; Sakizlis et al., 2003; Sakizlis, Perkins, & Pistikopoulos, 2005) results in a switching proportional state-feedback control law that is mathematically proven to have optimal control performances. However, the control design (i.e., design of parametric controllers) is still an iterative procedure, which has to be augmented to the design problem (i.e., repeated for each candidate design during the solution of the design problem), which in our opinion may be prohibitive in computational cost in some circumstances. Another issue is the handling of candidate designs (during iterations of solving the design problem) that are infeasible with
respect to the (dynamic) path constraints; the methods developed for implementation of parametric controllers for IPDC do not seem to address such cases. Even if it is addressed through various exception handling procedures (such as selective constraint removal, as is the case with industrial MPC methods), this issue may render the parametric controller design step exceedingly cumbersome for the already computationally expensive IPDC problems.

This work introduces an alternative off-line centralized controller-based IPDC approach, based on optimal control. The proposed method enables quick identification of an optimal state-feedback control law for each candidate design during the process synthesis stage, based on a modified linear quadratic regulator formulation. Further, the control law developed is a proportional-integral state-feedback controller, hence promising flexibility as compared to the parametric controllers. The meshing of the proposed approach with a design algorithm enables practical consideration of both cost and control performance during design process.

2. IPDC with modified linear quadratic regulator (IPDC-mLQR)

In this work, an optimal control approach by employing a modified linear quadratic regulator (mLQR) is investigated. The basic issue here is how to solve a dynamic optimization problem where the closed-loop control performance is evaluated for each design alternative. The approach is presented starting with a formulation of the IPDC problem.

2.1. A general IPDC formulation

IPDC in this work is treated as a dynamic optimization problem where control-related dynamic properties are considered simultaneously with cost-related static properties in order to design a cost effective, highly controllable process. A general formulation for IPDC problem can be presented below.

\[ \min_{\text{w.r.t. } x^n, u^n, x(t), u(t)} I \]

\[ = \Phi(d, x^n, u^n) \]

\[ + \lambda \int_0^{\infty} \left( \dot{x}^T \cdot Q \cdot \dot{x} + \ddot{u}^T \cdot R_d \cdot \ddot{u} + u^T \cdot R_B \cdot u \right) dt \]

s.t.

\[ x = f(x(t), u(t), d, \Theta(t), t) \]  

\[ f^n(d, x^n, u^n) = 0 \]  

\[ x_{\text{min}}^s \leq x^n \leq x_{\text{max}}^s \]  

\[ u_{\text{min}}^s \leq u^n \leq u_{\text{max}}^s \]  

\[ h^n(d, x^n, u^n, \Theta^n) = 0 \]  

\[ g^n(d, x^n, u^n, \Theta^n) \leq 0 \]  

\[ x_{\text{min}}^d \leq x(t) \leq x_{\text{max}}^d \]  

where

\[ x(t) = x(t) - x^d |_{n=0} \]  

\[ u(t) = u(t) - u^d |_{n=0} \]  

\[ x|_{t=t_0} = x^d |_{n=0} \]  

\[ u|_{t=t_0} = u^d |_{n=0} \]  

\[ \Theta(t) = \Theta^d(t) \]

where \( x \) is the vector of state variables; \( u \) the vector of control variables, \( \Theta \) the vector of disturbances, and \( d \) is the vector of design variables. Superscripts \( d \) and \( s \) denote dynamic and static of relevant variables, respectively. The symbols used in this work are outlined in ‘Nomenclature’. In the objective function, the first part, \( \Phi \), represents the design cost (e.g., total annualized cost (TAC)), and the second part represents the equivalent control cost reflected by dynamic performance criterion; an integral square error form has been used for compatibility with the linear quadratic regulator (LQR) formulation, as it will be discussed later in the text. It has been noted that the integral square error form also has an indirect correlation to the cost (Chintapalli & Douglas, 1975; Shunta, 1995), although the \( \ddot{u} \) term, which leads to a smoother integral control action, does not have an economic significance.

The system dynamics is described by a set of differential equations given in Eq. (2). It should be underlined that the formulation above imposes not only dynamic constraints but also steady-state feasibility constraints in a multi-period formulation given by Eqs. (3)–(7). Superscript \( n \) is used on steady-state control and state variables. The system equations, \( f^n \), and the steady-state constraints, \( h^n \) and \( g^n \), correspond to the particular operation period (note that all variables with superscript \( n \) are static). Period marked by index \( n = 0 \) identifies nominal conditions. For instance, Eq. (3) with \( n = 0 \) requires the system to be at steady-state when there is no disturbance applied to the system. The other operation periods correspond to the vertices of the disturbance space, defined by \( \Theta^n \) as it will be explained below. Note that the problem considers designing a process that can handle a finite number of disturbances (i.e., a design that can cope with the worst case scenarios), rather than a dynamic flexibility index type of study (Dimitriadis & Pistikopoulos, 1995). A typical \( \Theta^n \) vector, therefore, would be a combination of positive and negative deviations in each disturbance variable, hence leading to a total number of \( 2^{\text{dim}(\Theta)} \) equal to the number of corners of the hyper-rectangular disturbance space. In general, the dynamic disturbance profile, \( \Theta^d \), is formed by combinations of step disturbances with the values in \( \Theta^n \). It is important to ensure that each combination of disturbance lasts long enough for the system to settle to the new post-disturbance steady-state in order to fairly measure the effects of each disturbance on the system.
Eqs. (4) and (5) represent the steady-state bounds on system and control variables, respectively. Eqs. (8) and (9) are, respectively, the dynamic bounds on system and control variables. Eqs. (6), (7), (10) and (11) signify other possible static and dynamic (path) constraints.

2.2. The IPDC-mLQR methodology—a bi-level reformulation

There are two alternative solution methods for a dynamic optimization problem, simultaneous and sequential (usually, the former is used in an infeasible-path solution method, while the latter is employed in a feasible-path solution approach). A simultaneous approach discretizes the dynamic equations so that the problem is transformed to a conventional static optimization problem. The system model equations are then imposed as a number of algebraic equality constraints, and system variables are introduced as additional decision variables (Bequette, 1991; Chachuat, Singer, & Barton, 2005). The problem with this method is that it produces an exponentially amplified number of decision variables and equality constraints. A sequential approach, on the other hand, uses an integrator to solve the differential equations in an inner subroutine, hence creating a pseudo-static correlation between the design decisions and dynamic properties such as control performance. This results in a small number of black-box functions that have to be evaluated during the optimization which does not increase the dimensionality of the problem. However, since the inner routine typically involves numerical solution techniques, it produces very non-linear correlations, which can create various issues in problem solving.

The IPDC problem here is formulated as a bi-level optimization problem, which is then solved using a two-stage sequential, feasible-path method that keeps the problem size manageable. This new problem consists of a main optimization step with simultaneous and sequential, path constraints.

This sub-problem is with reduced size and only the control part more efficiently. The pseudo-static design problem does not include the dynamic variables as decision variables, and enables the evaluation of the dynamic functions (constraints and the second part of the objective) as if they are static functions of the decision variables. The dynamic part, on the other hand, is specifically formulated so as to be solvable separately and analytically. In this way, computational requirements on this most demanding part of the IPDC problem can be significantly reduced.

2.3. Dynamic optimization via modified linear quadratic regulator

In the inner, dynamic optimization, the objective is to minimize the integral of squared deviations from the set points. Eq. (28) is a system model, with its nominal conditions in Eqs. (14) and (15). This sub-problem is with reduced size and only the control variables appear as decision variables.

Different control schemes, such as MPC and parametric controllers, can and have been used to solve such dynamic optimization problems (e.g., Uygun, 2004). Non-linear constrained MPC is a general and flexible approach, but the high computational requirements and possible convergence problems make the approach limited.

The essential idea in this work is to use the LQR to solve this problem, with the mathematical elegance of optimal control, to reduce the computational load. As Lewis stated (1992), “Since naturally occurring systems exhibit optimality in their motion, it makes sense to design man-made control systems in an optimal fashion”. The LQR (as well as the linear quadratic Gaussian regulator) was introduced by Kalman (1960a,b) to whom the development of many modern control concepts can be attributed. Kalman studied the LQR problem where the control action is assumed through the manipulation of rate of change of the control variables, , rather than direct manipulation, . The reason of this change is to incorporate integral action into the control solution, which will be discussed in Section 2.3. A major advantage of this formulation is that the control and design problems are compartmentalized, which enables focusing on the more problematic control part more efficiently. The pseudo-static design problem does not include the dynamic variables as decision variables, and enables the evaluation of the dynamic functions (constraints and the second part of the objective) as if they are static functions of the decision variables. The dynamic part, on the other hand, is specifically formulated so as to be solvable separately and analytically. In this way, computational requirements on this most demanding part of the IPDC problem can be significantly reduced.

\[
\begin{align*}
\min \ J &= \Phi(d, x^d, u^d) \\
& \quad + \frac{\lambda}{2} \int_0^\infty (\ddot{x}^T \cdot Q \cdot \ddot{x} + \dddot{u}^T \cdot R_d \cdot \dddot{u} + \dddot{u}^T \cdot R_\theta \cdot \dddot{\Theta}) \, dt \\
\text{s.t.} \\
& \quad f^d(d, x^d, u^d) = 0 \\
& \quad x^d_{\min} \leq x^d \leq x^d_{\max} \\
& \quad u^d_{\min} \leq u^d \leq u^d_{\max} \\
& \quad h^d(d, x^d, u^d, \Theta^d) = 0 \\
& \quad g^d(d, x^d, u^d, \Theta^d) \leq 0 \\
& \quad x^d_{\min} \leq x(t) \leq x^d_{\max} \\
& \quad u^d_{\min} \leq u(t) \leq u^d_{\max}
\end{align*}
\]
be mathematically defined as follows:

$$\min_{\bar{u}(t)} J = \frac{1}{2} \int (\bar{x}^T \cdot Q \cdot \bar{x} + \bar{u}^T \cdot R \cdot \bar{u}) \, dt$$  \hspace{1cm} (29)$$

$$\bar{x} = A\bar{x}(t) + B\bar{u}(t)$$  \hspace{1cm} (30)$$

The objective function $J$ penalizes the squared input and state deviations from the origin, and includes separate state and input weight matrices $Q$ and $R$ to allow for tuning trade-offs. Note that here we assume a linear/linearized model, and both the state and control variables are taken as the deviation variables; appropriate methods to obtain linearized models will be discussed in the following sections. The solution of the LQR problem yields a state-feedback proportional controller (Bryson and Ho, 1975), with a gain matrix $K$ computed from the solution of the algebraic Riccati equation (ARE) specified in the following equation:

$$A^T \cdot S + S \cdot A + Q - S \cdot B \cdot R^{-1} \cdot B^T \cdot S = 0$$  \hspace{1cm} (31)$$

$$\bar{u} = -K \cdot \bar{x}(t)$$  \hspace{1cm} (32)$$

$$K = R^{-1} \cdot B^T \cdot S$$  \hspace{1cm} (33)$$

It should be noted that the ARE solution uses a steady-state gain matrix, and is sub-optimal with respect to the full (time-dependent) Riccati equation. This sub-optimality is generally negligible, and very often the ARE is preferred since its solution is much simpler than that from a full Riccati equation.

The traditional LQR formulation produces a proportional-only state-feedback controller, similar to the parametric controllers (Sakizlis et al., 2004). Although a state-feedback controller performs (with a proper tuning), in general, better than a collection of traditional single-input single-output controllers, a proportional-only controller may lead to either steady-state offsets or overly aggressive control variable movements, and it is not possible with this formulation to tune for a smoother controller profile. To overcome this shortcoming, it is desirable to reformulate the optimal feedback control problem so as to allow a smoother integral control action. The integral action can be included by introducing a penalty term for the manipulated variables movement. This leads to a modified LQR (i.e., mLQR) as follows:

$$\min_{\bar{u}(t)} J = \frac{1}{2} \int (\bar{x}^T \cdot Q \cdot \bar{x} + \bar{u}^T \cdot R \cdot \bar{u} + \bar{u}^T \cdot R \cdot \dot{\bar{u}}) \, dt$$  \hspace{1cm} (34)$$

This is exactly the same as Eq. (27). This alternative objective formulation can be handled by some simple modifications, which yields a proportional integral (PI) control law as stated below (see Appendix A for derivation and calculation of gain matrices $K_1$ and $K_2$):

$$\bar{u}(t) = -K_2 \cdot \bar{x}(t) - K_1 \cdot \int_0^t \dot{\bar{x}}(t) \, dt$$  \hspace{1cm} (35)$$

The solution of the mLQR problem gives an optimally tuned state-feedback PI control scheme. A major advantage of this approach is that an explicit control law is derived. Therefore, given a candidate design (i.e., matrices $A$ and $B$) and an objective function, the ARE can be solved to calculate the optimal gain matrices. The solution of the ARE, while not trivial, is very quick and fairly reliable. Once the gain matrices are evaluated, the differential equations describing the state and control variables can be integrated simultaneously to yield a simulation with the optimal mLQR controller, thus enabling evaluation of any dynamic performance measures desired.

2.4. Implementation

The solution of the mLQR problem yields an optimally tuned state-feedback PI control scheme, and simulation with this controller enables evaluation of the entire set of path constraints as well as the control performance. The performance of the optimal controller is the best achievable performance for the defined problem, and thus is a fair method for comparing the actual control performances that will be displayed by different candidate designs. The centralized control scheme enables avoiding the complexity of control system design. However, since the mLQR scheme does not require a large number of on-line optimizations during simulation, the computational efficiency is vastly improved as compared to MPC, rendering the introduced IPDC-mLQR method a practical solution for IPDC problems.

For implementation, the modified LQR is used as the solution of the dynamic optimization. The main optimization then acquires the evaluation of the dynamic performance, combines this information with the static criteria such as cost and flexibility constraints, and produces a new candidate by adjusting the vector of design variables. The new candidate is then evaluated in the dynamic optimization part (with a new optimal controller based on the new design), until the iterations converge on a feasible design that cannot be improved further. Fig. 1 depicts the proposed sequential optimization algorithm. Note that this strategy is sequential since the dynamic optimization is solved with an integration routine as opposed to a discretization of the system equations. The sequential nature of the evaluation algorithm does not change the fact that the methodology evaluates a design via simultaneously considering cost and controllability. Also note that the focus of this work is on finding a practical method for the solution of control problem, and the design problem is solved with a fairly generic method. Development of IPDC methods taking further advantage of the basic ideas proposed here could prove a profitable direction of research. Implementation of global mixed-integer dynamic optimization techniques (Chachuat et al., 2005), for instance, appears to be the next logical step.

It should be noted that a linear system model is required for the mLQR setup. However, this requirement can be satisfied for most non-linear systems by linearization of the model around the nominal conditions. Depending on the non-linearity of the model, the linearization can be performed either once at the start of the simulation (i.e., once for each candidate design), or multiple times at a desired frequency during the simulation in an adaptive-control like fashion. The linearization approach produces satisfactory results for most cases (Morari and Lee, 1999). One other attractive alternative is to construct multiple linear models to approximate the actual system model at the start of the simulation, and corresponding optimal controllers, and then...
switching between these models and controllers as necessary during simulation (Banerjee and Arkun, 1998).

The LQR formulation does not allow for the inclusion of inequality constraints in the dynamic optimization. Accordingly, the path constraints (Eqs. (18)–(21)) are imposed during the main optimization routine. Ideally, these constraints should be imposed within the dynamic optimization. However, as the control problem is implemented within the design problem, it is possible to encounter designs that cannot satisfy these path constraints (i.e., designs that are completely infeasible). The solution algorithm should be able to continue iterations in such cases, which means that these constraints have to be formulated either as soft-constraints (i.e., violations penalized in the objective) or be hierarchically dropped if impossible to meet them. Either case is very sophisticated for direct implementation within an IPDC algorithm. Hence, the algorithm employed here does not enforce the constraints in the dynamic optimization (i.e., the constraints are not considered during controller design). The final designs will adhere to these constraints as they are evaluated and enforced in the main optimization. It should be noted however that omission of these constraints during controller design will likely result in sub-optimality; this is accepted as a trade-off to avoid to the exponential increase in problem complexity and computational burden.

3. Case study—integrated design and control of an evaporation process

To test the effectiveness of the developed method, an evaporation process, which was investigated by Newell and Lee (1989), and Kookos and Perkins (2001) is studied. This process removes a volatile liquid from a non-volatile solute through evaporation with steam, producing a concentrated product stream as the bottom product. Fig. 2 depicts the process. It uses two heat exchangers as the evaporator and the condenser, and a pump for recycling part of the condensate stream. The flow rate of coolant and the inlet pressure of steam are used as manipulated variables to adjust product composition and operating pressure.

3.1. System model

The evaporator system is modeled by a set of differential and algebraic equations, as given by Kookos and Perkins (2001) and reproduced here:

\[ M \frac{dC_2}{dt} = F_1 C_1 - F_2 C_2 \]  \hspace{1cm} (36)

\[ C \frac{dF_2}{dt} = F_4 - F_5 \]  \hspace{1cm} (37)

A number algebraic equations are derived based on mass and energy balances:

\[ F_1 - F_4 - F_2 = 0 \]  \hspace{1cm} (38)

\[ F_1 C_p T_1 - F_4 (\lambda + C_p T_4) - F_2 C_p T_2 + Q_{100} = 0 \]  \hspace{1cm} (39)

\[ 0.5616 P_2 + 0.3126 C_2 + 48.43 - T_2 = 0 \]  \hspace{1cm} (40)

\[ 0.5070 P_2 + 55 - T_4 = 0 \]  \hspace{1cm} (41)

\[ 0.1538 P_{100} + 90 - T_{100} = 0 \]  \hspace{1cm} (42)

\[ Q_{100} - UA_1 (T_{100} - T_2) = 0 \]  \hspace{1cm} (43)

\[ Q_{100} - F_{100} \lambda_s = 0 \]  \hspace{1cm} (44)

\[ Q_{200} - F_{200} C_p (T_{201} - T_{200}) = 0 \]  \hspace{1cm} (45)
There are a number of inequality constraints in operation:

\begin{align}
25 - C_2 &\leq 0 \\
C_2 - 30 &\leq 0 \\
40 - P_2 &\leq 0 \\
P_2 - 80 &\leq 0 \\
P_{100} - 400 &\leq 0 \\
F_{200} - 600 &\leq 0
\end{align}

where $F_1$ is the feed flow rate (kg/min), $F_2$ the product flow rate (kg/min), $F_3$ the circulation flow rate (kg/min), $F_4$ the vapor flow rate (kg/min), $F_5$ the condensate flow rate (kg/min), $F_{100}$ the steam flow rate (kg/min), $F_{200}$ the cooling water flow rate (kg/min), $T_1$ the feed temperature (°C), $T_2$ the product temperature (°C), $T_3$ the vapor temperature (°C), $T_{100}$ the steam temperature (°C), $T_{200}$ the cooling water inlet temperature (°C), $T_1$ the steam temperature (°C), $C_1$ the feed composition (%), $C_2$ the product composition (%), $P_2$ the pressure (kPa), $P_{100}$ the steam pressure (kPa), $A_1$ and $A_2$ are heat transfer areas of evaporator and condenser, respectively (m²).

It should be noted that the algebraic Eqs. (38)–(47) are linear and well defined. This has allowed the algebraic substitution of these variables, when the differential Eqs. (36) and (37) were being solved. This eliminates the need for a differential-algebraic solver. In the original design, the flow rate of coolant, and the inlet pressure of steam are used as manipulated variables to adjust product composition and operating pressure.

### 3.2. Problem description

The design objective is to minimize the TAC, including the utility cost of steam and cooling water and the equipment cost. To ensure the choice of manipulated variables is not trivial, the costs of measuring devices and control valves are also included in the design cost, therefore reflecting the drawback in using the additional manipulated variables. The cost functions and parameters used in calculations are listed in Table 1.

It should be noted that the design is required to handle a multi-period operation both at steady-state and dynamically. The first period corresponds to the nominal case with no disturbances. The other four periods represent the four vertices of the disturbance space with ±5% disturbances in the feed flow rate and inlet composition. The system is required to be capable of functioning with these disturbances at steady-state by virtue of Eqs. (18)–(22) in the problem formulation. Also, the system should be able to satisfy the path constraints which require that the concentration remains between 25 and 30%, the pressure remains within 40–80 kPa, the coolant flow rate can-
was less than 10 min (on a Pentium 4 desktop PC). The total time requirement separately using analysis of this study problem. All the four cases were solved separately with an NLP solver, which enables better controllers, rather than employing an MINLP solver, each case was solved separately with an NLP solver, which enables better analysis of this study problem. All the four cases were solved separately using *fmincon* non-linear constrained optimization programming solver in MATLAB. The total time requirement was less than 10 min (on a Pentium 4 desktop PC).

not exceed 600 kg/min, and the steam pressure cannot exceed 400 kPa. These constraints correspond to Eqs. (23)–(26) in the general formulation. The same operational limits are imposed statically and dynamically; hence the constraints listed should be met by the systems at all times. Note that it is possible to specify different limits for steady-state and dynamic behavior, but this path has been avoided in order not to unnecessarily complicate the case study.

The inclusion of the controller selection decisions in the design problem results in a mixed-integer non-linear programming (MINLP) problem. Since the number of combinations are only four (Case I—no controller, Case II—pressure control only, Case III—flow control only, and Case IV—with both controllers), rather than employing an MINLP solver, each case was solved separately with an NLP solver, which enables better analysis of this study problem. All the four cases were solved separately using *fmincon* non-linear constrained optimization programming solver in MATLAB. The total time requirement was less than 10 min (on a Pentium 4 desktop PC).

Feasible results were attained for all the cases, except for the case employing no controllers. For Case I (employing no controllers), the operational constraints requirement that the concentration should not fall below 25% could not be satisfied. Thus, an infeasible result was obtained. Case IV (employing both the controllers) is the optimal design, which results in the least total annualized cost and a control performance that matches the required static and dynamic constraints. The total annualized cost for the optimal design is US$ 2340 less then the original design and control performance is also much better when compared to original design. The results of optimization are summarized in Table 2 (this problem/design is designated as optimal design I). Fig. 3 depicts the disturbance profile, and Fig. 4 depicts the effect of disturbance on the uncontrolled system. Figs. 5–8 display the dynamic response of the optimal design under mLQR control.

### Table 1

**Objective and cost functions used in the case study**

<table>
<thead>
<tr>
<th>Objective function</th>
<th>Utility cost = controller cost + equipment cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility cost ($)</td>
<td>Utility flow rate × cost factor × total hours of operation per year</td>
</tr>
<tr>
<td>Steam</td>
<td>(F_{100} \times 0.6 \times 24 \times 320)</td>
</tr>
<tr>
<td>Cooling water</td>
<td>(F_{200} \times 0.00061 \times 24 \times 320)</td>
</tr>
<tr>
<td>Controller cost ($)</td>
<td>Controller cost × depreciation factor</td>
</tr>
<tr>
<td>Steam controller</td>
<td>6000/2</td>
</tr>
<tr>
<td>Cooling water controller</td>
<td>4000/2</td>
</tr>
</tbody>
</table>

**Equipment cost ($) = evaporator + condenser + shell + recycle pump**

<table>
<thead>
<tr>
<th>Total equipment</th>
<th>Equipment installed cost × depreciation factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>Inflation factor (\times 101.3) (\times 1.35)</td>
</tr>
<tr>
<td>Inflation factor</td>
<td>Index value at present/index value at base year</td>
</tr>
<tr>
<td>Correction factor</td>
<td>(design factor + pressure factor) (\times) material factor</td>
</tr>
<tr>
<td>Evaporator</td>
<td>(394.3/119 \times 101.3 \times A_1^{0.65} \times (2.29 + 2.81 \times 1.35)/10)</td>
</tr>
<tr>
<td>Condenser</td>
<td>(394.3/119 \times 101.3 \times 200.65 \times (2.29 + 1 \times 0.85)/10)</td>
</tr>
<tr>
<td>Shell</td>
<td>45,000/10</td>
</tr>
<tr>
<td>Recycle pump</td>
<td>10,000/5</td>
</tr>
</tbody>
</table>

### Table 2

**Results**

<table>
<thead>
<tr>
<th>Decisions</th>
<th>Original design (both controllers exist)</th>
<th>Optimal design (both controllers exist)</th>
<th>Alternate design (no controller exists)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decisions</td>
<td>(UA_1 = 9.60, UA_2 = 6.84)</td>
<td>(UA_1 = 4.93, UA_2 = 8.49)</td>
<td>(UA_1 = 5.30, UA_2 = 8.67)</td>
</tr>
<tr>
<td>Nominal control values</td>
<td>(P_{100} = 193.43, F_{200} = 207.36)</td>
<td>(P_{100} = 371.38, F_{200} = 157.76)</td>
<td>(P_{100} = 374.62, F_{200} = 164.75)</td>
</tr>
<tr>
<td>Nominal state values</td>
<td>(C_2 = 25.00, P_2 = 50.57)</td>
<td>(C_2 = 25.62, P_2 = 40.45)</td>
<td>(C_2 = 31.09, P_2 = 41.92)</td>
</tr>
<tr>
<td>Total annualized cost ($)</td>
<td>64,136</td>
<td>61,795</td>
<td>58,812</td>
</tr>
<tr>
<td>Dynamic constraints</td>
<td>(30 &gt; C_2 &gt; 25, 80 &gt; P_1 &gt; 40, 600 &gt; P_{100} &gt; 101.33, 400 &gt; F_{200} &gt; 0)</td>
<td>(30 &gt; C_2 &gt; 25, 80 &gt; P_1 &gt; 40, 600 &gt; P_{100} &gt; 101.33, 400 &gt; F_{200} &gt; 0)</td>
<td>(100 &gt; C_2 &gt; 25, 80 &gt; P_1 &gt; 40, 600 &gt; P_{100} &gt; 101.33, 400 &gt; F_{200} &gt; 0)</td>
</tr>
<tr>
<td>Manipulated variable bounds</td>
<td>(600 &gt; P_{100} &gt; 101.33, 400 &gt; F_{200} &gt; 0)</td>
<td>(600 &gt; P_{100} &gt; 101.33, 400 &gt; F_{200} &gt; 0)</td>
<td>(600 &gt; P_{100} &gt; 101.33, 400 &gt; F_{200} &gt; 0)</td>
</tr>
<tr>
<td>Constraint adherence</td>
<td>violation of dynamic constraints (minimum composition)</td>
<td>Conforms to constraints</td>
<td>Conforms to constraints</td>
</tr>
</tbody>
</table>
Fig. 4. Effect of disturbance on the uncontrolled system.

Fig. 5. Dynamic behavior of original design (dotted line represents the constraint: $C_2 > 25\%$).

Fig. 6. Dynamic behavior of optimal design (dotted line represents the constraint: $C_2 > 25\%$).

Remarkably, it was observed that the optimal solution involves no controllers, results in the least annualized cost, and provides a poor control performance that nevertheless matches the required operational constraints. The result of optimization is summarized in Table 2 under the “Alternate design” column. The result is interesting and counterintuitive in nature, as the result of an integration of process design and control study is a design without controllers. Despite the lack of controllers, the concentration fluctuates between 25 and 40%, and the operational constraints are not violated (see Fig. 9). This is evidently achieved by overdesigning the system so that the nominal operation concentration is about 33%. The eliminated cost of the controllers compensate for the cost of this overdesign. It should be noted that, the case with both controllers existent results in a design with excellent control performance, but one that has an annualized cost of US$ 61,795, significantly more than Case II with the cost of US$ 58,812.

While counterintuitive, Case II demonstrates the true value of IPDC algorithms, i.e., novel designs that push the boundaries to create the optimal for the particular problem studied. The objective in the problem here (and in general design problems) is to minimize the total operational cost, as long as the operational constraints can be satisfied. As demonstrated above, slightly different operational constraints can result in dramatically different designs with true IPDC algorithms, whereas the

Fig. 7. Manipulated variable movements of the original design.

Fig. 8. Manipulated variable movements of the optimal design.
In this work, an optimal-control-based approach has been introduced for achieving integration of process design and control in a practical manner. The principal idea proposed is to utilize an optimal controller (a modified linear quadratic regulator) to practically evaluate the best achievable control performance for each candidate design during the process synthesis stage. The evaluation of complete and detailed closed-loop system dynamics are then meshed with a process design algorithm, thus enabling consideration of both cost and operability in design of the process. The practicality and computational efficiency introduced by the IPDC-mLQR method enables the solution of this complex dynamic optimization problem within a very reasonable computational time.

In the IPDC-mLQR method, the use of optimal control ensures that each candidate design is fully evaluated in terms of its closed-loop control performance, whereas the design algorithm employed ensures all static and dynamic criteria are simultaneously considered. As a result, the IPDC-mLQR algorithm is computationally far more effective compared to methods that involve heavy online-optimization, such as model predictive control. On the other hand, the control performance assessment method is not questionable as is the case with the methods employing limited control structures.

It should be noted, however, that while the IPDC-mLQR algorithm is computationally more practical as compared to IPDC approaches employing parametric controllers, this advantage comes at the expense of ability to address path constraints during the mLQR controller design stage. However, in our opinion, the possibility of infeasibility renders full consideration of these constraints extremely problematic. The proposed approach considers these constraints only within the static design procedure, which should be a satisfactory approximation for most cases. However, it should be re-emphasized that this is a premeditated trade-off between computational cost and accuracy.

The proposed IPDC-mLQR algorithm presents a method that can be utilized to design or improve process plants through integration of process design and control. However, it has certain limitations. One of them is that the algorithm allows checking only certain predetermined points in the disturbance space, which should logically correspond to the vertices of the uncertainty space for the system. While the number of points checked can be increased, it is not possible to consider the entire uncertainty space with the current algorithm. The problem, however, can be resolved by adding an additional layer of iterations for ensuring dynamic flexibility, as outlined by Sakizlis et al. (2004).

A major advantage of the optimal control-based approach is that a unique control law is identified for each candidate process. One possible idea that is currently being explored is to create multiple linear approximations to the process model, combine them with the mLQR control law identified (which can also be expressed in linear form), and then solve the system differential equations analytically. This approach would result in a single unified MINLP problem that could be solved much more efficiently, and could also help resolve some of the issues with constraints in the dynamic optimization.

Acknowledgments

This work is in part supported by NSF (CTS 0091398), ACS-PRF, and the Institute of Manufacturing Research of Wayne State University.

Appendix A

Derivation of Eq. (35).

The mLQR objective is:

$$\min_{\tilde{u}(t)} J = \frac{1}{2} \int (\ddot{x}^T \cdot \dot{x} + \tilde{u}^T \cdot R_\alpha \cdot \ddot{u} + \dot{\tilde{u}}^T \cdot R_\beta \cdot \dot{\tilde{u}}) \, dt$$  \hspace{1cm} (34)$$

First, the state equation in Eq. (30) is differentiated, which yields:

$$\ddot{x}(t) = A\dot{x}(t) + B\tilde{u}(t)$$  \hspace{1cm} (A-1)$$
This new state equation can be rewritten as:

\[
\mathbf{w} = \begin{bmatrix} 0 & 1 \\ 0 & \mathbf{A} \end{bmatrix} \cdot \mathbf{w} + \begin{bmatrix} 0 \\ \mathbf{B} \end{bmatrix} \cdot \mathbf{v} \tag{A-2}
\]

where

\[
\mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}(t) \\ \hat{\mathbf{s}}(t) \end{bmatrix} \tag{A-3}
\]

\[
\mathbf{v}(t) = \hat{\mathbf{u}}(t) \tag{A-4}
\]

Therefore, the objective function becomes:

\[
\min_{\mathbf{v}(t)} J = \frac{1}{2} \int (\mathbf{w}^T \cdot \hat{\mathbf{Q}} \cdot \mathbf{w} + \mathbf{v}^T \cdot \hat{\mathbf{R}} \cdot \mathbf{v}) \, dt \tag{A-5}
\]

Note that Eq. (A-5) is different from Eq. (29), as an additional term for derivative of the control variables also appears in it. The modified ARE can be written as:

\[
\hat{\mathbf{A}}^T \cdot \hat{\mathbf{S}} + \hat{\mathbf{S}} \cdot \hat{\mathbf{A}} + \hat{\mathbf{Q}} - \hat{\mathbf{S}} \cdot \hat{\mathbf{B}} \cdot \hat{\mathbf{R}}^{-1} \cdot \hat{\mathbf{B}}^T \cdot \hat{\mathbf{S}} = 0 \tag{A-6}
\]

where

\[
\hat{\mathbf{A}} = \begin{bmatrix} 0 & 1 \\ 0 & \mathbf{A} \end{bmatrix} \tag{A-7}
\]

\[
\hat{\mathbf{B}} = \begin{bmatrix} 0 \\ \mathbf{B} \end{bmatrix} \tag{A-8}
\]

\[
\hat{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & 0 \\ 0 & 0 \end{bmatrix} \tag{A-9}
\]

\[
\hat{\mathbf{R}} = \mathbf{R} \tag{A-10}
\]

The solution of the above stated Riccati equation gives feedback control law:

\[
\mathbf{v} = -\hat{\mathbf{K}} \cdot \mathbf{w}(t) \tag{A-11}
\]

where the gain matrix, \(\hat{\mathbf{K}}\), can be split into gain matrices \(\mathbf{K}_1\) (integral term) and \(\mathbf{K}_2\) (proportional term), and thus,

\[
\hat{\mathbf{u}}(t) = -\left[ \mathbf{K}_1 \quad \mathbf{K}_2 \right] \cdot \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{s}} \end{bmatrix} \tag{A-12}
\]

Integrating Eq. (A-12) yields a proportional integral (PI) control law as stated below:

\[
\hat{\mathbf{u}}(t) = -\mathbf{K}_2 \cdot \hat{\mathbf{x}}(t) - \mathbf{K}_1 \int_0^t \hat{\mathbf{x}}(t) \, dt \tag{35}
\]

References


