PERGAMON

Automatica 35 (1999) 407–427

Control of systems integrating logic, dynamics, and constraints[1]

Alberto Bemporad, Manfred Morari*

Institut für Automatik, ETH - Swiss Federal Institute of Technology, ETHZ - ETL, CH 8092 Zürich, Switzerland

Received 17 March 1998; received in final form 16 September 1998

Systems described by interdependent physical laws, logic rules, and operating constraints are described by linear equations and inequalities involving continuous and integer variables. Model predictive control based on mixed-integer quadratic programming provides a systematic controller synthesis procedure.

Abstract

This paper proposes a framework for modeling and controlling systems described by interdependent physical laws, logic rules, and operating constraints, denoted as *mixed logical dynamical* (MLD) systems. These are described by linear dynamic equations subject to linear inequalities involving real and integer variables. MLD systems include linear hybrid systems, finite state machines, some classes of discrete event systems, constrained linear systems, and nonlinear systems which can be approximated by piecewise linear functions. A predictive control scheme is proposed which is able to stabilize MLD systems on desired reference trajectories while fulfilling operating constraints, and possibly take into account previous qualitative knowledge in the form of heuristic rules. Due to the presence of integer variables, the resulting on-line optimization procedures are solved through *mixed integer quadratic programming* (MIQP), for which efficient solvers have been recently developed. Some examples and a simulation case study on a complex gas supply system are reported.

Keywords: Hybrid systems; Predictive control; Dynamic models; Binary logic systems; Boolean logic; Mixed-integer programming; Optimization problems

1. Introduction

The concept of *model* of a system is traditionally associated with differential or difference equations, typically derived by physical laws governing the *dynamics* of the system under consideration. Consequently, most of the control theory and tools have been developed for such systems, in particular for systems whose evolution is described by smooth linear or nonlinear state transition functions. On the other hand, in many applications the system to be controlled is also constituted by parts described by *logic*, such as for instance on/off switches or valves, gears or speed selectors, evolutions dependent on if-then-else rules. Often, the control of these systems is left to schemes based on heuristic rules inferred by practical plant operation.

Recently, in the literature researchers started dealing with *hybrid systems*, namely hierarchical systems constituted by dynamical components at the lower level, governed by upper level logical/discrete components (Grossmann et al., 1993; Branicky et al., 1998). Hybrid systems arise in a large number of application areas, and are attracting increasing attention in both academic theory-oriented circles as well as in industry. Our interest is motivated by several clearly discernible trends in the process industries which point toward an extended need for new tools to design control and supervisory schemes for hybrid systems and to analyze their performance.

For this class of systems, design procedures have been proposed which naturally lead to hierarchical, hybrid

* Corresponding author. Tel.: +41 1 632 7626; fax: +41 1 632 1211; e-mail: morari@aut.ee.ethz.ch.

[1] This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor A. L. Tits under the direction of Editor T. Başar, and accepted by the Guest Editors J. M. Schumacher, A. S. Morse, C. C. Pantelides, and S. Sastry.

0005-1098/99/$—see front matter
PII: S0005-1098(98)00178-2

control schemes, with continuous controllers at the lower level calibrated for each dynamical subsystem in order to provide regulation and tracking properties, and discrete controllers supervising, resolving conflicts, and planning strategies at a higher level (Lygeros et al., 1996). However, in some applications a precise distinction between different hierarchic levels is not possible, especially when dynamical and logical facts are dramatically interdependent. For such a class of systems, not only it is not clear how to design feedback controllers, but even how to obtain models in a systematic way.

This paper proposes a framework for modeling and controlling models of systems described by interacting physical laws, logical rules, and operating constraints. According to techniques described e.g. in Williams (1993), Cavalier et al. (1990) and Raman and Grossmann (1992), propositional logic is transformed into linear inequalities involving integer and continuous variables. This allows to arrive at *mixed logical dynamical* (MLD) systems described by linear dynamic equations subject to linear mixed-integer inequalities, i.e. inequalities involving both *continuous* and *binary* (or *logical*, or *0–1*) variables. These include physical/discrete states, continuous/integer inputs, and continuous/binary auxiliary variables. MLD systems generalize a wide set of models, among which there are linear hybrid systems, finite state machines, some classes of discrete event systems, constrained linear systems, and nonlinear systems whose nonlinearities can be expressed (or, at least, suitably approximated) by piecewise linear functions.

Mixed-integer optimization techniques have been investigated in (Raman and Grossmann, 1991; Raman and Grossmann, 1992), for chemical process synthesis. For feedback control purposes, we propose a predictive control scheme which is able to stabilize MLD systems on desired reference trajectories while fulfilling operating constraints, and possibly take into account previous qualitative knowledge in the form of heuristic rules. Moving horizon optimal control and model predictive control have been widely adopted for tracking problems of systems subject to constraints (Lee and Cooley, 1997; Mayne, 1997; Qin and Badgewell, 1997). These methods are based on the so called *receding horizon* philosophy: a sequence of future control actions is chosen according to a prediction of the future evolution of the system and applied to the plant until new measurements are available. Then, a new sequence is determined which replaces the previous one. Each sequence is evaluated by means of an optimization procedure which take into account two objectives: optimize the tracking performance, and protect the system from possible constraint violations. In the present context, due to the presence of integer variables, the optimization procedure is a *mixed integer quadratic programming* (MIQP) problem (Fletcher and Leyffer, 1995; Lazimy, 1985; Roschchin et al., 1987), for which efficient solvers exist (Fletcher and Leyffer, 1994). A first attempt to use on-line mixed-integer programming to control dynamic systems subject to logical conditions has appeared in (Tyler and Morari, n.d.). Other attempts of combining MPC to hybrid control have also appeared in Slupphaug and Foss (1997) and Slupphaug et al. (1997).

This paper is organized as follows. In Section 2 some basic facts from propositional calculus, Boolean algebra, and mixed-integer linear inequalities are reviewed. These tools are used in Section 3 to motivate the definition of MLD systems and provide examples of systems which can be modeled within this framework. Stability definitions and related issues are discussed in Section 4. Section 5 deals with the optimal control of MLD systems and shows how heuristics can eventually be taken into account. These results are then used in Section 6 to develop a *mixed-integer predictive controller* (MIPC), which essentially solves on-line at each time step an optimal control problem through MIQP, and apply the optimal solution according to the aforementioned receding horizon philosophy. A brief description of available MIQP solvers is given in Section 7. Finally, a simulation study on the complex gas supply system reported in Akimoto et al. (1991) is described in Section 8.

2. Propositional calculus and linear integer programming

By following standard notation (Williams, 1977; Cavalier et al., 1990; Williams, 1993), we adopt capital letters X_i to represent statements, e.g. "$x \geq 0$" or "Temperature is hot". X_i is commonly referred to as *literal*, and has a *truth value* of either "T" (true) or "F" (false). Boolean algebra enables statements to be combined in compound statements by means of *connectives*: "$\wedge$" (and), "$\vee$" (or), "$\sim$" (not), "$\rightarrow$" (implies), "$\leftrightarrow$" (if and only if), "$\oplus$" (exclusive or) (a more comprehensive treatment of Boolean calculus can be found in digital circuit design texts, e.g. Christiansen (1997) and Hayes (1993). For a rigorous exposition, see e.g. Mendelson (1964). Connectives are defined by means of the *truth table* reported in Table 1. Other connectives may be similarly defined. Connectives satisfy several properties (see e.g. Christiansen, 1997), which can be used to transform compound statements into equivalent statements involving different connectives, and simplify complex statements. It is known that all connectives can be defined in terms of a subset of them, for instance $\{\vee, \sim\}$, which is said to be a complete set of connectives. Below we report some properties which will be used in the sequel

$$X_1 \rightarrow X_2 \text{ is the same as } \sim X_1 \vee X_2, \tag{1a}$$

$$X_1 \rightarrow X_2 \text{ is the same as } \sim X_2 \rightarrow \sim X_1, \tag{1b}$$

$$X_1 \leftrightarrow X_2 \text{ is the same as } (X_1 \rightarrow X_2) \wedge (X_2 \rightarrow X_1). \tag{1c}$$

Correspondingly one can associate with a literal X_i a *logical variable* $\delta_i \in \{0, 1\}$, which has a value of either 1

Table 1
Truth table

X_1	X_2	$\sim X_1$	$X_1 \vee X_2$	$X_1 \wedge X_2$	$X_1 \rightarrow X_2$	$X_1 \leftrightarrow X_2$	$X_1 \oplus X_2$
F	F	T	F	F	T	T	F
F	T	T	T	F	T	F	T
T	F	F	T	F	F	F	T
T	T	F	T	T	T	T	F

if $X_i = \mathrm{T}$, or 0 otherwise. Integer programming has been advocated as an efficient inference engine to perform automated deduction (Cavalier et al., 1990). A propositional logic problem, where a statement X_1 must be proved to be true given a set of (compound) statements involving literals $X_1, \ldots, X_n$, can be in fact solved by means of a linear integer program, by suitably translating the original compound statements into linear inequalities involving logical variables δ_i. In fact, the following propositions and linear constraints can easily be seen to be equivalent (Williams, 1993, p. 176)

$$X_1 \vee X_2 \text{ is equivalent to } \delta_1 + \delta_2 \geq 1, \tag{2a}$$

$$X_1 \wedge X_2 \text{ is equivalent to } \delta_1 = 1, \delta_2 = 1, \tag{2b}$$

$$\sim X_1 \text{ is equivalent to } \delta_1 = 0, \tag{2c}$$

$$X_1 \rightarrow X_2 \text{ is equivalent to } \delta_1 - \delta_2 \leq 0, \tag{2d}$$

$$X_1 \leftrightarrow X_2 \text{ is equivalent to } \delta_1 - \delta_2 = 0, \tag{2e}$$

$$X_1 \oplus X_2 \text{ is equivalent to } \delta_1 + \delta_2 = 1. \tag{2f}$$

We borrow this computational inference technique to model logical parts of processes (on/off switches, discrete mechanisms, combinational and sequential networks) and heuristics knowledge about plant operation as integer linear inequalities. As we are interested in systems which have both logic and dynamics, we wish to establish a link between the two worlds. In particular, we need to establish how to build statements from operating events concerning physical dynamics. As will be shown in a moment, we end up with *mixed-integer linear inequalities*, i.e. linear inequalities involving both *continuous variables* $x \in \mathbb{R}^n$ and logical (*indicator*) variables $\delta \in \{0,1\}$. Consider the statement $X \triangleq [f(x) \leq 0]$, where $f: \mathbb{R}^n \mapsto \mathbb{R}$ is linear, assume that $x \in \mathscr{X}$, where $\mathscr{X}$ is a given bounded set, and define

$$M \triangleq \max_{x \in \mathscr{X}} f(x), \tag{3a}$$

$$m \triangleq \min_{x \in \mathscr{X}} f(x). \tag{3b}$$

Theoretically, an over[under]-estimate of $M[m]$ suffices for our purpose. However, more realistic estimates provide computational benefits (Williams, 1993, p. 171). It is easy to verify that

$$[f(x) \leq 0] \wedge [\delta = 1] \text{ is true iff } f(x) - \delta \leq -1 + m(1-\delta), \tag{4a}$$

$$[f(x) \leq 0] \vee [\delta = 1] \text{ is true iff } f(x) \leq M\delta, \tag{4b}$$

$$\sim [f(x) \leq 0] \text{ is true iff } f(x) \geq \varepsilon, \tag{4c}$$

where ε is a small tolerance (typically the machine precision), beyond which the constraint is regarded as violated. By Eqs. (1a) and (4b), it also follows

$$[f(x) \leq 0] \rightarrow [\delta = 1] \text{ is true iff } f(x) \geq \varepsilon + (m - \varepsilon)\delta, \tag{4d}$$

$$[f(x) \leq 0] \leftrightarrow [\delta = 1] \text{ is true iff } \begin{cases} f(x) \leq M(1-\delta), \\ f(x) \geq \varepsilon + (m-\varepsilon)\delta. \end{cases} \tag{4e}$$

Finally, we report procedures to transform products of logical variables, and of continuous and logical variables, in terms of linear inequalities, which however require the introduction of *auxiliary variables* (Williams, 1993, p. 178). The product term $\delta_1\delta_2$ can be replaced by an auxiliary logical variable $\delta_3 \triangleq \delta_1\delta_2$. Then, $[\delta_3 = 1] \leftrightarrow [\delta_1 = 1] \wedge [\delta_2 = 1]$, and therefore

$$\delta_3 = \delta_1\delta_2 \text{ is equivalent to } \begin{cases} -\delta_1 + \delta_3 \leq 0, \\ -\delta_2 + \delta_3 \leq 0, \\ \delta_1 + \delta_2 - \delta_3 \leq 1. \end{cases} \tag{5a}$$

Moreover, the term $\delta f(x)$, where $f: \mathbb{R}^n \mapsto \mathbb{R}$ and $\delta \in \{0,1\}$, can be replaced by an auxiliary real variable $y \triangleq \delta f(x)$, which satisfies $[\delta = 0] \rightarrow [y = 0], [\delta = 1] \rightarrow [y = f(x)]$. Therefore, by defining M, m as in Eq. (3), $y = \delta f(x)$ is equivalent to

$$\begin{aligned} &y \leq M\delta, \\ &y \geq m\delta, \\ &y \leq f(x) - m(1-\delta), \\ &y \geq f(x) - M(1-\delta). \end{aligned} \tag{5b}$$

Alternative methods and formulations for transforming propositional logic problems into equivalent integer programs exist. For instance, Cavalier et al. (1990) compare the approach above with the approach which utilizes *conjunctive normal forms* (CNF), and conclude that efficiency of a modeling approach depends on the form of logical statements. The problem of finding *minimal forms*, is also well known in the digital network design realm, where the need arises to minimize the number of gates and connections. A variety of methods exist to perform such a task. The reader is referred to Hayes (1993, Chapter 5) for a detailed exposition.

3. Mixed logical dynamical (MLD) systems

In the previous section we have provided some tools to transform logical facts involving continuous variables into linear inequalities. These tools will be used now to express relations describing the evolution of systems where physical laws, logic rules, and operating constraints are interdependent. Before giving a general definition of such a class of systems, consider the following system:

$$x(t+1) = \begin{cases} 0.8x(t) + u(t) & \text{if } x(t) \geq 0, \\ -0.8x(t) + u(t) & \text{if } x(t) < 0, \end{cases} \tag{6}$$

where $x(t) \in [-10, 10]$, and $u(t) \in [-1, 1]$. The condition $x(t) \geq 0$ can be associated to a binary variable $\delta(t)$ such that

$$[\delta(t) = 1] \leftrightarrow [x(t) \geq 0]. \tag{7}$$

By using the transformation (4e), Eq. (7) can be expressed by the inequalities

$$-m\delta(t) \leq x(t) - m,$$

$$-(M + \varepsilon)\delta \leq -x - \varepsilon,$$

where $M = -m = 10$, and ε is a small positive scalar. Then Eq. (6) can be rewritten as

$$x(t+1) = 1.6\delta(t)x(t) - 0.8x(t) + u(t). \tag{8}$$

By defining a new variable $z(t) = \delta(t)x(t)$ which, by Eq. (5b), can be expressed as

$$z(t) \leq M\delta(t),$$

$$z(t) \geq m\delta(t),$$

$$z(t) \leq x(t) - m(1 - \delta(t)),$$

$$z(t) \geq x(t) - M(1 - \delta(t)),$$

the evolution of system (6) is ruled by the linear equation

$$x(t+1) = 1.6z(t) - 0.8x(t) + u(t)$$

subject to the linear constraints above. This example can be generalized by describing mixed logical dynamical (MLD) systems through the following linear relations:

$$x(t+1) = A_t x(t) + B_{1t}u(t) + B_{2t}\delta(t) + B_{3t}z(t) \tag{9a}$$

$$y(t) = C_t x(t) + D_{1t}u(t) + D_{2t}\delta(t) + D_{3t}z(t) \tag{9b}$$

$$E_{2t}\delta(t) + E_{3t}z(t) \leq E_{1t}u(t) + E_{4t}x(t) + E_{5t} \tag{9c}$$

where $t \in \mathbb{Z}$,

$$x = \begin{bmatrix} x_c \\ x_\ell \end{bmatrix}, \quad x_c \in \mathbb{R}^{n_c},\ x_\ell \in \{0,1\}^{n_\ell},\ n \triangleq n_c + n_\ell$$

is the state of the system, whose components are distinguished between continuous x_c and 0–1 x_ℓ;

$$y = \begin{bmatrix} y_c \\ y_\ell \end{bmatrix}, \quad y_c \in \mathbb{R}^{p_c},\ y_\ell \in \{0,1\}^{p_\ell},\ p \triangleq p_c + p_\ell$$

is the output vector,

$$u = \begin{bmatrix} u_c \\ u_\ell \end{bmatrix}, \quad u_c \in \mathbb{R}^{m_c},\ u_\ell \in \{0,1\}^{m_\ell},\ m \triangleq m_c + m_\ell$$

is the command input, collecting both continuous commands u_c, and binary (on/off) commands u_ℓ (discrete commands, i.e. assuming values within a finite set of reals, can be modeled as 0–1 commands, as described later); $\delta \in \{0,1\}^{r_\ell}$ and $z \in \mathbb{R}^{r_c}$ represent respectively auxiliary logical and continuous variables.

The form (9) involves linear discrete-time dynamics. One might formulate a continuous time version by replacing $x(t+1)$ by $\dot{x}(t)$ in Eq. (9a), or a nonlinear version by changing the linear equations and inequalities in Eq. (9) to more general nonlinear functions. We restrict the dynamics to be linear and discrete-time in order to obtain computationally tractable control schemes, as will be described in the next sections. Nevertheless, we believe that this framework permits the description of a very broad class of systems.

In principle, the inequality in Eq. (9) might be satisfied for many values of $\delta(t)$ and/or $z(t)$. On the other hand, we wish that $x(t+1)$ and $y(t)$ were uniquely determined by $x(t)$ and $u(t)$. To this aim, we introduce the following definition

Definition 1. Let $\mathcal{I}_{Bt}$ denote the set of all indices $i \in \{1, \ldots, r_\ell\}$, such that $[B_{2t}]^i \neq 0$, where $[B_{2t}]^i$ denotes the ith column of B_{2t}. Let $\mathcal{I}_{Dt}$, $\mathcal{J}_{Bt}$, $\mathcal{J}_{Dt}$ be defined analogously by collecting the positions of nonzero columns of D_{2t}, B_{3t}, and D_{3t} respectively. Let $\mathcal{I}_t \triangleq \mathcal{I}_{Bt} \cup \mathcal{I}_{Dt}$, $\mathcal{J}_t \triangleq \mathcal{J}_{Bt} \cup \mathcal{J}_{Dt}$. A MLD system (9) is said to be *well posed* if, $\forall t \in \mathbb{Z}$,

(i) $x(t)$ and $u(t)$ satisfy Eq. (9c) for some $\delta(t) \in \{0,1\}^{r_\ell}$, $z(t) \in \mathbb{R}^{r_c}$, and $x_\ell(t+1) \in \{0, 1\}^{n_\ell}$;
(ii) $\forall i \in \mathcal{I}_t$ there exists a mapping $\mathcal{D}_{it}: \mathbb{R}^{n+m} \mapsto \{0,1\}$ such that the ith component $\delta_i(t) = \mathcal{D}_{it}(x(t), u(t))$, and $\forall j \in \mathcal{J}_t$ there exists a mapping $\mathcal{Z}_{jt}: \mathbb{R}^{n+m} \mapsto \mathbb{R}$ such that $z_j(t) = \mathcal{Z}_{jt}(x(t), u(t))$.

A MLD system (9) is said to be *completely well posed* if in addition $\mathcal{I}_t = \{1, \ldots, r_\ell\}$ and $\mathcal{J}_t = \{1, \ldots, r_c\}$, $\forall t \in \mathbb{Z}$.

Note that the functions $\mathcal{D}_{it}$, $\mathcal{Z}_{jt}$ are implicitly defined by the inequalities (9c). Note also that these functions are nonlinear, the nonlinearity being caused by the integer constraint $\delta_i \in \{0, 1\}$.

In the sequel, we shall say that an auxiliary variable $\delta_i(t)$ ($z_j(t)$) is *well posed* if $i \in \mathcal{I}_t$ ($j \in \mathcal{J}_t$), or *indefinite* otherwise.

Hereafter, we shall assume that system (9) is well posed. This property entails that, once $x(t)$ and $u(t)$ are assigned, $x(t+1)$ and $y(t)$ are uniquely defined, and therefore *trajectories* in the x-space and y-space for system (9) can be defined. In particular, we will denote by $x(t, t_0, x_0, u_{t_0}^{t-1})$ the trajectory generated in accordance to Eq. (9) by applying the command inputs $u(t_0)$, $u(t_0+1)$, $\dots$, $u(t-1)$ from initial state $x(t_0)=x_0$. Although typically a model derived from a real system is well posed, a simple numerical test for checking this property is reported in the appendix.

In order to transform propositional logic into linear inequalities, and because of the physical constraints present during plant operation (e.g. saturating actuators, safety conditions, …), we include in the control problem the following constraint:

$$\begin{bmatrix} x \\ u \end{bmatrix} \in \mathscr{C} \triangleq \left\{ \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n+m} : Fx + Gu \le H \right\}. \tag{10}$$

Since typically physical constraints are specified on continuous components, often Eq. (10) can be expressed as the Cartesian product $\mathscr{C} = \mathscr{C}_c \times [0,1]^{n_\ell + m_\ell}$ where $\mathscr{C}_c \triangleq \{[\begin{smallmatrix} x_c \\ u_c \end{smallmatrix}] \in \mathbb{R}^{n_c+m_c} : F_c x_c + G_c u_c \le H_c\}$. Note that the constraint $Fx + Gu \le H$ can be included in (9c).

To express logical facts involving continuous state variables by using the tools presented in Section 2 we will often have to define upper- and lower-bounds as in Eq. (3), therefore from now on we assume that

Assumption 1. $\mathscr{C}$ is a polytope.

Note that assuming that $\mathscr{C}$ is bounded is not restrictive in practice. In fact, continuous inputs and states are often bounded by physical reasons, and logical input/state components are intrinsically bounded. The following developments will be meaningful if, in addition, $\mathscr{C}$ has a nonempty interior.

In the sequel, we shall denote by $\|\cdot\|$ the standard Euclidean norm. Note that for pure logical vectors v, $\|v\|^2$ is a nonnegative integer corresponding to the number of nonzero components of v. The symbol $B(x_0, \delta)$ will denote the ball $\{x: \|x - x_0\| \le \delta\}$.

Observe that the class of MLD systems includes the following important classes of systems:

- Linear hybrid systems.
- Sequential logical systems (Finite State Machines, Automata) ($n_c = m_c = p_c = 0$).
- Nonlinear dynamic systems, where the nonlinearity can be expressed through combinational logic ($n_\ell = 0$).
- Some classes of discrete event systems ($n_c = p_c = 0$).
- Constrained linear systems ($n_\ell = m_\ell = p_\ell = r_\ell = r_c = 0$).
- Linear systems ($n_\ell = m_\ell = p_\ell = r_\ell = r_c = 0$, $E_{it} = 0$, $i = 1,4,5$).

The terms "combinational" and "sequential" are borrowed from digital circuit design jargon. The remaining part of this section is devoted to show in detail examples of systems that can be expressed as MLD systems.

3.1. Piece-wise linear dynamic systems

Consider the following *piece-wise linear time-invariant* (PWLTI) dynamic system

$$x(t+1) = \begin{cases} A_1 x(t) + B_1 u(t) & \text{if } \delta_1(t) = 1, \\ \vdots \\ A_s x(t) + B_s u(t) & \text{if } \delta_s(t) = 1, \end{cases} \tag{11}$$

where $\delta_i(t) \in \{0,1\}$, $\forall i = 1, \dots, s$, are 0–1 variables satisfying the exclusive-or condition

$$\bigoplus_{i=1}^{s} [\delta_i(t) = 1]. \tag{12}$$

System (11) is completely well posed iff $\mathscr{C}$ can be partitioned in s parts $\mathscr{C}_i$ such that

$$\mathscr{C}_i \cap \mathscr{C}_j = \emptyset, \quad \forall i \ne j, \tag{13a}$$

$$\bigcup_{i=1}^{s} \mathscr{C}_i = \mathscr{C} \tag{13b}$$

and δ_i's are defined as

$$[\delta_i = 1] \leftrightarrow \left[\begin{bmatrix} x \\ u \end{bmatrix} \in \mathscr{C}_i \right], \tag{14}$$

A frequent representation of Eq. (11) arises in gain-scheduling, where the linear model (and, consequently, the controller) is switched among a finite set of models, according to changes of the operating conditions. Several nonlinear models can be approximated by a model of the form (11), although this approximation capability is limited for computational reasons by the number s of logical variables.

When the sets $\mathscr{C}_i$ are polytopes of the form $\mathscr{C}_i = \{[\begin{smallmatrix} x \\ u \end{smallmatrix}] : S_i x + R_i u \le T_i\}$, the $\leftarrow$ implication in Eq. (14) corresponds to

$$[\delta_i = 0] \rightarrow \bigvee_{j=1}^{n_i} [S_i^j x + R_i^j u > T_i^j], \tag{15}$$

where S_i^j denotes the jth row of S_i^j. Eq. (15) cannot be easily tackled. However, it is easy to see that Eq. (15) is implied by Eqs. (12) and (13), and therefore can be omitted. In fact, let $[\begin{smallmatrix} x \\ u \end{smallmatrix}] \in \mathscr{C}_i$ and $\delta_i = 0$. Then, by Eq. (12) there exists some $\delta_j = 1$, which implies $[\begin{smallmatrix} x \\ u \end{smallmatrix}] \in \mathscr{C}_j$, a contradiction by Eq. (13a). Eqs. (12)–(14) are therefore equivalent to

$$S_i x(t) + R_i u(t) - T_i \le M_i^*[1 - \delta_i(t)], \tag{16a}$$

$$\sum_{i=1}^{s} \delta_i(t) = 1, \tag{16b}$$

where $M_i^* \triangleq \max_{x \in \mathscr{C}} S_i x(t) + R_i u(t) - T_i$. Eq. (11) can be rewritten as

$$x(t+1) = \sum_{i=1}^{s} [A_i x(t) + B_i u(t)] \delta_i(t). \tag{17}$$

Unfortunately, Eq. (17) is nonlinear, since it involves products between logical variables, states, and inputs. We adopt the procedure (5b) to translate Eq. (17) into equivalent mixed-integer linear inequalities. To this aim, set

$$x(t+1) = \sum_{i=1}^{s} z_i(t), \tag{18}$$

$$z_i(t) \triangleq [A_i x(t) + B_i u(t)]\delta_i(t) \tag{19}$$

and define the vectors $M = [M_1 \ \dots \ M_n]'$, $m = [m_1 \ \dots \ m_n]'$ as

$$M_j \triangleq \max_{i=1,\dots,s} \left\{ \max_{\left[\begin{smallmatrix} x \\ u \end{smallmatrix}\right] \in \mathscr{C}} A_i^j x + B_i^j u \right\}, \tag{20}$$

$$m_j \triangleq \min_{i=1,\dots,s} \left\{ \max_{\left[\begin{smallmatrix} x \\ u \end{smallmatrix}\right] \in \mathscr{C}} A_i^j x + B_i^j.u \right\}. \tag{21}$$

Note that by Assumption 1, M and m are finite, and can be either estimated or exactly computed by solving $2ns$ linear programs. Then, Eq. (19) is equivalent to

$$\begin{aligned} &z_i(t) \le M\delta_i(t), \\ &z_i(t) \ge m\delta_i(t), \\ &z_i(t) \le A_i x(t) + B_i u(t) - m(1-\delta_i(t)), \\ &z_i(t) \ge A_i x(t) + B_i u(t) - M(1-\delta_i(t)). \end{aligned} \tag{22}$$

Therefore, Eqs. (16), (18), and (22) represent Eq. (11) in the form Eq. (9).

For $s > 2$, the number of 0–1 variables can be reduced by setting $h \triangleq \lceil \log_2 s \rceil$ ($\lceil x \rceil$ denoting the smallest integer greater than or equal to x), and

$$i \triangleq \sum_{j=0}^{h-1} 2^j \delta_j(t) \in \{0, \dots, s-1\}. \tag{23}$$

Consider, for instance, $s = 5$ ($h = 3$), and $i = 5 = (101)_2$. Eq. (22) can be replaced by

$$\begin{aligned} &z_5(t) \le M\delta_0(t), \qquad z_5(t) \ge m\delta_0(t), \\ &z_5(t) \le M(1-\delta_1(t)), \qquad z_5(t) \ge m(1-\delta_1(t)), \\ &z_5(t) \le M\delta_2(t), \qquad z_5(t) \ge m\delta_2(t), \\ &z_5(t) \le A_5 x(t) + B_5 u(t) + (M-m) \\ &\qquad \times [(1-\delta_0(t)) + \delta_1(t) + (1-\delta_2(t))], \\ &z_5(t) \ge A_5 x(t) + B_5 u(t) - (M-m) \\ &\qquad \times [(1-\delta_0(t)) + \delta_1(t) + (1-\delta_2(t))]. \end{aligned} \tag{24}$$

The condition $i \le s \le 5$ (i.e. $i \ne 6{,}7$) provides the extra constraint

$$\delta_1(t) + \delta_2(t) \le 1. \tag{25}$$

Note that, although the number of logical variables has been minimized, now δ_j's are no longer constrained by the strong exclusive-or condition (12)–(16b). On the other hand, the number of inequalities has increased from 5×4 in Eq. (22) to 5×8 in Eq. (24). Therefore, the computational benefits arising from adopting (12)–(14) instead of Eqs. (23)–(25) depend on the particular algorithm which is used as a solver. For instance, while enumerative methods take great advantage of a reduction of logical variables, it is not easy to predict the effect on branch and bound algorithms (Williams, 1993).

3.2. Piece-wise linear output functions

In practical applications, it frequently happens that a process can be modeled as a linear dynamic system cascaded by a nonlinear output function $y = h(x)$. When this can be approximated by a piece-wise linear function, by introducing some auxiliary logical variables δ, we obtain the MLD form (9). As an example, consider the following system

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t), \\ y(t) &= \mathrm{sat}(Cx(t)) \end{aligned} \tag{26}$$

along with $x \in \mathscr{X}$, $\mathscr{X}$ bounded, where $\mathrm{sat}(\cdot)$ is the standard saturation function (see Fig. 1)

$$\mathrm{sat}(y) = \begin{cases} -1 & \text{if } y \le -1, \\ y & \text{if } -1 \le y \le 1, \\ 1 & \text{if } y \ge 1. \end{cases} \tag{27}$$

Introduce the following auxiliary logical variables $\delta_1(t)$, $\delta_2(t)$, defined as

$$[Cx > 1] \to [\delta_2 = 1], \tag{28a}$$

$$[Cx < -1] \to [\delta_1 = 1], \tag{28b}$$

$$[Cx < 1] \to [\delta_2 = 0], \tag{28c}$$

$$[Cx > -1] \to [\delta_1 = 0]. \tag{28d}$$

By setting $M \triangleq \max_{x\in\mathscr{X}}\{Cx\}$, $m \triangleq \min_{x\in\mathscr{X}}\{Cx\}$, the logical conditions (28) can be rewritten, respectively, as

$$-Cx + (M-1)\delta_2 \ge -1, \tag{29a}$$

$$Cx - (m+1)\delta_1 \ge -1, \tag{29b}$$

$$Cx + (1-m)(1-\delta_2) \ge 1, \tag{29c}$$

$$Cx - (1+M)(1-\delta_1) \le -1. \tag{29d}$$

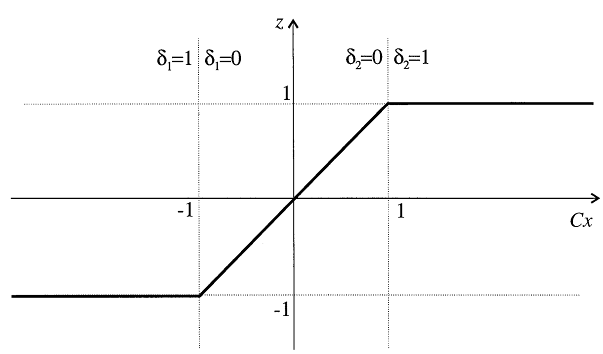


Fig. 1. Saturation function $z = \mathrm{sat}(Cx)$ and role of δ_1, δ_2.

Also, δ_1, δ_2 are related by the logical equations

$$[\delta_1 = 1] \rightarrow [\delta_2 = 0], \tag{30a}$$

$$[\delta_2 = 1] \rightarrow [\delta_1 = 0] \tag{30b}$$

which can be rewritten as

$$\delta_1 - (1 - \delta_2) \leq 0, \tag{31a}$$

$$\delta_2 - (1 - \delta_1) \leq 0. \tag{31b}$$

Introduce the auxiliary variable $z \triangleq \text{sat}(Cx)$. It is clear that

$$[\delta_1 = 0] \rightarrow [z \leq Cx], \tag{32a}$$

$$[\delta_2 = 0] \rightarrow [z \geq Cx], \tag{32b}$$

or, equivalently,

$$z - (M - m)\delta_1 \leq Cx, \tag{33a}$$

$$z + (M - m)\delta_2 \geq Cx \tag{33b}$$

and that

$$z \geq -1, \tag{34a}$$

$$z - (M + 1)(1 - \delta_1) \leq -1, \tag{34b}$$

$$z \leq 1, \tag{34c}$$

$$z + (1 - m)(1 - \delta_2) \geq 1. \tag{34d}$$

It is easy to verify that the above relations correctly define z also in the case $Cx = \pm 1$, which is not explicitly taken into account in (28). In conclusion, the output relation in (26) can be represented by the linear inequalities (29), (31), (33), (34), and consequently (26) belongs to the class of MLD systems (9).

The modeling of non-differentiable functions by using an integer variable for each discontinuity or point of non-differentiability is also discussed by Raman and Grossmann (1991).

3.3. Discrete inputs

Control laws typically provide command inputs ranging on a continuum. However, in applications frequently one has to cope with command inputs which are inherently discrete. Sometimes, the quantization process can be neglected, for instance when the control law is implemented on a digital microprocessor with a sufficiently high number of bits. On the other hand, some applications present intrinsically discrete command variables, such as "on/off" switches, gears or speed selectors, number of individuals or wares, etc. In this case, the quantization error cannot be neglected, since it may lead to very poor performance or even instability. This type of commands can be easily modeled by logical variables. Consider, for instance, the following system:

$$\begin{aligned} &x(t+1) = Ax(t) + Bu(t), \\ &u(t) \in \{u_1, u_2, u_3, u_4\}. \end{aligned} \tag{35}$$

By defining two logical inputs $u_{\ell 1}(t), u_{\ell 2}(t) \in \{0, 1\}$, and an auxiliary variable $z(t)$ such that

$$[u_{\ell 1}(t) = 0,\ u_{\ell 2}(t) = 0] \rightarrow [z(t) = u_1],$$

$$[u_{\ell 1}(t) = 0,\ u_{\ell 2}(t) = 1] \rightarrow [z(t) = u_2],$$

$$[u_{\ell 1}(t) = 1,\ u_{\ell 2}(t) = 0] \rightarrow [z(t) = u_3],$$

$$[u_{\ell 1}(t) = 1,\ u_{\ell 2}(t) = 1] \rightarrow [z(t) = u_4],$$

it follows that Eq. (35) admits the equivalent representation (9)

$$x(t+1) = Ax(t) + Bz(t),$$

$$\begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} z(t) \leq \begin{bmatrix} u_4 - u_1 & u_4 - u_1 \\ 0 & 0 \\ u_4 - u_2 & u_2 - u_4 \\ u_2 - u_1 & u_1 - u_2 \\ u_3 - u_4 & u_4 - u_3 \\ u_1 - u_3 & u_3 - u_1 \\ 0 & 0 \\ u_1 - u_4 & u_1 - u_4 \end{bmatrix} u_\ell(t) + \begin{bmatrix} u_1 \\ -u_1 \\ u_4 \\ -u_1 \\ u_4 \\ -u_1 \\ u_4 \\ u_4 - 2u_1 \end{bmatrix},$$

where $u_\ell(t) \triangleq [u_{\ell 1}(t)\ u_{\ell 2}(t)]'$. In alternative, by defining a four-dimensional logical input $u_\ell(t) \triangleq [u_{\ell 1}(t)\ u_{\ell 2}(t)\ u_{\ell 3}(t)\ u_{\ell 4}(t)]'$, Eq. (35) can be transformed as

$$x(t+1) = Ax(t) + B[u_1 \quad u_2 \quad u_3 \quad u_4]\, u_\ell(t),$$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix} u_\ell(t) + \begin{bmatrix} -1 \\ 1 \end{bmatrix}.$$

3.4. Qualitative outputs

Systems having qualitative outputs can be transformed in the form Eq. (9). Consider, for instance, the following example of a thermal system:

$$x(t+1) = ax(t) + bu(t),$$

$$Y(t) = \begin{cases} \text{COLD} & \text{if } x(t) \leq 5^\circ\text{C}, \\ \text{COOL} & \text{if } 5^\circ\text{C} < x(t) \leq 15^\circ\text{C}, \\ \text{NORMAL} & \text{if } 15^\circ\text{C} < x(t) \leq 35^\circ\text{C}, \\ \text{WARM} & \text{if } 35^\circ\text{C} < x(t) \leq 60^\circ\text{C}, \\ \text{HOT} & \text{if } 60^\circ\text{C} < x(t) \leq 90^\circ\text{C}, \\ \text{TOOHOT} & \text{if } x(t) > 90^\circ\text{C}. \end{cases} \tag{36}$$

Qualitative properties can be conventionally enumerated and associated with an integer y. Here we associate $y = 1$ with $Y =$ "COLD", $y = 2$ with $Y =$ "COOL", … , $y = 6$ with $Y =$ "TOO HOT". Similarly to the procedure adopted to define the saturation function (27), define the following logical variables:

$$[\delta_1(t) = 1] \leftrightarrow [x(t) \le 5],$$
$$[\delta_2(t) = 1] \leftrightarrow [x(t) \le 15],$$
$$[\delta_3(t) = 1] \leftrightarrow [x(t) \le 35],$$
$$[\delta_4(t) = 1] \leftrightarrow [x(t) \le 60],$$
$$[\delta_5(t) = 1] \leftrightarrow [x(t) \le 90],$$

which must satisfy the logical conditions

$$[\delta_1(t) = 1] \to [\delta_2(t) = \delta_3(t) = \delta_4(t) = \delta_5(t) = 1],$$
$$[\delta_2(t) = 1] \to [\delta_3(t) = \delta_4(t) = \delta_5(t) = 1],$$
$$[\delta_3(t) = 1] \to [\delta_4(t) = \delta_5(t) = 1],$$
$$[\delta_4(t) = 1] \to [\delta_5(t) = 1].$$

By using Eq. (4e), these logical conditions can be rewritten in the form (9c). Then, $y(t) \triangleq 1\delta_1(t) + 2(\delta_2(t) - \delta_1(t)) + 3(\delta_3(t) - \delta_2(t)) + 4(\delta_4(t) - \delta_3(t)) + 5(\delta_5(t) - \delta_4(t)) + 6(1 - \delta_5(t))$, which represents an equivalent output of the system which can take only six different values, and has the form (9b). This type of modeling is useful to include heuristics and rules of thumb in optimal control problems, as detailed later in Section 5.

3.5. Bilinear systems

Consider the class of nonlinear systems of the form

$$x(t+1) = Ax(t) + Bu(t) + \sum_{i=1}^{m} u_i(t)\, C_i x(t), \quad x \in \mathbb{R}^n,\ u \in \mathbb{R}^m. \tag{37}$$

If we assume that the input $u(t)$ is quantized, these can be transformed into MLD system (9). For the sake of simplicity, consider $m = 1$, and let

$$u(t) = D\delta(t), \qquad D \triangleq u_0[2^0 \quad \ldots \quad 2^{r-1}], \quad \delta(t) \in \{0,1\}^r \tag{38}$$

similarly to Eq. (23). Then, $x(t+1) = Ax(t) + BD\delta(t) + \delta'(t)D'C_1x(t)$. By introducing the auxiliary continuous vector $z(t) = \delta'(t)D'C_1x(t)$ and recalling (5b) the bilinear system (37)–(38) can be rewritten in the form (9).

3.6. Finite state machines (automata)

We consider here finite-state machines whose events are generated by an underlying LTI dynamic system. A typical and important example of systems which can be modeled within this framework are real-time systems, where physical processes are controlled by embedded digital controllers. Consider, for instance, the simple automaton and linear system depicted in Fig. 2, and described by the relations

$$\begin{aligned} &[x_\ell(t) = 0] \wedge [x_c \le 0] \to [x_\ell(t+1) = 0],\\ &[x_\ell(t) = 0] \wedge [x_c > 0] \to [x_\ell(t+1) = 1],\\ &[x_\ell(t) = 1] \to [x_\ell(t+1) = 0],\\ &x_c(t+1) = ax_c(t) + bu(t). \end{aligned} \tag{39}$$

The (0–1) finite-state $x_\ell(t)$ remains in 0 as long as the continuous state $x_c(t)$ is non-positive. If $x_c(t) > 0$ at some t, then x_c generates a digital impulse, i.e. $x_\ell(t+1) = 1$, $x_\ell(t+2) = 0$. The automaton's dynamics is hence driven by events generated by the underlying linear system. Let $x \triangleq [x_c' \ x_\ell']'$, and introduce the auxiliary logical variables $\delta_1(t)$, $\delta_2(t)$ defined as

$$[\delta_1(t) = 1] \leftrightarrow [x_c(t) \le 0], \tag{40a}$$

$$[\delta_2(t) = 1] \leftrightarrow [x_\ell(t) = 0] \wedge [\delta_1(t) = 0]. \tag{40b}$$

By Eq. (4e), Eq. (40a) can be rewritten as

$$x_c(t) \le M(1 - \delta_1(t)), \tag{41a}$$

$$x_c(t) \ge \varepsilon + (m - \varepsilon)\delta_1(t), \tag{41b}$$

where $\varepsilon > 0$ is a small tolerance (machine precision), and by Eq. (5a),

$$\delta_2(t) \le (1 - \delta_1(t)), \tag{42a}$$

$$\delta_2(t) \le (1 - x_\ell(t)), \tag{42b}$$

$$\delta_2(t) \ge (1 - \delta_1(t)) + (1 - x_\ell(t)) - 1. \tag{42c}$$

The mixed-integer linear inequalities (41)–(42) along with the equality $x_\ell(t+1) = \delta_2(t)$ define the automaton part in system (39), which hence is a MLD system. As a further example, Branicky et al. (1998) describe how to associate a finite automaton similar to the one depicted in Fig. 2 with hysteresis phenomena which frequently occur in different contexts (e.g. magnetic, electrical, etc.). Finally, time dependence can be emulated in the time-invariant MLD

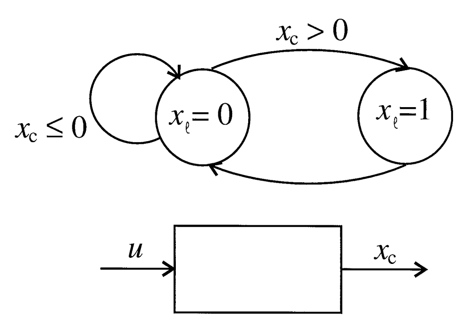


Fig. 2. Automaton driven by conditions on an underlying dynamic system.

framework by modeling time as the output of a digital clock, which is a finite-state machine in free evolution.

4. Stability of MLD systems

Since we treat systems having both real and logical states evolving within a bounded set $\mathscr{C}$, we adapt here standard definitions of stability (see e.g. (Keerthi and Gilbert, 1988) to MLD systems.

Definition 2. A vector $x_e \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$ is said to be an *equilibrium state* for (9) and input $u_e \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ if $[x_e' \ u_e']' \in \mathscr{C}$ and $x(t, t_0, x_e, u_e) = x_e$, $\forall t \geq t_0$, $\forall t_0 \in \mathbb{Z}$. The pair (x_e, u_e) is said to be an *equilibrium pair*.

Definition 3. Given an equilibrium pair (x_e, u_e), $x_e \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$ is said to be *stable* if, given $t_0 \in \mathbb{Z}$, $\forall \varepsilon > 0 \ \exists \delta(\varepsilon, t_0)$ such that $\|x_0 - x_e\| \leq \delta \Rightarrow \|x(t, t_0, x_0, u_e) - x_e\| \leq \varepsilon$, $\forall t \geq t_0$.

Definition 4. Given an equilibrium pair (x_e, u_e), $x_e \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$ is said to be *asymptotically stable* if x_e is stable and $\exists r > 0$ such that $\forall x_0 \in B(x_e, r)$ and $\forall \varepsilon > 0 \ \exists T(\varepsilon, t_0)$ such that $\|x(t, t_0, x_0, u_e) - x_e\| \leq \varepsilon$, $\forall t \geq T$.

Definition 5. Given an equilibrium pair (x_e, u_e), $x_e \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$ is said to be *exponentially stable* if x_e is asymptotically stable and in addition $\exists \delta > 0$, $\alpha > 0$, $0 \leq \beta < 1$ such that $\forall x_0 \in B(x_e, \delta)$ and $\|x(t, t_0, x_0, u_e) - x_e\| \leq \alpha \beta^{t - t_0} \|x_0 - x_e\|$.

Note that asymptotic convergence of the logical component $x_\ell(t)$ to $x_{\ell e}$ is equivalent to the existence of a finite time t_e such that $x_\ell(t) \equiv x_{\ell e}$, $\forall t \geq t_e$ (Passino et al., 1994). Consequently, local stability properties could be restated for the continuous part x_c only, by setting $x_\ell = x_{\ell e}$. Note also that there exists a set around the continuous part x_{ce} of the equilibrium state x_e such that, by perturbing $x_c(t)$ within that set, the equations of motion are again satisfied for $x_\ell(t) = x_{\ell e}$.

For an equilibrium pair (x_e, u_e), in the time-invariant case a corresponding equilibrium value can be established for well-posed components of auxiliary variables via the functions $\mathscr{D}_i$, $\mathscr{Z}_j$ introduced earlier. In addition, for indefinite components we relax the concept of "equilibrium" through the following definition

Definition 6. Let (x_e, u_e) be an equilibrium pair for a MLD system, and let the system be well posed. Assume that $\mathscr{I} \triangleq \lim_{t \to \infty} \mathscr{I}_t$ and $\mathscr{J} \triangleq \lim_{t \to \infty} \mathscr{J}_t$ exist. For $i \in \mathscr{I}$, $j \in \mathscr{J}$, let $\delta_{e,i}$, $z_{e,j}$ the corresponding equilibrium auxiliary variables. An auxiliary vector δ (or z) is said to be *definitely admissible* if $\delta_i = \delta_{e,i}$, $\forall i \in \mathscr{I}$, ($z_j = z_{e,j}$, $\forall j \in \mathscr{J}$), and $\exists t_e$ such that

$$E_{2t}\delta + E_{3t} z \leq E_{1t} u_e + E_{4t} x_e + E_{5t}, \ \forall t \geq t_e. \tag{43}$$

Note that for time-invariant MLD systems, $\mathscr{I} \equiv \mathscr{I}_t$, $\mathscr{J} \equiv \mathscr{J}_t$, $\forall t \in \mathbb{Z}$, and Eq. (43) reduces to only one set of linear inequalities.

Example 4.1. Consider the following system

$$\begin{aligned}
x(t+1) &= 0.8 \begin{bmatrix} \cos\alpha(t) & -\sin\alpha(t) \\ \sin\alpha(t) & \cos\alpha(t) \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \\
y(t) &= [1 \quad 0] x(t), \\
\alpha(t) &= \begin{cases} \frac{\pi}{3} & \text{if } [1 \quad 0]\, x(t) \geq 0, \\ -\frac{\pi}{3} & \text{if } [1 \quad 0]\, x(t) < 0, \end{cases} \\
x(t) &\in [-10, 10] \times [-10, 10], \\
u(t) &\in [-1, 1].
\end{aligned} \tag{44}$$

According to Eq. (22), by using auxiliary variables $z(t) \in \mathbb{R}^4$ and $\delta(t) \in \{0,1\}$ such that $[\delta(t) = 1] \leftrightarrow [[1 \quad 0] x(t) \geq 0]$, Eq. (44) can be rewritten in the form (9) as

$$x(t+1) = [I \quad I] z(t)$$

$$\begin{bmatrix} 10 \\ -10-\varepsilon \\ -M \\ -M \\ M \\ M \\ M \\ M \\ -M \\ -M \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ I & 0 \\ -I & 0 \\ 0 & I \\ 0 & -I \\ I & 0 \\ M & 0 \\ -I & I \\ 0 & -I \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} z(t) \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ B \\ -B \\ B \\ -B \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} u(t) + \begin{bmatrix} 1 \ \ 0 \\ -1 \ \ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ A_1 \\ -A_1 \\ A_2 \\ -A_2 \\ I \\ -I \\ 0 \\ 0 \end{bmatrix} x(t) + \begin{bmatrix} 10 \\ -\varepsilon \\ 0 \\ M \\ M \\ M \\ M \\ M \\ 0 \\ 0 \\ N \\ N \\ 1 \\ 1 \end{bmatrix},$$

where $B = [0 \ 1]'$, A_1, A_2 are obtained by Eq. (44) by setting respectively $\alpha = \frac{\pi}{3}$, $-\frac{\pi}{3}$, $M = 4(1 + \sqrt{3})$ $[1 \ 1]' + B$, $N \triangleq 10[1 \ 1]'$, and ε is a properly small positive scalar. The evolution starting from $x(0) = [1 \ 1]'$ for $u(t) \equiv 0$, $\forall t \geq 0$, is depicted in Fig. 3a. It is easy to prove that the origin $[0 \ 0]'$ is exponentially stable and has the

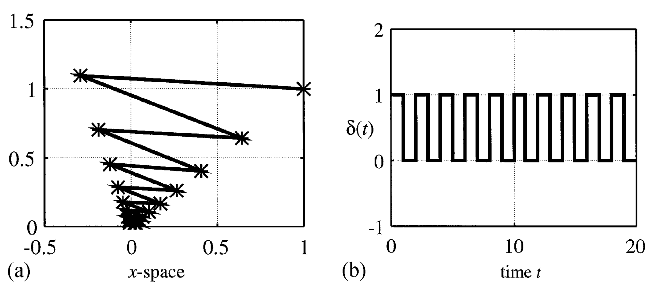


Fig. 3. Evolution of system (44). (a) State $x(t)$. (b) Logical variable $\delta(t)$.

whole set $\mathscr{C}$ as domain of attraction. However, if one defines a new system having state $rx(t) = [x'(t)\ \delta(t-1)]'$, neither $[0\ 0\ 0]'$nor $[0\ 0\ 1]'$ are equilibria, since $\delta(t)$, as shown in Fig. 3b, keeps oscillating as t proceeds.

5. Optimal control of MLD systems

In the previous sections we have presented a modeling framework for systems described by dynamics, logic, and constraints. In the following sections we motivate this choice by providing tools for synthesizing controllers for such a class of systems. To this aim, for a MLD system of the form Eq. (9), consider the following problem:

Problem 1. *Given an initial state x_0 and a final time T, find (if it exists) the control sequence $u_0^{T-1} \triangleq \{u(0), u(1), \ldots, u(T-1)\}$ which transfers the state from x_0 to x_f and minimizes the performance index*

$$J(u_0^{T-1}, x_0) \triangleq \sum_{t=0}^{T-1} \|u(t) - u_f\|^2_{Q_1} + \|\delta(t, x_0, u_0^t) - \delta_f\|^2_{Q_2} + \|z(t, x_0, u_0^t) - z_f\|^2_{Q_3} + \|x(t, x_0, u_0^{t-1}) - x_f\|^2_{Q_4} + \|y(t, x_0, u_0^{t-1}) - y_f\|^2_{Q_5} \tag{45}$$

subject to

$$x(T, x_0, u_0^{T-1}) = x_f \tag{46}$$

and the MLD system dynamics Eq. (9), where $\|x\|^2_Q \triangleq x'Qx$, $Q_i = Q'_i \geq 0$, $i = 1, \ldots, 5$, are given weight matrices, and x_f, u_f, δ_f, z_f, y_f are given offset vectors satisfying Eqs. (9b)–(9c).

Note that if $\delta_f = 0$ and Q_2 is diagonal, the second quadratic term is equivalent to the linear term $\sum_{i=0}^{r_\ell} Q_{ii}\delta_i(t, x_0, u_0^t)$.

Problem 1 can be solved as a *mixed-integer quadratic programming* (MIQP) problem. In fact, let $x(t)$ be a compact notation for $x(t, x_0, u_0^{t-1})$, the same convention being used for $\delta(t)$, $z(t)$. From Eq. (9a), for time-invariant systems we have the solution formula

$$x(t) = A^t x_0 + \sum_{i=0}^{t-1} A^i [B_1 u(t-1-i) + B_2 \delta(t-1-i) + B_3 z(t-1-i)], \tag{47}$$

where the relation between $x(t)$ and x_0, u_0^{t-1} is only apparently linear, because $\delta(i)$, $z(i)$ hide a nonlinear dependence on x_0 and u_0^{t-1}, as observed earlier. By plugging Eq. (47) into Eqs. (9c) and (45), and by defining the vectors

$$\Omega \triangleq \begin{bmatrix} u(0) \\ \vdots \\ u(T-1) \end{bmatrix}, \quad \Delta \triangleq \begin{bmatrix} \delta(0) \\ \vdots \\ \delta(T-1) \end{bmatrix}, \quad \Xi \triangleq \begin{bmatrix} z(0) \\ \vdots \\ z(T-1) \end{bmatrix}, \quad \mathscr{V} \triangleq \begin{bmatrix} \Omega \\ \Delta \\ \Xi \end{bmatrix},$$

we obtain the following equivalent formulation:

$$\begin{aligned} \min_{\mathscr{V}} \quad & \mathscr{V}'S_1\mathscr{V} + 2(S_2 + x_0'S_3)\mathscr{V} \\ \text{s.t.} \quad & F_1\mathscr{V} \leq F_2 + F_3 x_0, \end{aligned} \tag{48}$$

where matrices S_i, F_i, $i = 1, 2, 3$, are suitably defined. Then, existence, uniqueness, and continuity with respect to x_0 of the optimal control sequence can be investigated as feasibility, uniqueness, and continuity with respect to parameters of the solution of the MIQP problem (48).

Example 5.1. Consider again the MDL system of Example 4.1. In order to optimally transfer the state from $x_0 = [-1\ 1]'$ to $x_f = [0\ 0]'$, the performance index (45) is minimized subject to Eq. (46) and the MLD system dynamics Eq. (44), along with the weights $Q_1 = 1$, $Q_2 = 0.01$, $Q_3 = 0.01 I_4$, $Q_4 = I_2$, $Q_5 = 0$, and $z_f = [0\ 0\ 0\ 0]'$, $\delta_f = 1$, $u_f = 0$. The resulting optimal trajectories are shown in Fig. 4. Fig. 5 shows the effect of varying the ratio between weights, in particular the input weight Q_1 takes the values $10^{-6}, 0.1, 10$.

5.1. Soft constraints and constraint priority

In practical applications, it is common use to distinguish between *hard constraints*, which cannot be violated (for instance motor voltage limits), and *soft constraints*, whose violation is allowed (e.g. bounds on temperatures), even if penalized. When dealing with soft constraints, one

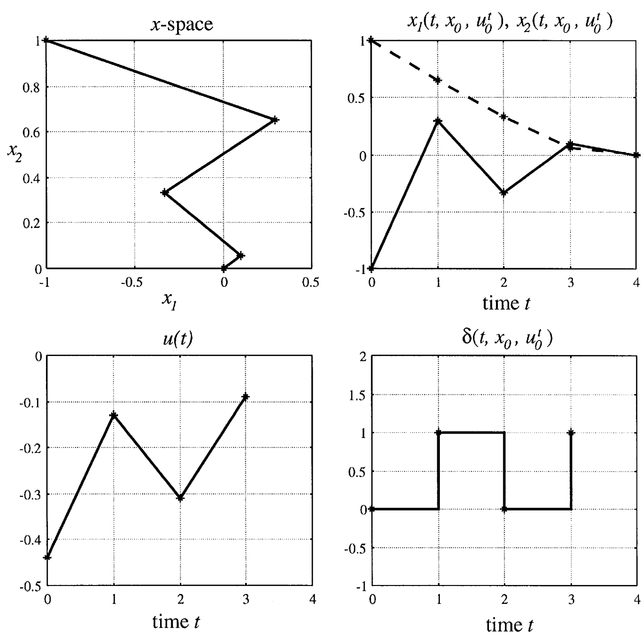


Fig. 4. Optimal control of system (44).

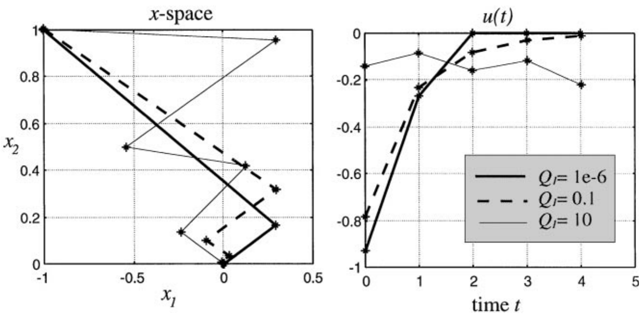


Fig. 5. Optimal control of system (44). Comparison between different relative weights ($Q_2 = 0.01$, $Q_3 = 0.01 I_4$, $Q_4 = I_2$).

would like to fulfill two requirements. First, if the set of feasible solutions to the hard constrained problem is nonempty, the minimizer should belong to that set. Second, if such a set is empty, one should be able to decide a trade-off between the cost and constraint violation.

Consider the following optimization problem:

$$\begin{aligned} &\min_{x \in X} \quad x'Sx \\ &\text{s.to.} \quad Ax \le B, \end{aligned} \tag{49}$$

where the set X is a bounded polyhedron. The simplest way to soften constraints is to modify Eq. (49) in the following problem

$$\begin{aligned} &\min_{x \in X, \varepsilon \ge 0} \quad x'Sx + \varepsilon' M_1 \varepsilon \\ &\text{s.t.} \quad Ax - C\varepsilon \le B, \end{aligned} \tag{50}$$

where $\varepsilon \in \mathbb{R}^s$ is a vector of slack variables, C is a vector whose components are 0 or 1 according if the corresponding constraint is hard or soft, and M_1 is a (large) penalty weight matrix. The problem with the formulation Eq. (50) is that the first requirement is not guaranteed. As an alternative, define a logical variable δ such that $[\delta = 0] \leftrightarrow [\varepsilon_i = 0, \forall i = 1, \ldots, s]$, and minimize $x'Sx + \varepsilon' M_1 \varepsilon + M_2 \delta$ with respect to x, ε, and δ, where $M_2 > \max_{x \in X} x'Sx$, and M_1 decides the trade-off between cost and constraint violation, when no feasible solution exists to the hard-constrained problem.

Constraint violation can also be considered at r levels of priority, by introducing r 0–1 variables δ_i, $i = 1, \ldots, r$, by letting

$$\begin{aligned} &[\delta_1 = 0] \leftrightarrow [\varepsilon_1 = \cdots = \varepsilon_{i_1} = 0], \\ &[\delta_2 = 0] \leftrightarrow [\delta_1 = 0] \wedge [\varepsilon_{i_1+1} = \cdots = \varepsilon_{i_2} = 0] \\ &\vdots \qquad\qquad \vdots \\ &[\delta_r = 0] \leftrightarrow [\delta_1 = \cdots = \delta_{r-1} = 0] \\ &\qquad\qquad \wedge [\varepsilon_{i_{r-1}+1} = \cdots = \varepsilon_{i_r} = 0], \end{aligned}$$

and by minimizing $x'Sx + \varepsilon' M_1 \varepsilon + M_2 \sum_{i=1}^{r} \delta_i$.

Note that soft constraints and constraint priority can be directly considered in the MLD structure (9). In fact, constraints in Eq. (9c) can be softened and/or prioritized by incorporating the slack vector $\varepsilon(t)$ in the z-vector, and the auxiliary logical variables $\delta_1(t), \ldots, \delta_r(t)$ in the δ-vector.

5.2. Integrating heuristics, logic, and dynamics

As shown by Raman and Grossmann (1991, 1992) for process synthesis, logic and heuristics can be integrated through propositional logic. This type of qualitative knowledge is useful for two purposes. First, in many cases solutions which reflect the operator's experience are simply preferred. Second, it may help to expedite the search for feasible solutions, for instance by generating a base case. On the other hand, qualitative knowledge are typically just rules of thumb which may not always hold, lead to solutions which are far away from optimality, and even be contradictory.

Heuristic rules can be expressed as "soft" logic facts, by considering instead of the clause D which expresses the rule the following

$$D \vee V. \tag{51}$$

Since the clause is also a disjunction, the conversion of Eq. (51) into linear inequalities is straightforward, for instance $[\sim P_1 \vee P_2] \vee V$ yields

$$1 - \delta_1 + \delta_2 + v \ge 1, \tag{52}$$

where v can also be interpreted as a slack variable that allows the violation of the inequality. Since Eq. (52) only involves 0–1 variables, in this case the variable v can be treated as a continuous nonnegative variable, despite the fact that it will take only 0-1 values. As a further example, consider $[T \ge T_{\mathrm{HOT}}] \rightarrow [\delta_1 = 1] \vee [v = 1]$, which is equivalent to $T - T_{\mathrm{HOT}} - M(\delta_1 + v) \le 0$, where M is a known upper bound on $T - T_{\mathrm{HOT}}$ and v is a binary variable that represents the violation of the heuristics (Raman and Grossmann, 1992).

When the fulfillment of heuristic rules is impossible or destroys optimality, one should violate the weaker (more uncertain) set of rules. A discrimination between weak and strong rules can be obtained by penalizing with different weights w_i the violation variables v_i. The penalty w_i is a nonnegative number expressing the uncertainty of the corresponding logical expression. The more uncertain the rule according to the designer's experience, the lower the penalty for its violation.

For the optimal control problem at hand, one can add the linear term $w'v$ in Eq. (45) and minimize with respect to $\mathcal{V}$ and v. As an alternative, if the performance index should not be mixed with heuristics violation penalties, one can first find the vector v^* which minimizes $w'v$ subject to linear constraints involving $\mathcal{V}$, and v (a *mixed integer linear problem* (MILP)), set $v = v^*$, and then minimize Eq. (45) with respect to $\mathscr{V}$ only. This corresponds to a preprocessing of the given set of logical, dynamical, and heuristic conditions in order to obtain the feasible set which better takes into account qualitative knowledge.

6. Predictive control of MLD systems

As observed in the previous sections, a large quantity of situations can be modeled through the MLD structure. Then, it is interesting from both a theoretical and practical point of view to ask whether or not a MLD system can be stabilized to an equilibrium state or can track a desired reference trajectory, possibly via feedback control. Finding such a control law is not an easy task, the system being neither linear nor even smooth. In this section, we show how predictive control provides successful tools to perform this task. For the sake of notational simplicity, the index t will be dropped from Eq. (9), by assuming that the system is time-invariant.

As mentioned in Section 1, the main idea of predictive control is to use a model of the plant to *predict* the future evolution of the system. Based on this prediction, at each time step t the controller selects a sequence of future command inputs through an on-line optimization procedure, which aims at maximizing the tracking performance, and enforces fulfillment of the constraints. Only the first sample of the optimal sequence is actually applied to the plant at time t. At time $t+1$, a new sequence is evaluated to replace the previous one. This on-line "re-planning" provides the desired feedback control feature.

Consider an equilibrium pair (x_e, u_e) and let (δ_e, z_e) be definitely admissible in the sense of Definition 6. Let the components $\delta_{e,i}$, $z_{e,j}$, $i \notin \mathscr{I}$, $j \notin \mathscr{J}$, correspond to desired steady-state values for the indefinite auxiliary variables. Let t be the current time, and $x(t)$ the current state. Consider the following optimal control problem

$$\min_{\{v_0^{T-1}\}} J(v_0^{T-1}, x(t)) \triangleq \sum_{k=0}^{T-1} \|v(k) - u_e\|_{Q_1}^2 + \|\delta(k|t) - \delta_e\|_{Q_2}^2 + \|z(k|t) - z_e\|_{Q_3}^2 + \|x(k|t) - x_e\|_{Q_4}^2 + \|y(k|t) - y_e\|_{Q_5}^2 \tag{53}$$

$$\text{s.t.} \begin{cases} x(T|t) = x_e, \\ x(k+1|t) = Ax(k|t) + B_1 v(k) + B_2\delta(k|t) + B_3 z(k|t), \\ y(k|t) = Cx(k|t) + D_1 v(k) + D_2\delta(k|t) + D_3 z(k|t), \\ E_2\delta(k|t) + E_3 z(k|t) \le E_1 v(k) + E_4 x(k|t) + E_5, \end{cases} \tag{54}$$

where $Q_1 = Q'_1 > 0$, $Q_2 = Q'_2 \ge 0$, $Q_3 = Q'_3 \ge 0$, $Q_4 = Q'_4 > 0$, $Q_5 = Q'_5 \ge 0$, $x(k|t) \triangleq x(t+k, x(t), v_0^{k-1})$, and $\delta(k|t)$, $z(k|t)$, $y(k|t)$ are similarly defined. Assume for the moment that the optimal solution $\{v_t^*(k)\}_{k=0,\dots,T-1}$ exists. According to the *receding horizon* philosophy mentioned above, set

$$u(t) = v_t^*(0), \tag{55}$$

disregard the subsequent optimal inputs $v_t^*(1), \dots, v_t^*(T-1)$, and repeat the whole optimization procedure at time $t+1$. The control law (53)–(55) will be referred to as the *mixed integer predictive control* (MIPC) law. Note that once x_e, u_e have been fixed, consistent steady-state vectors δ_e, z_e can be obtained by choosing feasible points in the domain described by Eq. (9c), for instance by solving a MILP (see also Section 6.1).

Several formulations of predictive controllers for MLD systems might be proposed. For instance, the number of control degrees of freedom can be reduced to $N_u < T$, by setting $u(k) \equiv u(N_u - 1)$, $\forall k = N_u, \dots, T$. However, while in other contexts this amounts to hugely down-sizing the optimization problem at the price of a reduced performance, here the computational gain is only partial, since all the T $\delta(k|t)$ and $z(k|t)$ variables remain in the optimization. Infinite horizon formulations are inappropriate for both practical and theoretical reasons. In fact, approximating the infinite horizon with a large T is computationally prohibitive, as the number of 0–1 variables involved in the MIQP depends linearly on T. Moreover, the quadratic term in δ might oscillate, as exemplified in Example 1, Fig. 3b, and hence "good" (i.e. asymptotically stabilizing) input sequences might be ruled out by a corresponding infinite value of the performance index; it could even happen that no input sequence has finite cost.

Theorem 1. *Let (x_e, u_e) be an equilibrium pair and (δ_e, z_e) definitely admissible. Assume that the initial state $x(0)$ is such that a feasible solution of problem* (53) *exists at time $t = 0$. Then $\forall Q_1 = Q'_1 > 0$, $Q_2 = Q'_2 \ge 0$, $Q_3 = Q'_3 \ge 0$, $Q_4 = Q'_4 > 0$, and $Q_5 = Q'_5 \ge 0$ the MIPC law* (53)–(55) *stabilizes the system in that*

$$\lim_{t\to\infty} x(t) = x_e,$$

$$\lim_{t\to\infty} u(t) = u_e,$$

$$\lim_{t\to\infty} \|\delta(t) - \delta_e\|_{Q_2} = 0,$$

$$\lim_{t\to\infty} \|z(t) - z_e\|_{Q_3} = 0,$$

$$\lim_{t\to\infty} \|y(t) - y_e\|_{Q_5} = 0,$$

while fulfilling the dynamic/relational constraints (9c).

Note that if $Q_2 > 0$ (or $Q_3 > 0$, $Q_5 > 0$), convergence of $\delta(t)$ (or $z(t)$, $y(t)$) follows as well.

Proof. The proof easily follows from standard Lyapunov arguments. Let $\mathscr{U}_t^*$ denote the optimal control sequence

$\{v_t^*(0), \ldots, v_t^*(T-1)\}$, let

$$V(t) \triangleq J(\mathcal{U}_t^*, x(t))$$

denote the corresponding value attained by the performance index, and let $\mathcal{U}_1$ be the sequence $\{v_t^*(1), \ldots, v_t^*(T-2), u_e\}$. Then, $\mathcal{U}_1$ is feasible at time $t+1$, along with the vectors $\delta(k|t+1) = \delta(k+1|t)$, $z(k|t+1) = z(k+1|t), k = 0, \ldots, T-2, \delta(T-1|t+1) = \delta_e, z(T-1|t+1) = z_e$, being $x(T-1|t+1) = x(T|t) = x_e$ and (δ_e, z_e) definitely admissible. Hence,

$$\begin{aligned} V(t+1) \le J(\mathcal{U}_1, x(t+1)) &= V(t) - \|x(t) - x_e\|_{Q_4} - \|u(t) \\ &- u_e\|_{Q_1} - \|\delta(t) - \delta_e\|_{Q_2} \\ &- \|z(t) - z_e\|_{Q_3} - \|y(t) - y_e\|_{Q_5} \end{aligned} \tag{56}$$

and $V(t)$ is decreasing. Since $V(t)$ is lower-bounded by 0, there exists $V_\infty = \lim_{t\to\infty} V(t)$, which implies $V(t+1) - V(t) \to 0$. Therefore, each term of the sum

$$\begin{aligned} &\|x(t) - x_e\|_{Q_4} + \|u(t) - u_e\|_{Q_1} + \|\delta(t) - \delta_e\|_{Q_2} \\ &+ \|z(t) - z_e\|_{Q_3} + \|y(t) - y_e\|_{Q_5} \le V(t) - V(t+1) \end{aligned}$$

converges to zero as well, which proves the theorem. □

Remark 1. Despite the fact that very effective methods exist to compute the (global) optimal solution of the MIQP problem (53)–(55) (see Section 7 below), in the worst case the solution time depends exponentially on the number of integer variables. In principle, this might limit the scope of application of the proposed method to very slow systems, since for real-time implementation the sampling time should be large enough to allow the worst-case computation. However, the proof of Theorem Table does not require that the evaluated control sequences $\{\mathcal{U}_t^*\}_{t=0}^{\infty}$ are global optima. In fact, Eq. (56) just requires that

$$J(\mathcal{U}_{t+1}^*, x(t+1)) \le J(\mathcal{U}_1, x(t+1)). \tag{57}$$

The sequence $\mathcal{U}_1$ is available from the previous computation (performed at time t), and can be used to initialize the MIQP solver at time $t+1$. The solver can then be interrupted at any intermediate step to obtain a suboptimal solution $\mathcal{U}_{t+1}^*$ which satisfies (57). For instance, when Branch and Bound methods are used to solve the MIQP problem, the new control sequence $\mathcal{U}_t^*$ can be selected as the solution to a QP subproblem which is integer-feasible and has the lowest value. Obviously in this case tracking performance deteriorates.

Remark 2. Since in general the implicitly defined functions $\mathcal{D}_i$, $\mathcal{Z}_j$ are not continuous, convergence of the well-posed components of δ, z cannot be inferred by convergence of x and u. For instance, a variable δ defined as $[\delta = 1] \leftrightarrow [x > 0]$ has a corresponding $\mathcal{D}$ function which is discontinuous in $x = 0$.

Remark 3. Nothing can be inferred about the asymptotic behavior of the indefinite components of δ, z, unless Q_2, $Q_3 > 0$. However, the behavior of unweighted indefinite variables are clearly of little interest.

Remark 4. The stability result proved in Theorem 1 is not affected by the presence of positive linear terms in Eq. (53). For instance, if $z \in \mathbb{R}$, $z_e = 0$ and the constraint $z(t) \ge 0$ is present, a term of the form $q_3 z$, $q_3 \ge 0$ can be included in Eq. (53). Hence, soft constraints or heuristic rules can be taken into account by modifying the performance index Eq. (53) as detailed in Section 5, without corrupting the warranty of stability.

Remark 5. Note that because of its receding horizon mechanism, MIPC is a closed-loop approach, and is clearly more robust than pure open-loop optimal control. On the other hand, MIPC control can be also adopted for off-line computation of open-loop input trajectories. Let N be the duration in time steps of the batch operation to be designed. Since short horizons T can be implemented within MIPC, this would require the solution of N MIQP problems of size T. On the other hand, pure optimal control would require the solution of one MIQP problem of size N. Assuming a worst case exponential dependence on the size of the problem, the first would have a complexity of $N2^T$, while the second of 2^N. For $N = 100$, $T = 5$ this is equivalent to 3200 versus about 10^{30}. This gain in computational efficiency, however, may be paid at the price of a deteriorated performance, due to the gap between the open-loop performance objective minimized at each step and the actual

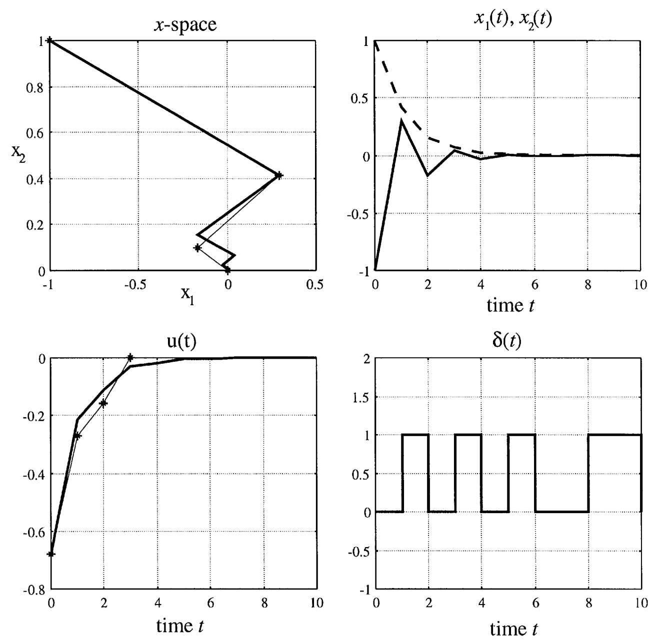


Fig. 6. Closed-loop regulation problem for system (44). Closed-loop trajectories (thick lines) and optimal solution at $t = 0$ (thin lines, right plots).

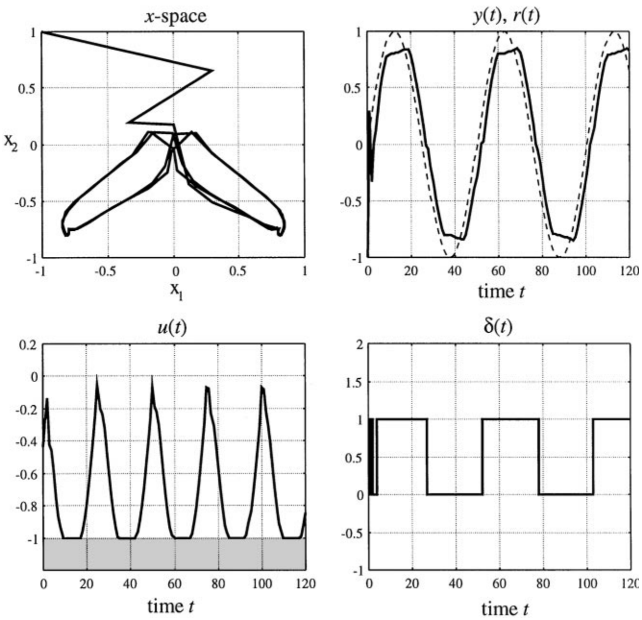


Fig. 7. Closed-loop tracking problem for system (44), with $y(t) = x_1(t)$.

performance. Note that this gap increases as the prediction horizon T gets shorter.

Example 6.1. Consider again the MDL system of Examples 4.1 and 5.1. In order to stabilize the system to the origin, the feedback control law (53)–(55) is adopted, along with the parameters $T = 3$, $u_e = 0$, $\delta_e = 0$, $z_e = [0\ 0\ 0\ 0]'$, $x_e = [0\ 0]'$, $y_e = 0$, and the same weights of Example 5.1 (Fig. 4). Fig. 6 shows the resulting trajectories. The trajectories obtained at time $t = 0$ by solving the optimal control problem (53)–(54) are also reported in the right plots (thin lines). Consider now a desired reference $r(t) = \sin(t/8)$ for the output $y(t)$. We apply the same MIPC controller, with the exception of $Q_4 = 10^{-8} I_2$, $Q_5 = 1$. The steady-state parameters are selected as $y_e = r(t)$, and u_e, x_e, δ_e, z_e consistently (see Section 6.1 below). Fig. 7 shows the resulting closed-loop trajectories. Notice that the constraint $-1 \le u(t) \le 1$ prevents the system from tracking the peaks of the sinusoid, and therefore the output trajectory is chopped.

6.1. Tracking problems

For tracking problems, the goal is that the output $y(t)$ follows a reference trajectory $r(t)$. For each time step t, the values for $y_e, x_e, u_e, \delta_e, z_e$ in Eq. (53) corresponding to $r(t)$ can be computed by solving the following MIQP problem

$$\min_{\{x_e, u_e, \delta_e, z_e\}} \|y_e - r(t)\|^2_{Q_5} + \rho(\|x_e\|^2 + \|u_e\|^2 + \|\delta_e\|^2 + \|z_e\|^2) \tag{58a}$$

$$\text{s.t.} \begin{cases} x_e = A x_e + B_1 u_e + B_2 \delta_e + B_3 z_e, \\ E_2 \delta_e + E_3 z_e \le E_1 u_e + E_4 x_e + E_5, \end{cases} \tag{58b}$$

where $y_e = C x_e + D_1 u_e + D_2 \delta_e + D_3 z_e$. The parameter $\rho > 0$ is any (small) positive number, and is needed to ensure strict convexity of the value function (58a). This procedure allows to define a set-point y_e which is as close as possible to $r(t)$, compatibly with the constraints.

7. MIQP solvers

With the exception of particular structures, mixed-integer programming problems involving 0–1 variables are classified as $\mathcal{NP}$-complete, which means that in the worst case, the solution time grows exponentially with the problem size (Raman and Grossmann, 1991). Despite this combinatorial nature, several algorithmic approaches have been proposed and applied successfully to medium and large size application problems (Floudas, 1995), the four major ones being

- *Cutting plane methods*, where new constraints (or "cuts") are generated and added to reduce the feasible domain until a 0–1 optimal solution is found.
- *Decomposition methods*, where the mathematical structure of the models is exploited via variable partitioning, duality, and relaxation methods.
- *Logic-based methods*, where disjunctive constraints or symbolic inference techniques are utilized which can be expressed in terms of binary variables.
- *Branch and bound methods*, where the 0–1 combinations are explored through a binary tree, the feasible region is partitioned into sub-domains systematically, and valid upper and lower bounds are generated at different levels of the binary tree.

For MIQP problems, Fletcher and Leyffer (1995) indicate Generalized Benders' Decomposition (GBD) (Lazimy, 1985), Outer Approximation (OA), LP/QP based branch and bound, and Branch and Bound as the major solvers. See Roschchin et al. (1987) for a review of these methods.

Several authors agree on the fact that branch and bound methods are the most successful for mixed integer programs. Fletcher and Leyffer (1995) report a numerical study which compares different approaches, and Branch and Bound is shown to be superior by an order of magnitude. While OA and GBD techniques can be attractive for general mixed-integer nonlinear problems (MINLP), for MIQP at each node the relaxed QP problem can be solved without approximations and reasonably quickly (for instance, the Hessian matrix of each relaxed QP is constant).

As described by Fletcher and Leyffer (1995), the Branch and Bound algorithm for MIQP consists of solving and generating new QP problems in accordance with a tree search, where the nodes of the tree correspond to QP subproblems. Branching is obtained by generating child-nodes from parent-nodes according to branching rules, which can be based, for instance, on a priori specified priorities on integer variables, or on the amount by which the integer constraints are violated. Nodes are labeled as either pending, if the corresponding QP problem has not yet been solved, or fathomed, if the node has already been fully explored. The algorithm stops when all nodes have been fathomed. The success of the branch and bound algorithm relies on the fact that whole subtrees can be excluded from further exploration by fathoming the corresponding root nodes. This happens if the corresponding QP subproblem is either infeasible or an integer solution is obtained. In the second case, the corresponding value of the cost function serves as an upper bound on the optimal solution of the MIQP problem, and is used to further fathoming other nodes having greater optimal value or lower bound.

Some of the simulation results reported in this paper have been obtained in Matlab by using the commercial Fortran package (Fletcher and Leyffer, 1994) as a MIQP solver. This package can handle both dense and sparse MIQP problems. The latter has proven to be particularly effective to solve most of the optimal control problems for MLD systems. In fact, because of Eq. (47), the constraints have a triangular structure, and in addition most of the constraints generated by representation of logic facts involve only a few variables, which often leads to sparse matrices.

8. A case study: control of a gas supply system

The theoretical framework for modeling and controlling MLD systems developed in the previous sections is applied to the Kawasaki Steel Mizushima Works gas supply system described in (Akimoto et al., 1991).

8.1. Gas supply system

The system is depicted in Fig. 8. A steel-works generates three by-product gas, namely blast furnace gas (B gas), coke oven gas (C gas), and mixed gas (M gas) such as converter gas. These are known disturbances whose flow rates F_{BR}, F_{CR}, F_{MR} fluctuate with time. In order to provide a stable supply of high-caloric gas F_{BS}, F_{CS}, F_{MS} to the joint electric power plant, three holders dampen the by-product gas flows. The electric power plant is constituted by five boilers. Nos. 1 and 2 can use B and M gas and heavy oil as fuel, nos. 3, 4, and 5 can also use C gas. M gas is mixed with B gas to increase the thermal values (in calories) of the B gas. It is desired to save heavy oil by supplying by-product gas to boilers at a stationary rate. The physical quantities describing the

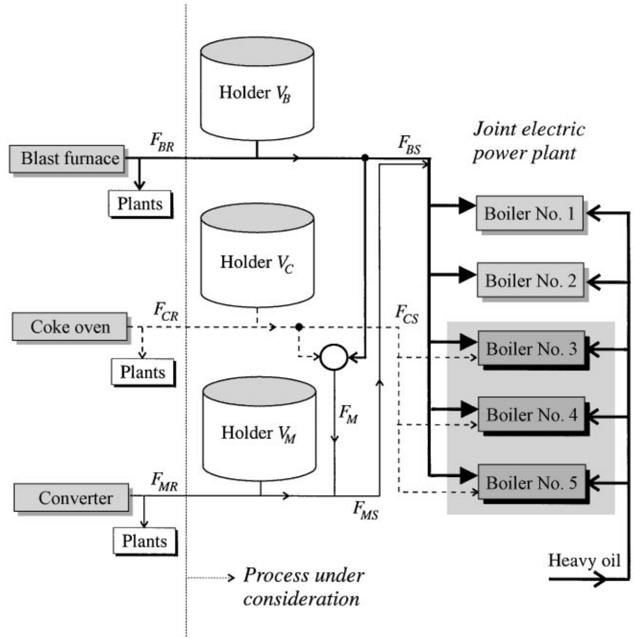


Fig. 8. Gas supply system.

model and the numerical values of the parameters are reported in Tables 2 and 3, respectively.

In order to model the system in discrete time, gas flow rates are assumed to be constant over the sampling period ΔT. Then, the dynamics of gas holders is given by

$$V_B(t+1) = V_B(t) + \Delta T[F_{BR}(t) - F_{BS}(t) - \alpha F_M(t)], \quad (59a)$$

$$V_C(t+1) = V_C(t) + \Delta T[F_{CR}(t) - F_{CS}(t) - (1-\alpha) F_M(t)], \quad (59b)$$

$$V_M(t+1) = V_M(t) + \Delta T[F_{MR}(t) - F_{MS}(t) + F_M(t)], \quad (59c)$$

where the amount of gas in each holder cannot exceed upper and lower limits:

$$\underline{V}_B \le V_B(t) \le \bar{V}_B, \quad (60a)$$

$$\underline{V}_C \le V_C(t) \le \bar{V}_C, \quad (60b)$$

$$\underline{V}_M \le V_M(t) \le \bar{V}_M. \quad (60c)$$

Due to boiler operation constraints at the joint electric power plant, the following input constraints hold for the supply amounts of B gas and C gas:

$$\underline{F}_{BS} \le F_{BS}(t) \le \bar{F}_{BS}, \quad (61a)$$

$$\underline{F}_{CS} \le F_{CS}(t) \le \bar{F}_{CS}. \quad (61b)$$

In addition,

$$F_{MS}(t) \ge 0, \quad (62a)$$

$$F_M(t) \ge 0. \quad (62b)$$

From the thermal balance before and after M gas is mixed with B gas, and considering the heating value of

Table 2
Physical quantities

Symbol	Meaning	Unit
$F_{BR}(t)$, $F_{CR}(t)$, $F_{MR}(t)$	Residual gas flow rates	m^3/h
$V_B(t)$, $V_C(t)$, $V_M(t)$	Gas volume held in the gas holder	m^3
$F_M(t)$	M gas flow rate produced from B + C	m^3/h
$F_{BS}(t)$, $F_{CS}(t)$, $F_{MS}(t)$	Gas flow rates suppliedto electric plant	m^3/h
r_1	Evaluated value of gas	yen/kcal
r_2	Profit obtained through combustion-of-gas-only	yen/h
r_3–r_8	Loss due to gas discharge of shortage ateach holder	yen/m^3
r_9	Loss because combustion-of-gas-only forless than $2\Delta T$ hours	yen
r_{10}–r_{15}	Penalty for suppressing the fluctuation of gas amount held	yen/m^3
r_{16}, r_{17}	Penalty for two/three boilers switchedsimultaneously	yen

Table 3
Model parameters

Symbol	Meaning	Value	Unit
α	Mixing ratio of B and C gas	0.52	
$\bar{V}_B$	Upper limit of B gas holder	180	km^3
$\underline{V}_B$	Lower limit of B gas holder	50	km^3
V_B^N	Standard value of B gas holder	110	km^3
$\bar{V}_C$	Upper limit of C gas holder	120	km^3
$\underline{V}_C$	Lower limit of C gas holder	10	km^3
V_C^N	Standard value of C gas holder	50	km^3
$\bar{V}_M$	Upper limit of M gas holder	90	km^3
$\underline{V}_M$	Lower limit of M gas holder	7	km^3
V_M^N	Standard value of M gas holder	40	km^3
$\bar{F}_{BS}$	Upper limit of B gas supply	1000	km^3/h
$\underline{F}_{BS}$	Lower limit of B gas supply	0	km^3/h
$\bar{F}_{CS}$	Upper limit of C gas supply	750	km^3/h
$\underline{F}_{CS}$	Lower limit of C gas supply	0	km^3/h
$\bar{q}_{BI}$	Upper limit of B gas calorie	1050	kcal/m^3
$\underline{q}_{BI}$	Lower limit of B gas calorie	720	kcal/m^3
$\bar{F}_{CS}^*$	Minimum value of gas-only	15	km^3/h
q_B	B gas calorie	742	kcal/m^3
q_C	C gas calorie	4600	kcal/m^3
q_M	M gas calorie	2520	kcal/m^3
ΔT	Sampling time	2	h
T	Horizons	4	Steps

the calorie-increased B gas, F_{MS}, F_{BS} must also satisfy the constraints

$$(\underline{q}_{BI} - q_B)F_{BS}(t) + (\underline{q}_{BI} - q_M)F_{MS}(t) \le 0, \quad (63a)$$

$$-(\bar{q}_{BI} - q_B)F_{BS}(t) - (\bar{q}_{BI} - q_M)F_{MS}(t) \le 0. \quad (63b)$$

Let F_{CS}^* be the minimum amount of C gas required for combustion-of-gas-only in an holder, and define $n(t)$ as the number of boilers burning C gas

$$n(t) = \begin{cases} 0 & \text{if } \underline{F}_{CS} \le F_{CS}(t) < F_{CS}^*, \\ 1 & \text{if } F_{CS}^* \le F_{CS}(t) < 2F_{CS}^*, \\ 2 & \text{if } 2F_{CS}^* \le F_{CS}(t) < 3F_{CS}^*, \\ 3 & \text{if } 3F_{CS}^* \le F_{CS}(t) < \bar{F}_{CS}, \end{cases}$$

where it is assumed that if C gas is enough for $n(t)$ boilers, $n(t)$ boilers burn C gas. The number of boilers $n(t)$ can be expressed as the sum of the 0–1 variables $n_1(t)$, $n_2(t)$, $n_3(t)$, defined by the relations

$$[n_1(t) = 0] \leftrightarrow [F_{CS} \ge F_{CS}^*],$$

$$[n_2(t) = 0] \leftrightarrow [F_{CS} \ge 2F_{CS}^*],$$

$$[n_3(t) = 0] \leftrightarrow [F_{CS} \ge 3F_{CS}^*],$$

$$[n_3(t) = 1] \rightarrow [n_1(t) = 1],\ [n_2(t) = 1],$$

$$[n_2(t) = 1] \rightarrow [n_1(t) = 1],$$

$$[n_1(t) = 0] \rightarrow [n_2(t) = 0],\ [n_3(t) = 0],$$

$$[n_2(t) = 0] \rightarrow [n_3(t) = 0]$$

or, by transforming into linear inequalities,

$$F_{CS}(t) \ge F_{CS}^* + (\underline{F}_{CS} - F_{CS}^*)[1 - n_1(t)], \quad (64a)$$

$$F_{CS}(t) \le F_{CS}^* - \varepsilon + (\bar{F}_{CS} - F_{CS}^* + \varepsilon)n_1(t), \quad (64b)$$

$$F_{CS}(t) \ge 2F_{CS}^* + (\underline{F}_{CS} - 2F_{CS}^*)[1 - n_2(t)], \quad (64c)$$

$$F_{CS}(t) \le 2F_{CS}^* - \varepsilon + (\bar{F}_{CS} - 2F_{CS}^* + \varepsilon)n_2(t), \quad (64d)$$

$$F_{CS}(t) \ge 3F_{CS}^* + (\underline{F}_{CS} - 3F_{CS}^*)[1 - n_3(t)], \quad (64e)$$

$$F_{CS}(t) \le 3F_{CS}^* - \varepsilon + (\bar{F}_{CS} - 3F_{CS}^* + \varepsilon)n_3(t), \quad (64f)$$

$$n_3(t) - n_1(t) \le 0, \quad (64g)$$

$$n_3(t) - n_2(t) \le 0, \quad (64h)$$

$$n_2(t) - n_1(t) \le 0, \quad (64i)$$

$$n(t) = n_1(t) + n_2(t) + n_3(t), \quad (64j)$$

where ε is a properly small positive constant. Moreover, the following specifications must be taken into account:

1. When the combustion-of-gas-only is practiced it should be continued for at least $2\Delta T$ hours.
2. If the number of boilers for combustion-of-gas-only decreases, the number of the decrease should be one at the time, and hence simultaneous changeover of multiple boilers needs high penalty.

In order to take into account these conditions, define $n(t) - n(t-1) = \Delta n^+(t) - \Delta n^-(t)$, $\Delta n^+(t), \Delta n^-(t) \geq 0$, and introduce three 0-1 variables $k_1(t)$, $k_2(t)$, $k_3(t)$, such that $\Delta n^-(t) = k_1(t) + k_2(t) + k_3(t)$. Then

$$k_1(t) \geq k_2(t), \tag{65a}$$

$$k_1(t) \geq k_3(t), \tag{65b}$$

$$k_2(t) \geq k_3(t), \tag{65c}$$

$$n_1(t) + n_2(t) + n_3(t) - n(t-1) + k_1(t) + k_2(t) + k_3(t) \geq 0, \tag{65d}$$

$$n_1(t) + n_2(t) + n_3(t) - n(t-1) + k_1(t) + k_2(t) + k_3(t) \leq 3[1 - k_1(t)]. \tag{65e}$$

In order to take into account the second specification, let

$$s(t) = \begin{cases} \Delta n^-(t) & \text{if } \Delta n^+(t-1) > 0 \\ 0 & \text{if } \Delta n^+(t-1) = 0 \end{cases}$$

and let $\gamma_1(t) \in \{0,1\}$ such that $[\gamma_1(t) = 1] \leftrightarrow [n(t-1) > n(t-2)]$. Then, by recalling Eq. (5b),

$$-n(t-1) + n(t-2) \geq \tfrac{7}{2} - 4\gamma_1(t), \tag{66a}$$

$$n(t-1) - n(t-2) \leq 4\gamma_1(t), \tag{66b}$$

$$s(t) \geq 0, \tag{66c}$$

$$s(t) \leq 3\gamma_1(t), \tag{66d}$$

$$s(t) \leq k_1(t) + k_2(t) + k_3(t), \tag{66e}$$

$$s(t) \geq k_1(t) + k_2(t) + k_3(t) - 3[1 - \gamma_1(t)]. \tag{66f}$$

Note that this formulation assumes that boilers nos. 3–5 are activated according to a predefined hierarchy, otherwise a more complex description which distinguishes the numbers of the boilers burning C gas should be adopted.

In order to take into account the profit figure defined in Akimoto et al. (1991), consider the following *profit* variable:

$$p(t) \triangleq r_1 \Delta T [q_B F_{BS}(t) + q_C [F_{CS}(t) - F^*_{CS} n(t)] + q_M F_{MS}(t)] + r_2 \Delta T n(t)$$

which should be maximized. This can be achieved by minimizing a new (slack) variable $w(t)$ defined by the inequalities

$$p(t) \geq p_e(t) - w(t), \tag{67a}$$

$$w(t) \geq 0, \tag{67b}$$

where $p_e(t)$ a goal profit value, defined below.

In conclusion, the gas supply system can be represented as a time-varying MLD system described by

$$x(t) \triangleq \begin{bmatrix} V_B(t) \\ V_C(t) \\ V_M(t) \\ n(t-1) \\ n(t-2) \end{bmatrix}, \qquad u(t) \triangleq \begin{bmatrix} F_{BS}(t) - F_{BR}(t) \\ F_{CS}(t) - F_{CR}(t) \\ F_{MS}(t) - F_{MR}(t) \\ F_M(t) \end{bmatrix},$$

$$\delta(t) \triangleq \begin{bmatrix} n_1(t) \\ n_2(t) \\ n_3(t) \\ \gamma_1(t) \\ k_1(t) \\ k_2(t) \\ k_3(t) \end{bmatrix}, \qquad z(t) \triangleq \begin{bmatrix} s(t) \\ w(t) \end{bmatrix},$$

and Eqs. (59)–(67).

8.2. Predictive control of the gas supply system

As during actual plant operation human operators try to keep the volume of gas in each holder as constant as possible at normal values V^N_B, V^N_C, V^N_M, respectively, it is natural to define these values as set-points for V_B, V_C, V_M.

The approach described in Section 6 requires the existence of an equilibrium pair, and hence in principle only step gas flow disturbances allow the definition of such an equilibrium. We hence assume that $F_{BR}(k|t) = F_{BR,e}(t) \triangleq F_{BR}(T|t)$ for all $k \geq T$ ($F_{CR,e}(t)$ and $F_{MR,e}(t)$ are defined analogously), relying on the fact that the receding horizon mechanism will mitigate such a restrictive assumption about future disturbances (Campo and Morari, 1989), and that these disturbances are in any case obtained by other controlled processes. Then, by defining $n_e(t) = n_{1,e}(t) + n_{2,e}(t) + n_{3,e}(t)$ as the number of boilers which can be fed by a constant gas rate $F_{CR,e}(t)$, we set $x_e(t) \triangleq [V^N_B \; V^N_C \; V^N_M \; n_e(t) \; n_e(t)]'$, $u_e \triangleq [0 \; 0 \; 0 \; 0]'$, $\delta_e(t) \triangleq [n_{1,t}(t) \; n_{2,e}(t) \; n_{3,e}(t) \; 0 \; 0 \; 0 \; 0]'$, $z_e \triangleq [0 \; 0]'$, and define the quantity $p_e(t)$ in (Eq. (67a)) accordingly. Note that, because of the terminal constraint $x(T|t) = x_e(t)$, feasibility is guaranteed only for gas flow disturbances which are constant for $t \geq T$.

The feedback control law (53)–(55) is adopted in order to operate the gas supply system. In addition, we add in (53) the linear term $Q_\$ w(k|t)$. Since $w_e = 0$, $w(t) \geq 0$, as observed in Remark 4 such a modification does not alter the stability results of Theorem 1. The resulting trajectories are depicted in Figs. 9 and 10, and correspond to the prediction horizon $T = 4$, and weights $Q_\$ = 50$, $Q_1 = \text{diag}(10^{-2}, 10^{-1}, 10^{-1}, 10^2)$, $Q_2 = \text{diag}\,(10^{-2}, 10^{-2}, 10^{-2}, 10^{-2}, 10^3, Qr_{16}, Qr_{17})$, $Q_3 = \text{diag}(r_9, 10^{-3})$, $Q_4 = \text{diag}(\tfrac{1}{2}(r_{10} + r_{11}), \tfrac{1}{2}(r_{12} + r_{13}), \tfrac{1}{2}(r_{14} + r_{15}), 10^{-4}, 10^{-4})$.

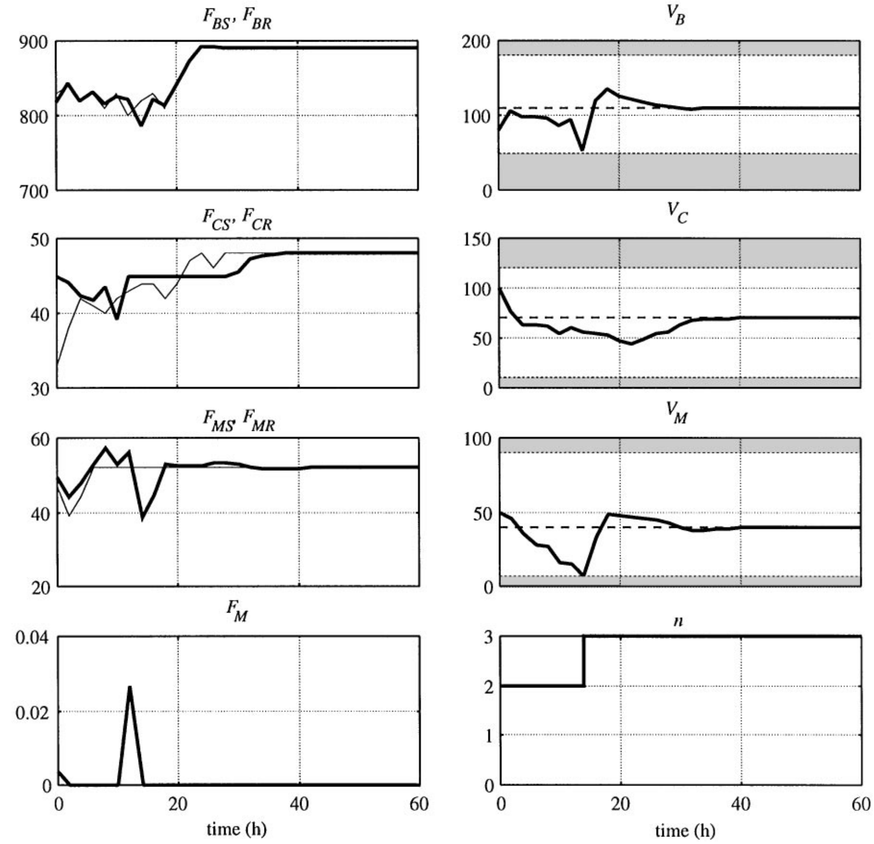


Fig. 9. Predictive control of the gas supply system, $w(k|t) \geq 0$, $\forall k$, $\forall t$. Thick lines: F_{BS}, F_{CS}, F_{MS}, F_M, V_B, V_C, V_M, n; thin lines: F_{BR}, F_{CR}, F_{MR}; dashed lines: V_B^N, V_C^N, V_M^N.

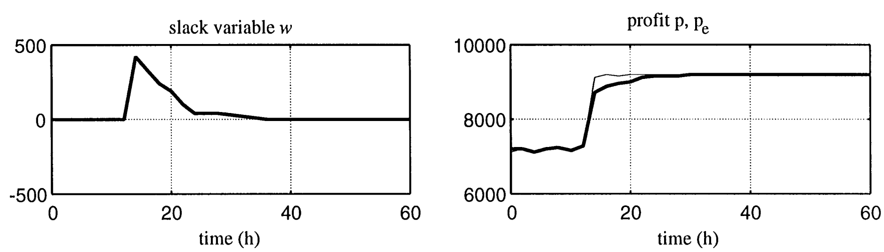


Fig. 10. Predictive control of the gas supply system, $w(k|t) \geq 0$, $\forall k$, $\forall t$. Slack variable w and profit variable p (thick lines), p_e (thin line).

In order to maximize the profit, one could be tempted to remove the constraint (67b). On the other hand, with such a modification stability properties are no longer guaranteed. For converging gas flow disturbances, a compromise is obtained by introducing constraint (67b) only after a finite time $t_S > 0$, namely by imposing in the optimization problem the constraints $w(k|t) \geq 0$ only for $k \geq T - t + t_S$. In this way, stability is restored, and feasibility preserved. Figs. 11 and 12 show the results obtained by setting $t_S = 12$. Note that the risky approach consisting of maximizing profit without constraining $w(t)$ results in a more aggressive transient behavior, as witnessed by the V_B, V_C, and V_M trajectories.

8.3. Computational complexity

At each time step t, the MIQP problem which derives from Eqs. (9), (45), and (46) has the structure (48), involves 121 linear constraints, 25 continuous variables, and 28 integer variables. The problem has a sparseness of around 93%. Concerning computational times, on a Sun SPARCStation 4 at time $t = 0$, for instance, the MIQP problem is solved by the sparse version of the package (Fletcher and Leyffer, 1994) in 1.20 s (8 QP subproblems). Simulation computational time is also saved by exploiting the information about the previous solution, namely by shifting the previous optimal solution, which is a

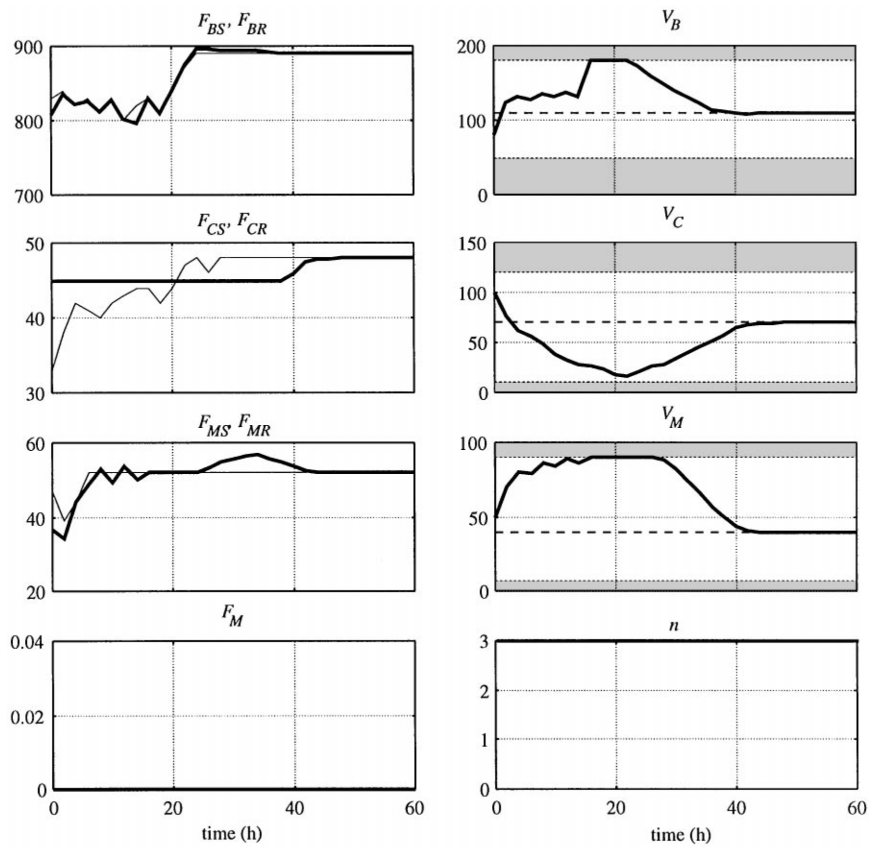


Fig. 11. Predictive control of the gas supply system, $w(k|t) \geq 0\ \forall k \geq T - t + t_S$. Thick lines: F_{BS}, F_{CS}, F_{MS}, F_M, V_B, V_C, V_M, n; thin lines: F_{BR}, F_{CR}, F_{MR}; dashed lines: V_B^N, V_C^N, V_M^N.

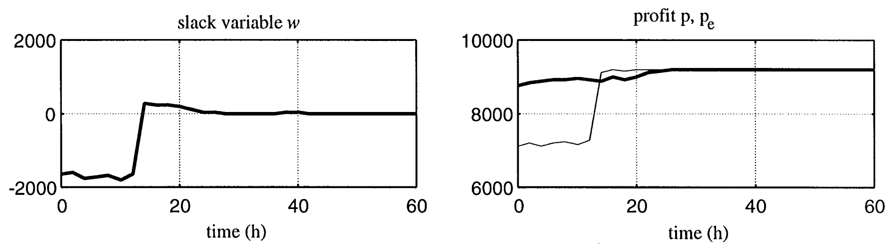


Fig. 12. Predictive control of the gas supply system, $w(k|t) \geq 0$, $\forall k$, $\forall t$. Slack variable w and profit variable p (thick lines), p_e (thin line).

Table 4
Profit and loss coefficients for on-line optimization.

Symbol	Value	Symbol	Value
r_1	0.0025	r_{10}	0.01
r_2	200	r_{11}	0.01
r_3	0.0019	r_{12}	0.10
r_4	0.0022	r_{13}	0.10
r_5	0.0021	r_{14}	0.01
r_6	100	r_{15}	0.01
r_7	100	r_{16}	500
r_8	100	r_{17}	1000
r_9	500		

feasible initial condition, as observed in the proof of Theorem 1. Considering that for the gas supply system $\Delta T = 2$ h, real time implementation of the proposed scheme is reasonable.

9. Conclusions

Motivated by the key idea of transforming propositional logic into linear mixed-integer inequalities, and by the existence of techniques for solving mixed-integer quadratic programming, this paper has presented a framework for modeling and controlling systems described by both dynamics and logic, and subject to operating constraints, denoted as *mixed logical dynamical* (MLD) systems. For these systems, a systematic control design method based on model predictive control ideas has been presented, which provides stability, tracking, and constraint fulfillment properties. The proposed strategy seems to be particularly appealing for higher-level control and optimization of complex systems.

Appendix A

Below we describe a simple algorithm to test well-posedness of a system in the form (9). Consider the problem of checking if for all $v \in X_v$ there exists only one vector $s \in X_s$ satisfying

$$s = H_1 w + H_2 v,$$

$$K_1 w \leq K_2 v + K_3$$

for some $w \in X_w$ ($X_{v(sw)}$ are of the form $\mathbb{R}^i \times \{0,1\}^j$, $i, j \geq 0$). If this does not hold, then there exist vectors $v \in X_v$, $s_- \neq s_+ \in X_s$, and an index $i \in \{1, \ldots, n_s\}$ such that

$$s_-^i = H_1 w_- + H_2 v,$$

$$s_+^i = H_1 w_+ + H_2 v,$$

$$K_1 w_- \leq K_2 v + K_3,$$
$$K_1 w_+ \leq K_2 v + K_3,$$

$$s_-^i < s_+^i,$$

where s^i denotes the ith component (or row) of s. An algorithm for testing this condition is the following

Algorithm 1

1. Let ε be a small tolerance.
2. For $i = 1, \ldots, n_s$

   2.1. Test feasibility of the problem

$$\begin{cases} H_1^i (w_- - w_+) \leq -\varepsilon, \\ K_1 w_- \leq K_2 v + K_3, \\ K_1 w_+ \leq K_2 v + K_3. \end{cases} \quad \text{(A.1)}$$

   2.2. If Eq. (A.1) is feasible, the system is not well posed. Stop.
3. Stop. The system is well posed.

Note that there is no need to check Eq. (A.1) for $H_1^i = 0$, as it is trivially infeasible. To test well-posedness of systems (9), one can apply Algorithm 1 along with

$$s = \begin{bmatrix} x(t+1) \\ y(t) \end{bmatrix}, \quad v = \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}, \quad w = \begin{bmatrix} \delta(1) \\ z(t) \end{bmatrix},$$

$$H_1 = \begin{bmatrix} B_2 & B_3 \\ D_2 & D_3 \end{bmatrix}, \quad H_2 = \begin{bmatrix} A & B_1 \\ C & D_1 \end{bmatrix},$$

$$K_1 = [E_2 \quad E_3], \quad K_2 = [E_4 \quad E_1], \quad K_3 = E_5.$$

Checking if A, B_1, B_2, B_3 satisfy the integrality condition on $x_\ell(t+1)$ is usually not needed, as typically

$$x_\ell^j(t+1) = e^i \begin{bmatrix} \delta(t) \\ x_\ell(t) \end{bmatrix},$$

where e^i denotes the ith row of the $(r_\ell + n_\ell)$ identity matrix.

Acknowledgements

The authors thank Kazuya Asano and Akimoto Keiichi for providing informations about the Kawasaki gas supply system, and Sven Leyffer and Roger Fletcher for the MIQP solver. Alberto Bemporad was supported by the Swiss National Science Foundation.

References

Akimoto, K., Sannomiya, N., Nishikawa, Y., & Tsuda, T. (1991). An optimal gas supply for a power plant using a mixed integer programming model. *Automatica*, *27*(3), 513–518.

Branicky, M. S., Borkar, V. S., & Mitter, S. K. (1998). A unified framework for hybrid control: model and optimal control theory. *IEEE Trans. Automat. Control*, *43*(1), 31–45.

Campo, P. J., & Morari, M. (1989). Model predictive optimal averaging level control. *AIChE J.*, *35*(4), 579–591.

Cavalier, T. M., Pardalos, P. M., & Soyster, A. L. (1990). Modeling and integer programming techniques applied to propositional calculus. *Comput. Oper. Res.*, *17*(6), 561–570.

Christiansen, D. (1997). *Electronics engineers' handbook* (*4th ed.*). New York: IEEE Press/McGraw Hill, Inc.

Fletcher, R., & Leyffer, S. (1994). *A mixed integer quadratic programming package*. Technical Report. Dept. of Mathematics, University of Dundee, Scotland, U.K.

Fletcher, R., & Leyffer, S. (1995). *Numerical experience with lower bounds for MIQP branch-and-bound*. Technical Report. Dept. of Mathematics, University of Dundee, Scotland, U.K. SIAM J. Optim., submitted. http://www.mcs.dundee.ac.uk:8080/sleyffer/miqp_art.ps.Z.

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

Grossmann, R. L., Nerode, A., Ravn, A. P., & Rischel, H. (1993). *Hybrid systems*. Lecture Notes in Computer Science (Vol. 736). New York: Springer Verlag.

Hayes, J. P. (1993). *Introduction to digital logic design*. Reading: Addison-Wesley.

Keerthi, S. S., & Gilbert, E. (1988). Optimal infinite-horizon feedback laws for a general class of constrained discrete time systems: stability and moving-horizon approximations. *J. Optimi. Theory Appli.*, *2*, 265–293.

Lazimy, R. (1985). Improved algorithm for mixed-integer quadratic programs and a computational study. *Math. Programming*, *32*, 100–113.

Lee, J. H., & Cooley, B. (1997). Recent advances in model predictive control. In: *Chemical Process Control — V* (Vol. 93, no. 316, pp. 201–216b), AIChe Sympo. Ser. New York: American Institute of Chemical Engineers.

Leyffer, S. (1993). *Deterministic methods for mixed integer nonlinear programming*. Ph.D. dissertation. University of Dundee. Scotland, U.K.

Lygeros, J., Godbole, D.N., & Sastry, S.S. (1996). A game theoretic approach to hybrid system design. In: Alur, R., & Henzinger, T. (Eds.), *Hybrid systems III* (pp. 1–12). *Lecture Notes in Computer Science* (Vol. 1066). Berlin: Springer.

Mayne D. Q. (1997). Nonlinear model predictive control: an assessment In: *Chemical Process Control — V* (Vol. 93, No. 316, pp. 217–231), AIChe Symp. Ser. New York: American Institute of Chemical Engineers.

Mendelson, E. (1964). *Introduction to mathematical logic*. New York: Van Nostrand.

Mitter, S. K. (1997). Logic and mathematical programming. In: Morse, A. S. (Ed.), *Control using logic based switching*, *Fall 1995 Block Island*, *RI*, *USA* (pp. 79–91). London UK: Springer-Verlag.

Passino, K. M., Michel, A. N., & Antsaklis, P. J. (1994). Lyapunov stability of a class of discrete event systems. *IEEE Trans. Automat. Control*, *2*(39), 269–279.

Qin, S. J., & Badgewell, T. A. (1997). An overview of industrial model predictive control technology. In: *Chemical process control — V* (Vol. 93, No. 316, pp. 232–256), AIChe Symp. Ser. New York, American Institute of Chemical Engineers.

Raman, R., & Grossmann, I. E. (1991). Relation between MILP modeling and logical inference for chemical process synthesis. *Comput. Chem. Engng*, *15*(2), 73–84.

Raman, R., & Grossmann, I. E. (1992). Integration of logic and heuristic knowledge in MINLP optimization for process synthesis. *Computers Chem. Engng*, *16*(3), 155–171.

Roschchin, V. A., Volkovich, O. V., & Sergienko, I. V. (1987). Models and methods of solution of quadratic integer programming problems. *Cybernetics*, *23*, 289–305.

Slupphaug, O., & Foss, B. A. (1997). Model predictive control for a class of hybrid systems. *Proc. European Control Conf.* Brussels, Belgium.

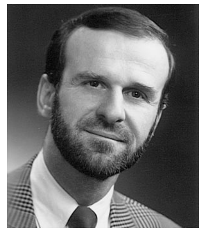
Slupphaug, O., Vada, J., & Foss, B. A. (1997). MPC in systems with continuous and discrete control inputs. *Proc. American Control Conf.* Albuquerque, NM, USA.

Tyler, M. L., & Morari, M. (1999). Propositional logic in control and monitoring problems. *Automatica* (in print).

Williams, H. P. (1977). Logical problems and integer programming. *Bull. Institute Math. Appli.*, *13*, 18–20.

Williams, H. P. (1987). Linear and integer programming applied to the propositional calculus. *Int. J. Systems Res. Inform Sci.*, *2*, 81–100.

Williams, H. P. (1993). *Model building in mathematical programming* (3rd ed.). New York: Wiley.



Manfred Morari was appointed head of the Automatic Control Laboratory at the Swiss Federal Institute of Technology (ETH) in Zurich in 1994. Before that he was the McCollum-Corcoran Professor of Chemical Engineering and Executive Officer for Control and Dynamical Systems at the California Institute of Technology. He obtained the diploma from ETH Zurich and the PhD from the University of Minnesota, both in chemical engineering. His interests are in the areas of process control and design. In recognition of his research contributions, he received numerous awards among them, the Donald P. Eckman Award of the Automatic Control Council, the Allan P. Colburn Award and the Professional Progress Award of the AIChE, the Curtis W. McGraw Research Award of the ASEE and was elected to the National Academy of Engineering (US). Professor Morari has held appointments with Exxon R & E and ICI and has consulted internationally for a number of major corporations.

Alberto Bemporad joined the Automatic Control Laboratory at the Swiss Federal Institute of Technology (ETH) in Zurich as a postdoctoral fellow in 1997. He obtained the degree in electrical engineering from the University of Florence, Italy in 1993 and the PhD in control engineering from the Dipartimento di Sistemi e Informatica, University of Florence, Italy in 1997. He spent the academic year 1996/97 at the Center for Robotics and Automation, Dept. Systems Science & Mathemathics, Washington University, St. Louis, MO. He received the IEEE Centre and South Italy section "G. Barzilai" and the AEI (Italian Electrical Association) "R. Mariani" awards. His research interests are in the area of predictive control, hybrid systems, and robotics.