European Journal of Control (2003)9:177–189

European Journal of Control

Hybrid Systems Modeling and Control

Manfred Morari*, Mato Baotić†, Francesco Borrelli‡

Automatic Control Laboratory, ETH Zentrum, CH-8092, Zurich, Switzerland

Piece-wise affine and mixed logical dynamical models for discrete time linear hybrid systems are reviewed. Constrained optimal control problems with linear and quadratic objective functions are defined. Some results on the structure and computation of the optimal control laws are presented. The effectiveness of the techniques is illustrated on a wide range of practical applications.

Keywords: Hybrid Systems; Mixed-Integer Programming; Optimal Control; Parametric Programming; Switched Systems

1. Introduction

Hybrid systems can switch between many operating modes where each mode is governed by its own characteristic dynamical laws. Mode transitions are triggered by variables crossing specific thresholds (state events), by the lapse of certain time periods (time events), or by external inputs (input events). A simple example of a hybrid system would be a car. The dynamics of the car switch when a gear shift occurs, either because the driver moves the stick shift (input event) or because the state variable "speed" exceeds a specified threshold (state event) in the case of an automatic transmission.

A hybrid system can have both continuous and discrete states. If a digital computer (with discrete states only) is used to control a physical system (typically with continuous states only) a hybrid system is formed. Indeed, a mathematical definition of hybrid systems was first introduced in computer science in this context.

Another special example of a hybrid system would be a linear system under feedback control with actuator constraints. When the actuator hits a constraint the dynamics change. Finally, consider a metabolic network, where the (dis)appearance of an enzyme turns a particular reaction step on or off. Its behavior can also be modeled as a hybrid system.

These examples illustrate that hybrid systems are very common in engineering and many systems encountered in every day life can be effectively modeled as hybrid systems as well. Hybrid systems are a special class of nonlinear systems but most of the nonlinear system and control theory does not apply because it requires certain smoothness assumptions. For the same reason we also cannot simply use linear control theory in some approximate manner to design controllers for hybrid systems.

Not surprisingly until the 1990s most tools available for the analysis and control of hybrid systems were ad hoc supported by extensive simulation. For example, the design of anti-windup compensators was an art

Correspondence and offprint requests to: Manfred Morari
Automatic Control Laboratory, ETH Zentrum, CH-8092, Zurich, Switzerland.
*Email: morari@control.ee.ethz.ch
†Email: baotic@control.ee.ethz.ch
‡Email: borrelli@control.ee.ethz.ch

Received 2 April 2003; Accepted 17 April 2003.
Recommended by L. Postlethwaite, J. M. Maciejowski, K. Glover and P. J. Fleming.

with numerous different structures proposed and "proven" through case studies. Finally, in [34] all structures were shown to be identical and to differ only through the choice of a parameter. This opened up the area to extensive research so that now systematic optimal control based tools for anti-windup design with stability and performance guarantees are available [30,37].

About five years ago two factors motivated us to embark on a broad research program in the hybrid system area. First of all, we were encouraged by the success of model predictive control to deal with complex constrained control problems in industry. Second, we were impressed by the progress in the performance of commercial tools for solving mixed-integer programming problems and their impact on decision making in complex industrial environments [10,11]. Our first foray [47] was to model hybrid systems so that the associated optimal control problems become mixed-integer programs and can be solved efficiently with commercial codes. The success led to a wide range of activities on the analysis and synthesis of hybrid systems as will be sketched out in this paper.

2. Scope

This overview paper is intended for the newcomer in the area. It should communicate the motivation for the problems studied and outline the approach taken. It should provide a glimpse at the type of theoretical results that have been obtained and the scope of the algorithms available. It will focus on optimal control. Topics like identification [24] and filter design [23] will not be covered. The emphasis will be on tools developed in our group and closely related work. To put our work in perspective the reader is referred to our web site http://control.ee.ethz.ch from which an extensive set of papers and software tools are available.

All our work has been limited to discrete-time systems where the continuous dynamics are affine. By virtue of this restriction many interesting and complex mathematical issues have been removed. Our experience has shown, that even with these assumptions the theoretical and computational challenges are formidable. However, together with our research partners we have been able to solve a large number of real problems as will be reported in this paper.

Large parts of this paper are based on [14] which contains an extensive list of references, more details on the theory and the algorithms as well as the proofs of the results reported here.

3. Modeling

3.1. Piece-Wise Affine Systems

In this paper, our discussion will focus on discrete-time piecewise affine (PWA) systems.

Definition 1 (Polyhedron). A set $\Theta \subseteq \mathbb{R}^q$ of the form $\Theta = \{\theta \in \mathbb{R}^q | \Gamma\theta \leq \gamma\}$ for some $\Gamma \in \mathbb{R}^{r \times q}, \gamma \in \mathbb{R}^r$ is called polyhedron.

Piece-wise affine systems [43] are defined by partitioning the state + input space into polyhedral regions and associating with each region a different linear state-update equation

$$\begin{aligned} x(t+1) = & A^i x(t) + B^i u(t) + f^i \\ & \text{if } \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathscr{P}_i \\ & i = 1, \ldots, s \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$, $\{\mathscr{P}_i\}_{i=1}^s$ is a polyhedral partition of the set of the state + input space $\mathscr{P} \subset \mathbb{R}^{n+m}$, $n \stackrel{\Delta}{=} n_c + n_\ell$, $m \stackrel{\Delta}{=} m_c + m_\ell$. We assume that $\mathscr{P}$ is closed and bounded and we denote with $x_c \in \mathbb{R}^{n_c}$ and $u_c \in \mathbb{R}^{m_c}$ the continuous components of the state and input vector, respectively.

Definition 2. We say that the PWA system (1) is *continuous* if the mapping $(x_c(t), u_c(t)) \mapsto x_c(t+1)$ is continuous and $n_\ell = m_\ell = 0$.

We emphasize that this modeling framework of PWA systems is very general. It allows (i) the system to be discontinuous, (ii) both states and inputs to assume continuous and logic values, (iii) events to be both internal, i.e., caused by the state reaching a particular boundary, and exogenous, i.e., one can decide when to switch to some other operating mode and (iv) states and inputs to fulfill piecewise-linear constraints.

Discrete-time PWA models can describe a large number of processes, such as discrete-time linear systems with static piecewise-linearities, discrete-time linear systems with logic states and inputs or switching systems where the dynamic behavior is described by a finite number of discrete-time linear models, together with a set of logic rules for switching among these models. Moreover, PWA systems can *approximate* nonlinear discrete-time dynamics via multiple linearizations at different operating points.

Also note that other types of hybrid systems, namely linear complementarity (LC) systems [31,32,48], extended linear complementarity (ELC) systems [18], max–min-plus-scaling (MMPS) systems [19], and mixed logical dynamical (MLD) systems [7] are all equivalent in their discrete time version to the discrete-time linear PWA systems (cf. [5,33]). Thus, the theoretical properties and tools can be easily transferred from one class to another.

PWA models are, however, in most contexts not a "natural" system description that follows directly from engineering specifications. Moreover, they are not suitable for recasting analysis/synthesis problems into more compact optimization problems. The MLD framework [7] described in the following section has been developed for such a purpose. In particular, MLD models can be used to recast hybrid dynamical optimization problems into mixed-integer linear and quadratic programs [38].

3.2. Mixed Logical Dynamical Systems

MLD systems [7] allow specifying the evolution of continuous variables through linear dynamic equations, of discrete variables through propositional logic statements and automata, and the mutual interaction between the two. The key idea of the approach consists of embedding the logic part in the state equations by transforming Boolean variables into 0–1 integers, and translating logic relations into mixed-integer linear inequalities [7,41,49]. The following correspondence between a Boolean variable X and its associated binary variable δ will be used:

$$\begin{aligned} X &= \text{true} \quad \Leftrightarrow \delta = 1, \\ X &= \text{false} \quad \Leftrightarrow \delta = 0. \end{aligned} \tag{2}$$

Linear dynamics are represented by difference equations $x(t+1) = Ax(t) + Bu(t)$, $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$. Boolean variables can be defined from linear-threshold conditions over the continuous variables: $[X = \text{true}] \leftrightarrow [a'x \leq b]$, $x,\ a \in \mathbb{R}^n$, $b \in \mathbb{R}$, with $(a'x - b) \in [m, M]$. This condition can be equivalently expressed by the two mixed-integer linear inequalities:

$$\begin{aligned} a'x - b &\leq M(1-\delta), \\ a'x - b &> m\delta. \end{aligned} \tag{3}$$

Similarly, a relation defining a continuous variable z depending on the logic value of a Boolean variable X

$$\text{IF } X \text{ THEN } z = a_1'x - b_1 \text{ ELSE } z = a_2'x - b_2,$$

can be expressed as

$$\begin{aligned} (m_2 - M_1)\delta + z &\leq a_2'x - b_2 \\ (m_1 - M_2)\delta - z &\leq -a_2'x - b_2 \\ (m_1 - M_2)(1-\delta) + z &\leq a_1'x - b_1 \\ (m_2 - M_1)(1-\delta) - z &\leq a_1'x + b_1, \end{aligned} \tag{4}$$

where, assuming that $x \in \mathscr{X} \subset \mathbb{R}^n$ and $\mathscr{X}$ is a given bounded set,

$$M_i \geq \sup_{x \in \mathscr{X}} (a_i'x - b_i), \quad m_i \leq \inf_{x \in \mathscr{X}} (a_i'x - b_i), \quad i = 1, 2, \tag{5}$$

are upper and lower bounds on $(a_1'x - b_2)$ and $(a_2'x - b_2)$, respectively. Note that (4) also represents the hybrid product $z = \delta(a_1'x - b_1) + (1-\delta)(a_2'x - b_2)$ between binary and continuous variables.

A Boolean variable X_n can be defined as a Boolean function of Boolean variables $f : \{\text{true}, \text{false}\}^{n-1} \to \{\text{true}, \text{false}\}$, namely

$$X_n \leftrightarrow f(X_1, X_2, \ldots, X_{n-1}), \tag{6}$$

where f is a combination of "not" ($\sim$), "and" ($\wedge$), "or" ($\vee$), "exclusive or" ($\oplus$), "implies" ($\leftarrow$), and "iff" ($\leftrightarrow$) operators. The logic expression (6) is equivalent to its conjunctive normal form (CNF) [49]

$$\bigwedge_{j=1}^{k} \Big(\Big(\bigvee_{i \in P_j} X_i \Big) \bigvee \Big(\bigvee_{i \in N_j} \sim X_i \Big) \Big),$$
$$N_j, P_j \subseteq \{1, \ldots, n\}.$$

Subsequently, the CNF can be translated into the set of integer linear inequalities

$$\begin{cases} 1 \leq \sum_{i \in P_1} \delta_i + \sum_{i \in N_1} (1 - \delta_i), \\ \vdots \\ 1 \leq \sum_{i \in P_k} \delta_i + \sum_{i \in N_k} (1 - \delta_i). \end{cases} \tag{7}$$

Alternative methods for translating any logical relation between Boolean variables into a set of linear integer inequalities can be found in chapter 2 of [35]. In [35], the reader can also find a more comprehensive and complete treatment of the topic.

By collecting the equalities and inequalities derived from the representation of the hybrid system we obtain the MLD system [7]

$$x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t), \tag{8a}$$
$$y(t) = Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t), \tag{8b}$$
$$E_2\delta(t) + E_3z(t) \leq E_1u(t) + E_4x(t) + E_5, \tag{8c}$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$ is a vector of continuous and binary states, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ are the inputs, $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$ the outputs, $\delta \in \{0,1\}^{r_\ell}$, $z \in \mathbb{R}^{r_c}$ represent auxiliary binary and continuous variables, respectively, which are introduced when transforming logic relations into mixed-integer linear inequalities, and A, B_{1-3}, C, D_{1-3}, E_{1-5} are matrices of suitable dimensions.

Note that the constraints (8c) allow us to specify additional linear constraints on continuous variables (e.g., constraints on physical variables of the system), and logic constraints of the type "$f(\delta_1, \dots, \delta_n) = 1$", where f is a Boolean expression. The ability to include constraints, constraint prioritization, and heuristics adds to the expressiveness and generality of the MLD framework. Note also that the description (8) appears to be linear, with the nonlinearity concentrated in the integrality constraints over binary variables.

The translation of a description of a hybrid dynamical system into mixed integer inequalities is the objective of the tool HYSDEL (HYbrid Systems DEscription Language), which automatically generates an MLD model from a high-level textual description of the system [45]. Given a textual description of the logic and dynamical parts of the hybrid system, HYSDEL returns the matrices A, B_{1-3}, C, D_{1-3}, E_{1-5} of the corresponding MLD form (8). A full description of HYSDEL is given in [46]. The compiler is available for download at http://control.ethz.ch/ ~ hybrid/hysdel.

Example. The following switching system

$$\begin{cases} x(t+1) = 0.8\begin{bmatrix} \cos\alpha(t) & -\sin\alpha(t) \\ \sin\alpha(t) & \cos\alpha(t) \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \\ \alpha(t) = \begin{cases} \dfrac{\pi}{3} & \text{if } [1\ 0]x(t) \geq 0 \\ \dfrac{\pi}{3} & \text{if } [1\ 0]x(t) < 0 \end{cases}, \\ x(t) \in [-10, 10] \times [-10, 10], \\ u(t) \in [-1, 1] \end{cases} \tag{9}$$

was first proposed in [7]. System (9) can be rewritten in form (8) as shown in [7].

$$\begin{cases} x(t+1) = [\quad I \quad I \quad]z(t) \\ \begin{bmatrix} 5 \\ -5-\varepsilon \\ -M \\ -M \\ M \\ M \\ M \\ M \\ -M \\ -M \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \delta(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ I & 0 \\ -I & 0 \\ 0 & 1 \\ 0 & -I \\ I & 0 \\ -I & 0 \\ 0 & I \\ 0 & -I \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} z(t) \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ B \\ -B \\ B \\ -B \\ 0 \\ 0 \\ 1 \\ -1 \end{bmatrix} u(t) + \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & \\ 0 & \\ 0 & \\ 0 & \\ A_1 & \\ -A_1 & \\ A_2 & \\ -A_2 & \\ I & \\ -I & \\ 0 & \\ 0 & \end{bmatrix} x(t) + \begin{bmatrix} 5 \\ -\varepsilon \\ 0 \\ 0 \\ M \\ M \\ M \\ M \\ 0 \\ 0 \\ N \\ N \\ 1 \\ 1 \end{bmatrix} \end{cases},$$

where $B = [0\ 1]'$, A_1, A_2 are obtained from (9) by setting respectively $\alpha = \pi/3,\ -\pi/3$,

$$M = 4(1+\sqrt{3})[1\ 1]' + B, N \triangleq 5[1\ 1]',$$

and ε is a properly chosen small positive scalar.

4. Formulation of the Optimal Control Problem

Consider the PWA system (1) subject to hard input and state constraints

$$Ex(t) + Lu(t) \leq M \tag{10}$$

for $t \geq 0$, and denote by *constrained PWA system* (CPWA) the restriction of the PWA system (1) over the set of states and inputs defined by (10),

$$x(t+1) = A^i x(t) + B^i u(t) + f^i \quad \text{if} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \tilde{\mathscr{P}}^i \tag{11}$$

where $\{\tilde{\mathscr{P}}^i\}_{i=1}^s$ is the new polyhedral partition of the sets of the state + input space $\mathbb{R}^{n+m}$ obtained by intersecting the sets $\mathscr{P}^i$ in (1) with the polyhedron described by (10).

We assume the origin to be an equilibrium point of the PWA system (11) (i.e. $0 \in \tilde{\mathscr{P}}^i \Rightarrow f^i = 0$) and define the following cost function

$$J(U_N, x(0)) \triangleq \|Px(N)\|_p + \sum_{k=0}^{N-1} \|Qx(k)\|_p + \|Ru(k)\|_p. \tag{12}$$

We consider the constrained finite-time optimal control (CFTOC) problem

$$J^*(x(0)) \triangleq \min_{\{U_N\}} J(U_N, x(0)) \tag{13}$$

$$\text{subject to} \begin{cases} x(k+1) = A^i x(k) + B^i u(k) + f^i \\ \text{if } \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \tilde{\mathscr{P}}^i \\ x(N) \in \mathscr{X}_f \end{cases} \tag{14}$$

where the column vector $U_N \triangleq [u'(0), \dots, u'(N-1)]' \in \mathbb{R}^{m_c N} \times \{0,1\}^{m_l N}$, is the optimization vector, N is the time horizon and $\mathscr{X}_f$ is the terminal region. In (12), $\|Qx\|_p$ denotes the p-norm of the vector x weighted with the matrix Q, $p = 1, 2, \infty$. We will assume that $Q = Q' \succeq 0$, $R = R' \succ 0$, $P \succeq 0$, for $p = 2$, and that Q, R, P are full column rank matrices for $p = 1, \infty$

In general, the optimal control problem (12)–(14) may not have a minimizer for some feasible $x(0)$, as we have assumed above, but only an infimizer. We remark here that situations where the minimizer does not exist may lead to arbitrarily high sensititvity of the performance of the controlled system to model errors.

The minimizer $U_N^*(x(0))$ may not be unique for a given $x(0)$. From an engineering point of view it make sense to refer to the function $U_N^*(x(0))$ as one of the possible single-valued functions.

5. Properties of the Solution of the Optimal Control Problem

In the following, we need to distinguish between optimal control based on the 2-norm and optimal control based on the 1-norm or ∞-norm. We are interested in determining structural properties of the value function and the feedback control law, in particular, its functional form.

Theorem 1. The solution to the CFTOC problem (12)–(14) with $p = 2$ is a PWA state feedback control law of the form $u^*(k) = f_k(x(k))$, where

$$f_k(x(k)) = F_k^i x(k) + G_k^i \quad \text{if } x(k) \in \mathscr{R}_k^i, \tag{15}$$

where $\mathscr{R}_k^i$, $i = 1, \ldots, N_k$ is a partition of the set $\mathscr{X}_k$ of feasible states $x(k)$ and the closure $\bar{\mathscr{R}}_k^i$ of the sets $\mathscr{R}_k^i$ has the following form:

$$\bar{\mathscr{R}}_k^i \triangleq \{x | x' L_k^i(j) x + M_k^i(j) x \leq N_k^i(j),\\ j = 1, \ldots, n_k^i\}, \quad k = 0, \ldots, N-1.$$

Theorem 1 implies that the control law has a simple form locally but that the partition is defined by quadratic surfaces, which makes it complex. However, often most of the regions of the partition or even the whole partition turn out to be polyhedral.

Theorem 2. Assume that the optimizer $U_{N^*}(x(0))$ of CFTOC problem (12)–(14) with $p = 2$ is unique for all $x(0)$. Then the solution to the optimal control problem (12)–(14) is a state feedback control law of the form

$$u^*(x(k)) = F_k^i x(k) + G_k^i \\ \text{if } x(k) \in \mathscr{R}_k^i \; k = 0, \ldots, N-1 \tag{16}$$

where $\mathscr{R}_k^i$, $i = 1, \ldots, N_k^r$ is a polyhedral partition of the set $\mathscr{X}_k$ of feasible states $x(k)$.

The previous results can be extended to piecewise linear cost functions, i.e., cost functions based on the 1-norm or the ∞-norm.

Theorem 3. The solution to the CFTOC problem (12)–(14) with $p = 1, \infty$ is a state feedback control law of the form $u^*(k) = f_k(x(k))$

$$f_k(x(k)) = F_k^i x(k) + G_k^i \quad \text{if } x(k) \in \mathscr{R}_k^i \tag{17}$$

where $\mathscr{R}_k^i$, $i = 1, \ldots, N_k^r$ is a polyhedral partition of the set $\mathscr{X}_k$ of feasible states $x(k)$.

By comparing Theorem 1 and Theorem 3 it is clear that it is simpler to solve problems with performance indices based on 1 or ∞ norms. In this case, the solution is piecewise affine on polyhedra and one does not need to deal with nonconvex ellipsoidal regions as in the 2-norm case. However, often the use of linear norms has practical disadvantages. A satisfactory performance may only be achieved with long time-horizons with a consequent increase of complexity. Also performance may not depend smoothly on the weights used in the performance index, i.e., slight changes of the weights could lead to very different optimal trajectories, so that the tuning of the controller becomes difficult.

6. Computation of the Solution of the Optimal Control Problem

6.1. Computation of the Sequence of Optimal Control Inputs

The main idea is to translate problem (12)–(14) into a mixed-integer program that can be solved by using standard commercial software. The translation is immediate if the equivalent MLD representation of the PWA system is available. When the performance index is quadratic, the optimization problem can be cast as a mixed-integer quadratic program (MIQP) [7]. Similarly, 1-norm and ∞-norm performance indices lead to mixed-integer linear programming (MILP) problems. This approach does not provide the state feedback law (15) or (17) but only the optimal control sequence $U_N^*(x(0))$ for a fixed initial state.

6.2. Computation of the Optimal Feedback Control Law

The basic method for computing the control law is multiparametric programming which we introduce next.

6.2.1. Multiparametric Programming

Consider the nonlinear mathematical program dependent on a parameter appearing in the cost function and the constraints

$$J^*(x) = \inf_z f(z, x) \text{ subject to } \quad g(z, x) \leq 0 \quad (18)$$

where $z \in M \subseteq \mathbb{R}^s$ is the optimization variable, $x \in \mathbb{R}^n$ is the parameter, $f : \mathbb{R}^s \times \mathbb{R}^n \to \mathbb{R}$ is the cost function and $g : \mathbb{R}^s \times \mathbb{R}^n \to \mathbb{R}^{n_g}$ are the constraints.

A small perturbation of the parameter x in the mathematical program (18) can cause a variety of results, i.e., depending on the properties of the functions f and g the solution $z^*(x)$ of the mathematical program may vary smoothly or change abruptly as a function of x. The process of finding the optimizer $z^*(x)$ as a function of the parameter x is referred to as multiparametric programming. We denote by K^* the set of feasible parameters, i.e.,

$$K^* = \{x \in \mathbb{R}^n | \exists z \in \mathbb{R}^s, g(z, x) \leq 0\} \quad (19)$$

by $J^*(x)$ the real-valued function, which expresses the dependence on x of the minimum value of the objective function over K^*, and by $Z^*(x)$ the point-to-set map which expresses the dependence on x of the set of optimizers, i.e.,

$$Z^*(x) = \{z \in M | f(z, x) = J^*(x)\} \quad (20)$$

$J^*(x)$ will be referred to as optimal value function or simply value function, $Z^*(x)$ will be referred to as optimal set. If $Z^*(x)$ is a singleton for all x, then $z^*(x) \triangleq Z^*(x)$ will be called optimizer function.

Fiacco [(25, chapter 2)] provides conditions under which the solution of nonlinear multiparametric programs (18) is locally well behaved and establishes properties of the solution as a function of the parameters. Below we restrict our attention to some special classes of multiparametric programming.

Definition 3. (*mp-LP*) If $f(\cdot, \cdot)$ and $g(\cdot, \cdot)$ are linear, $z \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$ we call the problem (18) a multiparametric linear program (mp-LP).

Definition 4. (*mp-QP*). If $f(\cdot, \cdot)$ is quadratic, $g(\cdot, \cdot)$ is linear, $z \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$ we call the problem (18) a multiparametric linear program (mp-QP).

If some of the optimization variables are restricted to be integer, the problems are referred to as multiparametric mixed integer programs.

Definition 5. (*mp-MILP*). If $f(\cdot, \cdot)$ and $g(\cdot, \cdot)$ are linear, $z \in \mathbb{R}^{m_c} \times \{0, 1\}^{m_l}$ and $x \in \mathbb{R}^n$ we call the problem (18) a multiparametric mixed integer linear program (mp-MILP).

Definition 6. (*mp-MIQP*). If $f(\cdot, \cdot)$ is quadratic $g(\cdot, \cdot)$ is linear, $z \in \mathbb{R}^{m_c} \times \{0, 1\}^{m_l}$ and $x \in \mathbb{R}^n$ we call the problem (18) a multiparametric mixed integer linear program (mp-MIQP).

6.2.2. Computation via multiparametric Mixed Integer Programming

By generalizing the results for linear systems in [8] to hybrid systems, the state vector $x(0)$, which appears in the objective function and in the constraints (12)–(14), can be handled as a vector of parameters that perturbs the solution of the mathematical program. Then, for performance indices based on the ∞-norm or 1-norm, the optimization problem can be treated as a *multiparametric MILP* (mp-MILP), while for performance indices based on the 2-norm, the optimization problem can be treated as a *multiparametric MIQP* (mp-MIQP). Solving an mp-MILP (mp-MIQP) amounts to expressing the solution of the MILP (MIQP) as a function of the parameters $x(0)$.

In [1,21] two approaches were proposed for solving mp-MILP problems. In both methods, the authors use an mp-LP algorithm and a branch and bound strategy that avoids the enumeration of all combinations of 0–1 integer variables. A method for solving mp-MIQPs has appeared recently in [20].

The solution of the mp-MILP (mp-MIQP) provides the state feedback solution $u^*(k) = f_k(x(k))$ ((17) or (15)) of the problem (12)–(14) for $k = 0$ and it also provides the open loop optimal control laws $u^*(k)$ as function of the initial state, i.e., $u^*(k) = u^*(k)(x(0))$. The PWA state feedback optimal control law f_k: $x(k) \mapsto u^*(k)$ for $k = 1, \ldots, N$ is obtained by solving N mp-MILPs (mp-MIQPs) over a shorter horizon [12].

More details on multiparametric programming can be found in [17,26] for linear programs, in [9, 42, 44] for quadratic programs in [1,21] for mixed integer linear programs and in [12,15,20] for mixed integer quadratic programs.

6.2.3. Computation via Dynamic Programming

In his thesis [12], Borrelli showed how linear and quadratic parametric programming can be used to solve the Hamilton–Jacobi–Bellman equation associated with problems (12)–(14). The PWA solution (15) is computed proceeding backwards in time using two tools: simple linear or quadratic parametric programming (depending on the cost function used) and – for the quadratic cost function – a special technique to store the solution.

This procedure becomes prohibitive for a system with a large number of binary inputs or states and

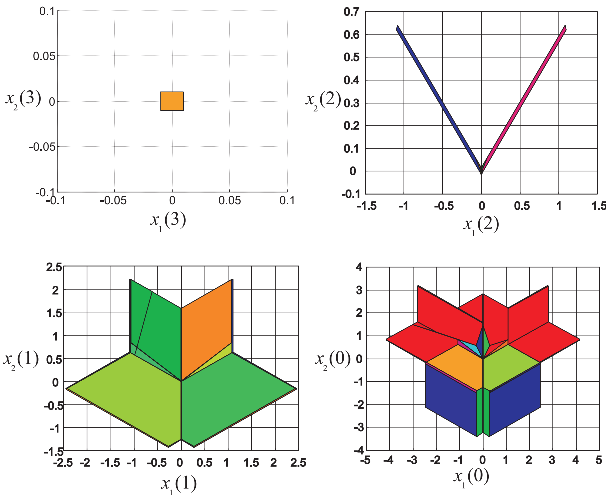


Fig. 1. State space partition for the time varying state feedback control law $u^*(x(k))$ for the Example 6.2.4. Each picture shows the control law at a particular time step $k = 0, \ldots, 3$. In each shaded region a different PWA state feedback law applies.

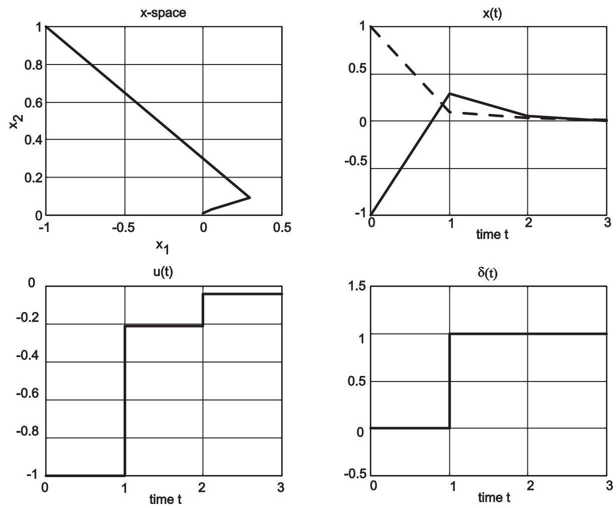


Fig. 2. Response of system described in 6.2.4 with controller shown in Fig. 1 for initial state $x(0) = [-1, 1]'$.

long horizons. In this case, multiparametric mixed-integer programming (or a combination of the two techniques) may be preferable. The multiparametric programming approach described in Section 6.2.2 and the dynamic programming approach have been compared in [2].

6.2.4. Example – Finite Time Optimal Control (∞-norm)

Consider the problem of steering the PWA system (9) to a small region around the origin in three steps while minimizing the cost function (12) with $p = \infty$. The finite-time constrained optimal control problem (12)–(14) is solved with $N = 3$, $P = 0$, $Q = 700I$, $R = 1$, and $\mathcal{X}_f = [-.0.01\ 0.01] \times [-0.01\ 0.01]$. The solution was computed with the algorithm presented in Section 6.2.3 in 71 s with Matlab 6.1 on a Pentium IV 2.2 GHz machine. The polyhedral regions corresponding to the state feedback solution $u^*(x(k))$; $k = 0, \ldots, 2$ in (17) are depicted in Fig. 1. The resulting optimal trajectories for the initial state $x(0) = [-1\ 1]'$ are shown in Fig. 2. The state feedback control law at time 0 comprises 106 polyhedral regions.

7. Case Studies

An infinite horizon controller can be obtained by implementing in a receding horizon fashion a finite-time optimal control law. In this case, the infinite time controller is simply obtained by repeatedly evaluating the PWA controller (15) for $k = 0$:

$$u(t) = u_0^*(x(t)) \quad \text{for } x(t) \in \mathcal{X}_0. \tag{21}$$

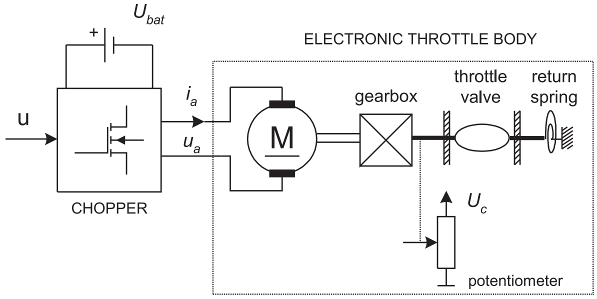
This approach was employed in all case studies described below.

7.1. Electronic Throttle [3]

In automotive applications an electronic throttle is used to control the inflow of air to the vehicle engine. The electronic throttle shown in Fig. 3 is a DC servo system, where the DC drive is supplied from a bipolar chopper. Motor shaft rotation is transmitted through a gearbox to the shaft with the throttle plate. Movement of the plate continues until the motor torque is balanced by the torque generated by the return spring which is attached to the plate's shaft.

Two nonlinearities in the throttle body make the control of the plate position a challenging task. One is friction in the gearbox transmission mechanism, and the other one is the so-called Limp-Home (LH) position nonlinearity that is a consequence of an embedded mechanical safety feature which guarantees a specific level of air inflow even in the case of total power failure.

All nonlinearities of the electronic throttle are PWA in continuous time (cf. [3]). It is reasonable to consider the following approach when constructing a discrete-time PWA model. We determine all possible combinations of affine parts of nonlinearities (5 from the friction model, and 3 from the LH model give 15 affine dynamics in total), and for each of those combinations we form an affine continuous description of the throttle system. Sampling each of the continuous



(a) Electronic throttle system

(b) Throttle valve

Fig. 3. The electronic throttle.

affine systems with a Zero Order Hold (ZOH) gives us an equivalent discrete-time representation. Combining them finally gives 15 discrete-time PWA dynamics, each of which is defined over a certain part of the state + input space. The state vector is $x=[\omega_m,\theta]$, where θ denotes position of the throttle valve, ω_m is the motor angular speed, and the manipulated variable u is the input voltage to the chopper (Fig. 3(a)). The sampling time was set to $T_s=10$ ms. The following constraints $\omega_m(k)\in[-100\ \ 100]$, $\theta(k)\in[13\ \ 90]$ and $u(k)\in[-5\ \ 5]$ where included in the discrete-time PWA model of the throttle. For more details see [3].

We designed a receding horizon controller based on the optimization problem (12)–(14) with

$$T=5,\ \ Q=\begin{bmatrix}0.01 & 0\\ 0 & 1\end{bmatrix},\ \ R=0.1,\ \ P=Q,$$
$$\mathcal{X}_f=[-0.2\ 24.95]\times[0.2\ 25.05],$$
$$x_e=[0\ 25]', u_e=1.098.$$

Thus, our goal is to open the throttle valve to the fixed reference of 25°. The corresponding optimal solution is composed of 568 regions in the state–space.

The optimal set point control scheme was applied to the real electronic throttle depicted in Fig. 3(b). Experiments were carried out with a Pentium III 1.7 GHz machine running MATLAB® 5.3 and using Real-TimeWorkshop® with an A/D-D/A card as a computer-process interface. Note that the optimal controller is applied in a receding horizon fashion.

In Fig. 4, we report the experimental results for a particular demanding task where the LH position $\theta_{\mathrm{LH}}=20°$ is crossed while opening the valve from 13.74° to a specified angle of 25°. The starting point $x(0)=[0\ 13.74]'$ is reached using a step change of the control input at $t=25$ s. The optimal controller action is turned on at $t=5$ s. We point out that only the position signal θ was measured, while ω_m was reconstructed from the position signal.

7.2. Multi-Object Adaptive Cruise Control [36]

Cruise control is a common and well known automotive driver assistance system in which the driver sets a reference speed and the engine is controlled so that this reference speed is maintained regardless of external loads such as wind, road slope or changing vehicle parameters. Adaptive Cruise Control (ACC) additionally takes into account the traffic in front of the car. In a multi-object adaptive cruise control problem the optimal acceleration of the driver's car is to be found respecting traffic rules, safety distances and driver intentions. The control objectives for a traffic scene depicted in Fig. 5 are

- to track the reference speed v_{ref};
- to respect safety distance $d_{\min}$ if a neighboring car is in the same lane; and
- not to overtake on the right-hand side of a neighboring car

while constraining acceleration and changes in acceleration and in deceleration. Loosely speaking, we would like to maintain a comfortable drive for the ego-car and prevent any obj-car from entering the shaded area in Fig. 5. The hybrid nature of the problem arises from the multiple objectives which include switches. The objective function is chosen to be quadratic. The optimal state-feedback control law is found by solving the underlying constrained finite time optimal control problem via dynamic programming.

Here, we report experimental results when the ego-car is driving in the right-most lane (neglecting the lateral dynamics). The finite-time constrained optimal control problem (12)–(14) was solved over a time horizon of 8.1 s with $v_{\mathrm{ref}}=30$ m/s and $d_{\min}=40$ m. The corresponding optimal solution is composed of 753 polyhedra in $\mathbb{R}^7$ (the dimension of the state space is 7).

The optimal state-feedback control law was tested on a research car Mercedes E430 (Fig. 6). Special interfaces to throttle and brakes, sensor fusion,

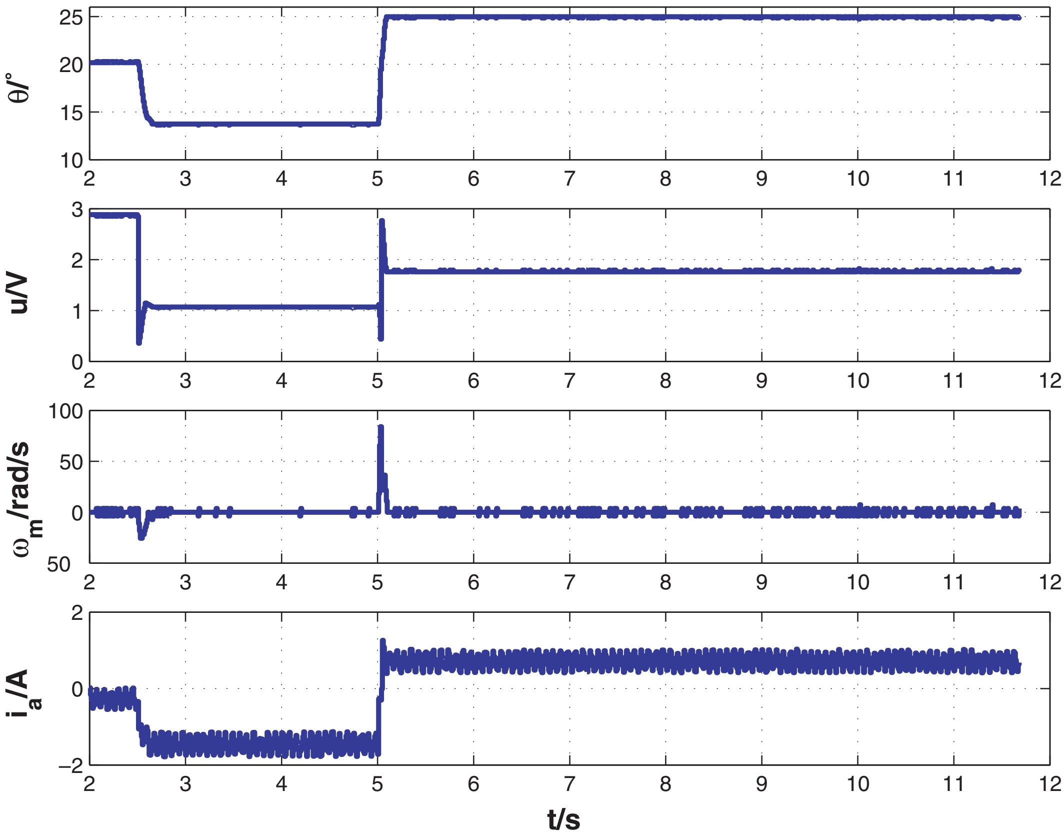


Fig. 4. Experimental results for a particulary demanding task where the LH position $\theta_{LH} = 20°$ is crossed while opening the throttle valve from 13.74° to a specified angle of 25°. The optimal controller action is turned on at $t = 5$ s.

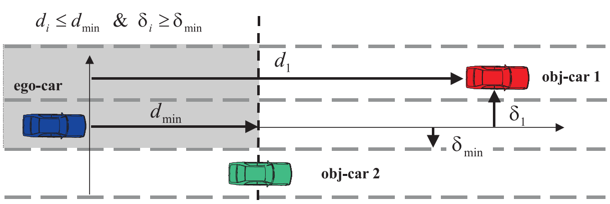


Fig. 5. Multi-object adaptive cruise control problem.

visualization and the ACC controller were running in a real-time environment with an 80 ms cycle time on an Intel Pentium4 1.4 GHz machine with 500 MByte RAM. In Fig. 7 experimental data from a test drive are shown. In Fig. 7(a) the distance d between ego-car and obj-car, velocity v_{ego} of ego-car and velocity v_{obj} of obj-car are reported. In Fig. 7(b) we see a corresponding controller action a_{ref} (reference acceleration) for ego-car, measured acceleration a_{ego} of ego-car and estimated acceleration a_{obj} of obj-car.

Initially ego-car is driving faster then obj-car ($v_{ego} = 26$ m/s, $v_{obj} = 24$ m/s, see Fig. 7(a)), and the distance between ego-car and obj-car $d = 28$ m is below the safety distance $d_{min} = 40$ m. As depicted in Fig. 7(b), controller reacts to such a situation by decelerating ego-car more and more strongly (e.g., hitting the brakes with an increasing force) until $t = 0.8$ s when the lower bound on the acceleration value $a_{obj} = -1$ m/s^2 is reached. Around $t = 2.5$ s both vehicles have the same velocity and from that moment on $v_{ego} < v_{obj}$ and the distance d between the vehicles is steadily increasing. From $t = 3.5$ s on brakes are gradually released, and around $t = 6$ s ego-car even begins accelerating. The reasoning is relatively simple. The closer we get to the safety distance the more important the tracking of the reference velocity becomes. However, since obj-car drives more slowly than the desired $v_{ref} = 30$ m/s, after $t = 11$ s ego-car tracks obj-car's speed while preserving the safety distance.

7.3. Other Case Studies

Cement Mill [27]. In a cement mill scheduling problem we have to decide, depending on the customers demand and energy prices, when to produce certain cement grades and on which mills. Due to the number

(a) Mercedes E430 (b) Look from the cockpit

Fig. 6. Experimental setup.

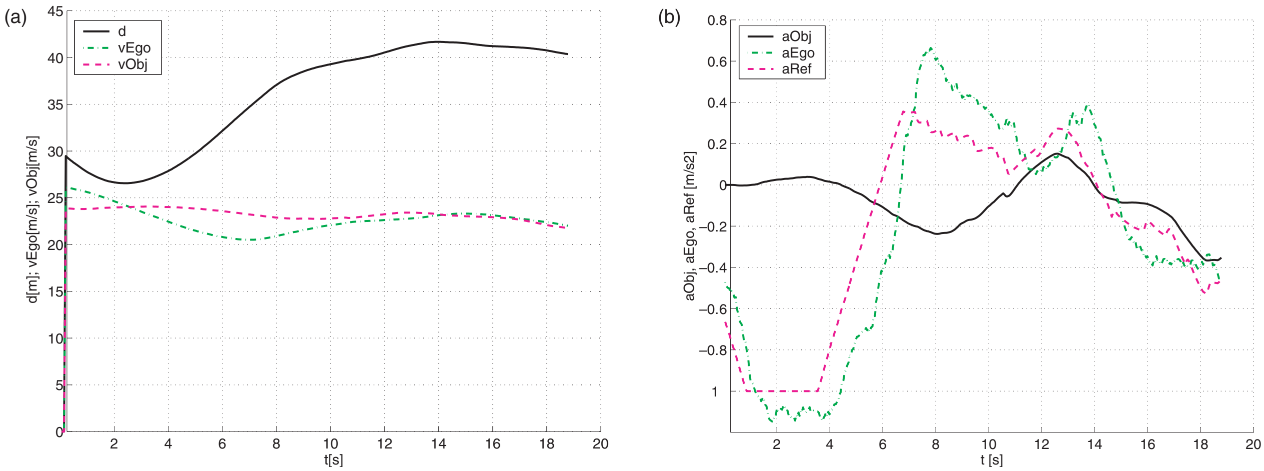


Fig. 7. Experimental results of the ACC system with implemented optimal state-feedback controller. After a transient ego-car manages to track the speed of obj-car while maintaining the safety distance $d_{\min} = 40\,\mathrm{m}$. (a) Measured distance between ego-car and obj-car d, and velocities v_{ego} and v_{obj}. (b) Manipulated variable a_{ref} and measured/estimated accelerations a_{ego} and a_{obj}.

of mills, grades, silos and the various operating constraints the problem is quite complex. The authors show how to achieve an economically optimal behavior by designing two model predictive control loops in a cascade: an "outer" one responsible for long term economic goals, and a faster "inner" one ensuring optimal reactions to deviations from the original plan. The hybrid characteristic arise from discrete values of decision variables, finite state machine models for the mills and operational constraints. The case of 2 mills, 2 silos, and 5 cement grades and a sampling time of one hour was considered. Simulation results obtained by solving an MILP problem are reported. The procedure has been implemented on a cement mill in Switzerland and is in test phase.

Co-generation Power Plant [22]. The authors consider optimal control of a power plant that comprises a steam turbine and a gas turbine. The possibility of turning on/off the gas and steam turbine, the operating constraints (minimum up and down times) and the different types of start-up of the turbines characterize the hybrid behavior of a combined cycle power plant. For various prediction horizons $N \in \{1, \ldots, 24\}$, at every time instant an MILP problem with $(46 \cdot N - 4)$ optimization variables ($27 \cdot N$ of which are integer), and $(119 \cdot N - 8)$ mixed-integer linear constraints was solved. The computation times (average and worst cases) needed for solving the MILPs, on a Pentium II-400 (running Matlab 5.3 for building the matrices defining the MILP and running CPLEX for solving it) are well below the sampling time of one hour.

Traction control [16]. Traction controllers are used to improve a driver's ability to control a vehicle under adverse external conditions such as wet or icy roads. The objective of the controller is to maximize the tractive torque while preserving the stability of the system. The wheel slip, i.e., the difference between the normalized vehicle speed and the speed of the wheel is used as the controlled variable. The complete MLD model, capturing all the hybrid features of the plant involves 6 state variables, 1 δ variable, 3 z variables and 1 continuous control variable. There are 14 inequalities stemming from the representation of the δ and z variables in the matrices E_{1-5} of MLD description. Rather than solving MILP problems on-line at all times (as was the case in the two previously mentioned applications), the authors were using an mp-MILP solver to compute off-line the PWA controller (15) for all the states $x(0)$ within a given polyhedral set. The controller was implemented on a prototype vehicle and performed satisfactorily [12].

Active vibration suppression [39]. Today, smart materials are used extensively to suppress vibration of mechanical structures. The authors describe active vibration suppression using shunted piezoelectric materials (PZTs). These PZTs convert mechanical vibration energy into electrical energy, which is thereafter dissipated by a passive electronic shunt circuit. Shunts with internal switches may lead to a better damping performance than shunts that use only resistors. Furthermore, shunts with internal switches remain simple as the switches can be realized with MOSFETs. The mechanical structure together with switched PZTs can be modeled as a PWA system with $s = 2$ dynamics, the state vector $x \in \mathbb{R}^3$, and one integer input $u \in \{0, 1\}$ that specifies if the shunt is switched off or switched on. It is shown in simulations that the solution to the underlining optimal control problem for a prediction horizon of $N = 17$ yields a superior state feedback control law (i.e., better damping) than previously used ad-hoc switching rules. The system is currently undergoing experimental testing.

Among other applications we mention control of a benchmark multi-tank system [35], a gas supply system [7], a gear shift operation on automotive vehicles [45], a direct injection stratified charge engine [6], integrated management of a power-train [4], voltage collapse in power systems [28] and direct torque control of three-phase symmetric induction motors [40].

8. Conclusions

The structure and computation of optimal control laws for hybrid systems were discussed and various successful applications were shown. The results are very encouraging but the research in this area is at the beginning. In particular, there is significant potential to improve the algorithms in terms of their numerical properties as well as their speed. As an example, the reader is referred to the new technique for computing the infinite time linear quadratic regulator for constrained systems [29]. From the presented analysis it is clear that the proposed optimal control law for hybrid systems is inherently complex and, therefore, the techniques for finding these control laws *exactly must* be computationally demanding. To deal with this issue various approaches to reduce the complexity (representation of the control law [13], model employed for the optimization, decomposition of the optimization problem) are investigated.

Note: All technical reports AUTxx–xx from the Automatic Control Laboratory are available from the web site http://control.ee.ethz.ch.

References

1. Acevedo J, Pistikopoulos EN. A multiparametric programming approach for linear process engineering problems under uncertainty. Ind Eng Chem Res 1997; 36: 717–728
2. Baotic M, Christophersen FJ, Morari M. A new algorithm for constrained finite time optimal control of hybrid systems with a linear performance index. Technical Report AUT02-18, Automatic Control Laboratory, ETH Zurich, Switzerland, December 2002
3. Baotic M, Vasak M, Morari M, Peric N. Hybrid theory based optimal control of electronic throttle. In Proc. American Control Conf., Denver, Colorado, USA, June 2003
4. Bemporad A, Borodani P, Mannelli M. Hybrid control of an automotive robotized gearbox for reduction of consumptions and emissions. In: Maler O. and Pnueli A. (eds). Hybrid Systems: Computation and Control, HSCC 2003, vol 2623. Lecture Notes in Computer Science, Springer Verlag, Berling, 2003, pp 81–96
5. Bemporad A, Ferrari-Trecate G, Morari M. Observability and controllability of piecewise affine and hybrid systems. IEEE Trans. Autom Control 2000; 45(10): 1864–1876
6. Bemporad A, Giorgetti N, Kolmanovsky IV, Hrovat D. Hybrid modeling and control of a direct injection stratified charge engine. In: Symposium on Advanced Automotive Technologies, ASME International Mechanical Engineering Congress and Exposition, New Orleans, Louisiana, November 2002
7. Bemporad A, Morari M. Control of systems integrating logic, dynamics, and constraints. Automatica 1999; 35(3): 407–427
8. Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit linear quadratic regulator for constrained systems. In: Proc. Am. Control Conf., Chicago, IL, USA, 2000, pp 872–876
9. Bemporad A, Morari M, Dua V, Pistikopoulos EN. The explicit linear quadratic regulator for constrained systems. Automatica 2002; 38(1): 3–20
10. Biegler LT, Grossmann IE. Retrospective on optimization. Comput Chem Eng 2003, in press
11. Bixby RE, Fenelon M, Gu Z, Rothberg E, Wunderling R. MIP: Theory and practice: Closing

the gap. In: Powell MJD and Scholtes S, (eds). System Modelling and Optimization, vol 174 IFIP Conference Proceedings, Kluwer, 2000, pp. 19–50
12. Borrelli F. Constrained Optimal Control of Linear and Hybrid Systems, vol 290 Lecture Notes in Control and Information Sciences, Springer, Berlin, 2003, in press
13. Borrelli F, Baotic M, Bemporad A, Morari M. Efficient on-line computation of constrained optimal control. In: Proc. 40th IEEE Conf. on Decision and Control, December 2001
14. Borrelli F, Baotic M, Bemporad A, Morari M. Constrained optimal control of discrete-time linear hybrid systems. Technical Report AUT03-05, Automatic Control Laboratory, ETH Zurich, Switzerland, 2003
15. Borrelli F, Baotic M, Bemporad A, Morari M. An efficient algorithm for computing the state feedback optimal control law for discrete time hybrid systems. In: Proc. American Control Conf., Denver, Colorado, USA, June 2003
16. Borrelli F, Bemporad A, Fodor M, Hrovat D. A hybrid approach to traction control. In: Sangiovanni-Vincentelli A and Di Benedetto MD (eds), Hybrid Systems: Computation and Control, vol 2034 Lecture Notes in Computer Science, Springer Verlag, Berlin, 2001, pp 162–174
17. Borrelli F, Bemporad A, Morari M. A geometric algorithm for multi-parametric linear programming. J Optim Theory Appl 2003, in press
18. De Schutter B, De Moor B. The extended linear complementarity problem and the modeling and analysis of hybrid systems. In: Antsaklis P, Kohn W, Lemmon M, Nerode A, and Sastry S. (eds) Hybrid Systems V, vol 1567, Lecture Notes in Computer Science, Springer, Berlin, 1999, pp 70–85
19. De Schutter B, van den Boom T. Model predictive control for max-plus-linear systems. In: Proceedings of the American Control Conference, Chicago, IL, June 2000, pp 4046–4050
20. Dua V, Bozinis NA, Pistikopoulos EN. A multiparametric programming approach for mixed-integer quadratic engineering problems. Comput Chem Eng 2002; 26(4–5): 715–733
21. Dua V, Pistikopoulos EN. An algorithm for the solution of multiparametric mixed integer linear programming problems. Ann Oper Res 2000; 99: 123–139
22. Ferrari-Trecate G, Gallestey E, Stothert A, Letizia P, Spedicato M, Morari M, Antoine M. Modeling and control of co-generation power plants: A hybrid system approach. In: Tomlin CJ and Greenstreet MR (eds). Proceedings of the 5th International Workshop on Hybrid Systems: Computation and Control, Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2002
23. Ferrari-Trecate G, Mignone D, Morari M. Moving horizon estimation for hybrid systems. IEEE Trans Autom Control 2002; 47(10): 1663–1676
24. Ferrari-Trecate G, Muselli M, Liberati D, Morari M. A clustering technique for the identification of piecewise affine systems. Automatica 2003; 39(2)
25. Fiacco AV. Introduction to sensitivity and stability analysis in nonlinear programming. Academic Press, London, UK, 1983
26. Gal T. Postoptimal analyses, parametric programming, and related topics. 2nd edn. de Gruyter, Berlin, 1995
27. Gallestey E, Stothert A, Castagnoli D, Ferrari-Trecate G, Morari M. Using model predictive control and hybrid systems for optimal scheduling of industrial processes. Automatisierungstechnik, 2003, in press
28. Geyer T, Larsson M, Morari M. Hybrid control of voltage collapse in power systems. Technical Report AUT02-12, Automatic Control Lab, ETH Zurich, 2002
29. Grieder P, Borrelli F, Torrisi FD, Morari M. Computation of the constrained infinite time linear quadratic regulator. In Proceedings of the American Control Conference Denver, Colorado, USA, June 2003
30. Grimm G, Postlethwaite I, Teel AR, Turner MC, Zaccarian L. Linear matrix inequalities for full and reduced order anti-windup synthesis. In: Proceedings of the American Control Conference, Arlington, Virginia, USA, 2001, pp 4134–4139
31. Heemels WPMH. Linear complementarity systems: a study in hybrid dynamics. PhD thesis, Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands, 1999
32. Heemels WPMH, Schumacher JM, Weiland S. Linear complementarity systems. SIAM J Appl Math 2000; 60(4): 1234–1269
33. Heemels WPMH, De Schutter B, Bemporad A. Equivalence of hybrid dynamical models. Automatica 2001; 37(7): 1085–1091
34. Kothare MV, Campo PJ, Morari M, Nett CN. A unified framework for the study of anti-windup designs. Automatica 1994; 30(12): 1869–1883
35. Mignone D. Control and Estimation of Hybrid systems via Mathematical Optimization. Dr. sc. tech. thesis, Automatic Control Labotatory – ETH, Zurich, 2002, http://control.ee.ethz.ch
36. Möbus R, Baotic M, Morari M. Multi-object adaptive cruise control. In: Maler O, Pnueli A, (eds). Hybrid Systems: Computation and Control, HSCC 2003, vol 2623, Lecture Notes in Computer Science, Springer Verlag, Berlin, 2003, pp 359–374
37. Mulder EF, Kothare MV, Morari M. Multivariable anti-windup controller synthesis using linear matrix inequalities. Automatica 2001; 37(9): 1407–1416
38. Nemhauser GL, Wolsey LA. Integer and Combinatorial Optimization. Wiley, New York, 1988
39. Niederberger D, Baotic M, Morari M. Active vibration suppression using switched PZTs. Technical Report AUT03-06, Automatic Control Lab, ETH Zurich, 2003
40. Papafotiou G, Geyer T, Morari M. Optimal direct torque control of three-phase symmetric induction motors. Technical Report AUT03-07, Automatic Control Lab, ETH Zurich, 2003
41. Raman R, Grossmann IE. Relation between MILP modeling and logical inference for chemical process synthesis. Comput Chem Eng 1991; 15(2): 73–84
42. Seron MM, DeDoná JA, Goodwin GC. Global analytical model predictive control with input constraints. In: Proceedings of the 39th IEEE Conference on Decision and Control, 154–159, Chicago, IL, 2000
43. Sontag ED. Nonlinear regulation: The piecewise linear approach. IEEE Trans. Autom Control 1981; 26(2): 346–358
44. Tøndel P, Johansen TA, Bemporad A. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. In: Proceedings of the 40th

IEEE Conference on Decision and Control, Orlando, Florida, 2001
45. Torrisi FD, Bemporad A. HYSDEL – a tool for generating computational hybrid models. Technical Report AUT02-03, ETH Zurich, 2002. IEEE Trans Control Systems Technol. Submitted for publication
46. Torrisi FD, Bemporad A, Mignone D. HYSDEL – A language for describing hybrid systems. Technical Report AUT00-03, ETH Zurich, 2000, http://control.ethz.ch/~hybrid/hysdel
47. Tyler ML, Morari M. Propositional logic in control and monitoring problems. Automatica 1999; 35(4): 565–582
48. van der Schaft AJ, Schumacher JM. Complementarity modelling of hybrid systems. IEEE Trans. Autom Control 1998; 43: 483–490
49. Williams HP. Model building in Mathematical programming 3rd edn. John Wiley & Sons, New York, 1993