# HYBRID SYSTEMS MODELING, PARAMETRIC PROGRAMMING, AND MODEL PREDICTIVE CONTROL - IMPACT ON PROCESS OPERATIONS

Vivek Dua[1], Efstratios N. Pistikopoulos[2] and Manfred Morari[3]
[1] University College London, U.K. (v.dua@ucl.ac.uk)
[2]Imperial College London, U.K. (e.pistikopoulos@imperial.ac.uk)
[3]ETH Zurich, Switzerland (morari@control.ee.ethz.ch)

**Abstract.**
Process operations often involve transitions between operating modes, which are typically described by their own dynamic models, constraints and specifications. Developing a formal theoretical framework and efficient computational tools to deal with such operational issues pose a formidable challenge.

In the paper, we give an overview of recent key developments in this direction, involving (i) modeling of hybrid systems, exhibiting both continuous and discrete dynamics, to rigorously represent complex operations, and (ii) parametric programming mathematical and computational tools, to design optimal feedback control laws for constrained linear dynamic systems, in the context of model predictive control principles. Applications to a wide range of industrial problems will be discussed.

*Keywords*
Hybrid systems, mixed-integer modeling and optimization, parametric programming, model predictive control.

## Introduction

Hybrid systems can be defined as systems comprising a number of continuous subsystems that are connected by logical or discrete switching (Branicky *et al.*, 1998). Each subsystem is governed by a unique set of differential and/or algebraic equations. Hybrid systems have been studied by mathematicians, biologists, computer scientists and engineers amongst others. This has resulted in a large number of publications and journal issues that are especially devoted to hybrid systems (Automatica 35(3) 1999; IEEE Transactions on Automatic Control 43(4) 1998).

A number of frameworks for modelling hybrid system problems have been proposed. While a State-Transition Network (STN) approach for modelling is presented in Pantelides (2001), a mixed-integer framework is described in Raman and Grossmann (1991) and subsequent works (Biegler and Grossmann, 2004) and Bemporad and Morari (1999*a*). In this paper we focus on piecewise affine (PWA) systems (Morari *et al.*, 2003). PWA systems are defined by partitioning the state and input space into polyhedral regions and associating with each region a different linear state update equation

$$x(t+1) = A^i x(t) + B^i u(t) + f^i \qquad (1)$$

$$\text{if} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in P^i$$

where $i = 1, ..., s$, $x \in \Re^{n_c} \times \{0,1\}^{n_l}, u \in \Re^{m_c} \times \{0,1\}^{m_l}, \{P_i\}_{i=1}^{s}$ is a polyhedral partition of the set of the state and iput space $P \subset \Re^{n+m}, n \triangleq n_c + n_l, m \triangleq m_c + m_l$. $P$ is assumed to be closed and bounded and $x_c \in R^{n_c}$ and $u_c \in \Re^{m_c}$ denote the continuous components of the state and input vector, respectively; $x_l \in \{0,1\}^{n_l}$ and $u_l \in \{0,1\}^{m_l}$ similarly denote the binary components.

PWA systems represent a modelling environment for wide variety of hybrid systems: Electronic Throttle (Baotic *et al.*, 2002), Multi-Object Adaptive Cruise Control (Möbus *et al.*, 2003), Cement Mill (Gallestey *et al.*, 2003), Co-generation Power Plant (Ferrari-Trecate *et al.*, 2002*b*), Traction Control (Borrelli *et al.*, 2001) and Active Vibration Suppression (Niederberger *et al.*, 2003). Modeling and control of a co-generation power plant is discussed in detail (Ferrari-Trecate *et al.*, 2004) in this paper.

Note that PWA models are not suitable for recasting analysis/synthesis problems into more compact optimization problems. The Mixed Logical Dynamical (MLD) framework (Bemporad and Morari, 1999*a*) described in the next section can be used to recast hybrid dynamical optimization problems into mixed-integer linear (MILP) and quadratic programs (MIQP). A parametric programming algorithm that avoids the on-line solution of MILP is also described. Note that other optimization frameworks, such as those based upon deriving necessary conditions for optimal trajectory within the framework of the Pontryagin's maximum principle (Piccoli, 1999; Riedinger *et al.*, 1999; Sussmann, 1999) are not discussed.

# Hybrid Systems Modeling

The derivation of the Mixed Logical Dynamical (MLD) form of a hybrid system involves basically three steps (Bemporad and Morari, 1999*a*). The first one is to associate with a statement $S$, that can be either true or false, a binary variable $\delta \in \{0,1\}$ that is 1 if and only if the statement holds true. Then, the combination of elementary statements $S_1, ..., S_q$ into a compound statement via the boolean operators AND $(\wedge)$ , OR $(\vee)$ , NOT $(\sim)$ can be represented as linear inequalities over the corresponding binary variables $\delta_i$, $i = 1, ..., q$. The inequalities stemming from the compound statements are reported in Table 1. As an example consider P3, which says that the statement $S_1 \vee S_2$ holds true if and only if $\delta_1$ and $\delta_2$ sum up at least to 1.

A special statement is given by the condition $a^T x \leq 0$, where $x \in X \subseteq R^n$ is a continuous variable and $X$ is a compact set. If one defines $m$ and $M$ as lower and upper bounds on $a^T x$ respectively, the inequalities in P9 assign the value $\delta = 1$ if and only if the value of $a^T x$ satisfies the threshold condition. Note that in P7 and P9, $\varepsilon > 0$ is a small tolerance (usually close to the machine precision) introduced to replace the strict inequalities by non-strict ones.

The second step is to represent the product between linear functions and logic variables by introducing an auxiliary variable $z = \delta a^T x$. Equivalently, $z$ is uniquely specified through the mixed integer linear inequalities in P10.

The third step is to include binary and auxiliary variables in an LTI discrete-time dynamic system in order to describe in a unified model the evolution of the continuous and logic components of the system. The general MLD form of a hybrid system is (Bemporad and Morari, 1999*a*)

$$x(t+1) = Ax(t) + B_1 u(t) + B_2 \delta(t) + B_3 z(t) \quad (2)$$

$$y(t) = Cx(t) + D_1 u(t) + D_2 \delta(t) + D_3 z(t) \quad (3)$$

$$E_2 \delta(t) + E_3 z(t) \leq E_1 u(t) + E_4 x(t) + E_5 \quad (4)$$

where $x = \begin{bmatrix} x_c^T & x_l^T \end{bmatrix}^T \in R^{n_c} \times \{0,1\}^{n_l}$ are the continuous and binary states, $u = \begin{bmatrix} u_c^T & u_l^T \end{bmatrix}^T \in R^{m_c} \times \{0,1\}^{m_l}$ are the inputs, $y = \begin{bmatrix} y_c^T & y_l^T \end{bmatrix}^T \in R^{p_c} \times \{0,1\}^{p_l}$ the outputs, and $\delta \in \{0,1\}^{r_l}$, $z \in R^{r_c}$ represent auxiliary binary and continuous variables respectively. All constraints on the states, the inputs, the $z$ and $\delta$ variables are summarized in the inequalities (4). Note that, although the description (2)-(3)-(4) seems to be linear, nonlinearity is hidden in the integrality constraints over the binary variables. MLD systems are a versatile framework to model various classes of systems. For a detailed description of such capabilities we defer the reader to (Bemporad and Morari, 1999*a*; Bemporad *et al.*, 2000*b*).

The discrete-time formulation of the MLD system allows developing numerically tractable schemes for solving complex problems, such as stability (Ferrari-Trecate *et al.*, 2002*c*; Mignone *et al.*, 2000), state estimation and fault detection (Ferrari-Trecate *et al.*, 2002*a*), formal verification of hybrid system (Bemporad and Morari, 1999*b*), and control (Bemporad and Morari, 1999*a*). In particular, MLD models were proven successful for recasting hybrid dynamic optimization problems into mixed-integer linear and quadratic programs solvable via branch and bound techniques.

The procedure for representing a hybrid system in the MLD form (2)-(4) can be automated. For this purpose, the compiler HYSDEL (HYbrid System DEscription Language), that generates the matrices of the MLD model starting from a high-level description of the dynamic and logic of the system, was developed at ETH Zürich (Torrisi *et al.*, 2000).

| | relation | logic | mixed integer inequalities |
|---|---|---|---|
| P1 | AND $(\wedge)$ | $S_1 \wedge S_2$ | $\delta_1 = 1$<br>$\delta_2 = 1$ |
| P2 | | $S_3 \Leftrightarrow (S_1 \wedge S_2)$ | $-\delta_1 + \delta_3 \leq 0$<br>$-\delta_2 + \delta_3 \leq 0$<br>$\delta_1 + \delta_2 - \delta_3 \leq 1$ |
| P3 | OR $(\vee)$ | $S_1 \vee S_2$ | $\delta_1 + \delta_2 \geq 1$ |
| P4 | NOT $(\sim)$ | $\sim S_1$ | $\delta_1 = 0$ |
| P5 | IMPLY $(\Rightarrow)$ | $S_1 \Rightarrow S_2$ | $\delta_1 - \delta_2 \leq 0$ |
| P6 | IFF $(\Leftrightarrow)$ | $S_1 \Leftrightarrow S_2$ | $\delta_1 - \delta_2 = 0$ |
| P7 | | $\left[a^T x \leq 0\right] \Rightarrow [\delta = 1]$ | $a^T x \geq \varepsilon + (m - \varepsilon)\,\delta$ |
| P8 | | $[\delta = 1] \Rightarrow \left[a^T x \leq 0\right]$ | $a^T x \leq M - M\delta$ |
| P9 | | $\left[a^T x \leq 0\right] \Leftrightarrow [\delta = 1]$ | $a^T x \leq M - M\delta$<br>$a^T x \geq \varepsilon + (m - \varepsilon)\,\delta$ |
| P10 | Mixed product | $z = \delta \cdot a^T x$ | $z \leq M\delta$<br>$z \geq m\delta$<br>$z \leq a^T x - m\,(1 - \delta)$<br>$z \geq a^T x - M\,(1 - \delta)$ |

Table 1: Basic conversion of logic relations into mixed integer inequalities.

# Predictive Control of MLD Systems

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 optimizing 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 set of measurements is taken and a new sequence is evaluated based on the current state.

Let $t$ be the current time, and $x(t)$ the current state. Consider the following optimal control problem

$$\min_{\{v_0^{T-1}\}} J(v_0^{T-1}, x(t)) \triangleq \sum_{k=0}^{T-1} \|v(k)\|_{Q_1}^2 + \|\delta(k|t)\|_{Q_2}^2 + \|z(k|t)\|_{Q_3}^2 + \|x(k|t)\|_{Q_4}^2 + \|y(k|t)\|_{Q_5}^2 \tag{5}$$

$$\text{subj. to} \begin{cases} x(k+1|t) = Ax(k|t) + B_1 v(k) + \\ B_2\delta(k|t) + B_3 z(k|t) \\ y(k|t) = Cx(k|t) + D_1 v(k) + \\ D_2\delta(k|t) + D_3 z(k|t) \\ E_2\delta(k|t) + E_3 z(k|t) \leq E_1 v(k) + \\ E_4 x(k|t) + E_5 \end{cases} \tag{6}$$

where $v_0^{T-1} \triangleq [v'(0), ..., v'(T-1)]'$, $Q_1 = Q_1' > 0$, $Q_2 = Q_2' \geq 0$, $Q_3 = Q_3' \geq 0$, $Q_4 = Q_4' > 0$ and $Q_5 = Q_5' \geq 0$. $x(k|t) \triangleq x(t+k, x(t), v_0^{k-1})$ is the state predicted at time $t+k$ resulting from the input $u(t+k) = v(k)$ to (2-4) starting from $x(0|t) = x(t)$. $\delta(k|t)$, $z(k|t)$ and $y(k|t)$ 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

$$u(t) = v_t^*(0), \tag{7}$$

disregard the subsequent optimal inputs $v_t^*(1), \dots, v_t^*(T-1)$, and repeat the whole optimization procedure at time $t+1$. Note that (5-6) is an MIQP. This problem can be formulated as an MILP if 1 norm instead of the 2 norm is considered in the objective function. The repetitive solution of the MIQP or MILP can be avoided by formulating (5-6) as a multiparametric program and solving it to obtain the control variables as set of explicit functions of the current state of the system and the regions in the space of the state variables where the explicit functions remain valid (Bemporad *et al.*, 2000*a*; Sakizlis *et al.*, 2002*a*). This is achieved by recasting (5-6) in a compact form as follows:

$$\begin{aligned} &\min_{\pi_c,\pi_d} \pi_c^T Q_c \pi_c + \phi^T \pi_d \\ \text{s.t.}\quad & G_c\pi_c + G_d\pi_d \leq S + Fx(t) \end{aligned} \tag{8}$$

where $\pi_c$ and $\pi_d$ are continuous and discrete variables of (5-6), $Q_c, \phi^T, G_c, G_d, S, F$ are constant matrices and vectors of appropriate dimensions and $Q_c$ is symmetric and positive definite. $x(t)$ is the state at the current time $t$. The objective is to obtain $\pi_c$ and $\pi_d$ as a function of $x(t)$ without exhaustively enumerating the entire space of $x(t)$. This can be achieved by using parametric programming, as described in the next section.

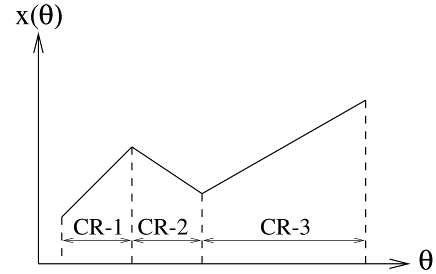


Figure 1: Parametric Optimization

# Parametric Programming and Control for Hybrid Systems

Consider the following multiparametric program:

$$\begin{aligned} & z(\theta) = \min_x \ f(x,\theta) \\ \text{s.t.}\quad & g_i(x,\theta) \leq 0, \quad \forall\ i = 1,...,p \\ & h_j(x,\theta) = 0, \quad \forall\ j = 1,...,q \\ & x \in X \subseteq \Re^n \\ & \theta \in \Theta \subseteq \Re^m \end{aligned} \tag{9}$$

Note that in an optimization framework $f$ is the performance criterion to be minimized, $g \leq 0$ and $h = 0$ are the constraints, $x$ is the vector of optimization variables and $\theta$ is the vector of parameters. In an optimal control framework, $x$ corresponds to the vector of control variables, $\pi_c$ and $\pi_d$ in (8) and $\theta$ corresponds to the vector of the current state of the system, $x(t)$ in (8). Parametric programming provides $x(\theta) = \arg\{\min_x f(x,\theta)$, subject to constraints in (9) }. Note that

$$x(\theta) = \begin{cases} x^1(\theta) & \text{if} \quad \theta \in CR^1 \\ x^2(\theta) & \text{if} \quad \theta \in CR^2 \\ & \vdots \\ x^i(\theta) & \text{if} \quad \theta \in CR^i \\ & \vdots \\ x^N(\theta) & \text{if} \quad \theta \in CR^N \end{cases}$$

such that $CR^i \cap CR^j = \emptyset, i \neq j, \forall i,j = 1,...,N$ and $CR^i \subseteq \Theta, \forall i = 1,...,N$. A $CR^i$ is known as a Critical Region. For the case when $f, g$ and $h$ are linear and separable in $x$ and $\theta$, the CRs are polyhedra and each CR corresponds to a uniqiue set of active constraints (Dua *et al.*, 2002). $x(\theta)$ when substituted into $f(x,\theta)$ provides the optimal objective function value, $z(\theta)$, as a function of $\theta$. See Figure 1, where $x(\theta)$ is plotted as a function of $\theta$.

The procedure for obtaining $x^i(\theta)$ and $CR^i$ depends upon whether $f$, $g$ and $h$ are linear, quadratic, nonlinear, convex, differentiable, or not, and also whether $x$ is vector of continuous or mixed - continuous and integer- variables (Dua and Pistikopoulos, 2000; Dua et al., 2002; Dua and Pistikopoulos, 1999; Dua et al., 2003; Sakizlis et al.,

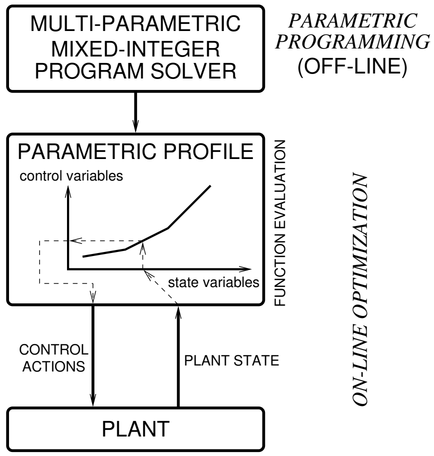


Figure 2: On-line optimization via off-line parametric programming

2002b). Recently algorithms for the case when (9) involves (i) differential and algebraic equations (Sakizlis et al., 2002a) and (ii) uncertain parameters (Sakizlis et al., 2004) have also been proposed. In the next section an algorithm for Multiparametric Mixed Integer Linear Programs (mp-MILP) is described. This reduces on-line hybrid system control problem to a function evaluation problem (Figure 2).

### Multiparametric Mixed-Integer Linear Programming

Consider a multiparametric Mixed Integer Linear Programming (mp-MILP) problem of the following form:

$$\begin{aligned} & z(\theta) = \min_{x,y} c^T x + d^T y \\ \text{s.t.} \quad & Ax + Ey \leq b + F\theta \\ & \theta_k^L \leq \theta_k \leq \theta_k^U, \ \ k = 1, \dots, m \\ & x \in \Re^n \\ & y \in \{0,1\}^l \\ & \theta \in \Theta \subseteq \Re^m, \end{aligned} \tag{10}$$

where $x$ is a vector of continuous variables, $y$ is vector of 0-1 binary variables, $\theta$ is a vector of parameters, $\theta^L$ and $\theta^U$ are the vectors of lower and upper bounds on $\theta$, $A$ is a $(p \times n)$ constant matrix, $E$ is a $(p \times l)$ constant matrix, $F$ is a $(p \times m)$ constant matrix, $b, c$ and $d$ are constant vectors of dimension $p$, $n$ and $l$ respectively.

Note that Acevedo and Pistikopoulos (1997) presented an algorithm for solving (10) based upon B&B principles. The algorithm described in this section is based upon decomposing (10) into an mp-LP and an MILP subproblem. The solution of the mp-LP, which is obtained by fixing the vector of binary variables, provides a parametric upper bound, whereas the solution of the MILP, which is obtained by treating $\theta$ as a free variable, provides a new integer vector.

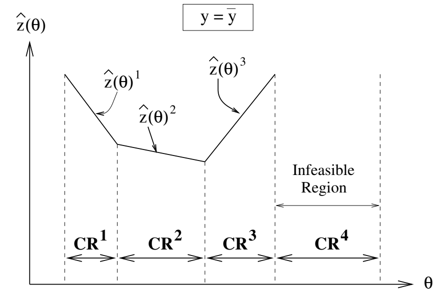


Figure 3: Parametric Solution - Multiparametric LP Subproblem for $y = \bar{y}$

The parametric solutions corresponding to two different integer solutions are then compared, using a procedure proposed by Acevedo and Pistikopoulos (1997), in order to keep as tight upper bounds as possible. The steps of the algorithm are described in detail in the following sections.

### Initialization

An initial feasible $y$ is obtained by solving the following MILP:

$$\begin{aligned} & z = \min_{x,y,\theta} c^T x + d^T y \\ \text{s.t.} \quad & Ax + Ey \leq b + F\theta \\ & \theta_k^L \leq \theta_k \leq \theta_k^U, \ \ k = 1, \dots, m \\ & x \in \Re^n \\ & y \in \{0,1\}^l, \end{aligned} \tag{11}$$

where $\theta$ is treated as a variable to find a starting feasible solution. Let the solution of (11) be given by $y = \bar{y}$.

### Multiparametric LP Subproblem

Fix $y = \bar{y}$ in (10) to obtain a multiparametric LP problem of the following form:

$$\begin{aligned} & \hat{z}(\theta) = \min_{x} c^T x + d^T \bar{y} \\ \text{s.t.} \quad & Ax + E\bar{y} \leq b + F\theta \\ & \theta_k^L \leq \theta_k \leq \theta_k^U, \ \ k = 1, \dots, m \\ & x \in \Re^n. \end{aligned} \tag{12}$$

The solution of (12) is given by a set of linear parametric profiles, $\hat{z}(\theta)^i$, where $\hat{z}(\theta)$ is convex, and corresponding critical regions, $\mathrm{CR}^i$ (Gal, 1995). The parametric solution has been graphically depicted in Figure 3, where $\hat{z}(\theta)^1, \hat{z}(\theta)^2$ and $\hat{z}(\theta)^3$ represent solution in the regions $\mathrm{CR}^1$, $\mathrm{CR}^2$ and $\mathrm{CR}^3$ respectively.

The region, $\mathrm{CR}^4$ in Figure 3, where no solution is found, represents an infeasible region for the current integer solution, $\bar{y}$. Note that unlike in the case of a single parameter where the infeasible region is simply given by intervals, for the multi-parametric case an infeasible region can be defined by polyhedral regions, which are obtained by systematically subdividing the initial region of $\theta$ as described in Dua and Pistikopoulos (2000). The final solution of the multiparametric LP subproblem in (12) which represents a parametric upper bound on the final solution is given by (i) a set of parametric profiles, $\hat{z}(\theta)^i$, and the corresponding critical regions, $\mathrm{CR}^i$, and (ii) a set of infeasible regions where $\hat{z}(\theta)^i = \infty$.

### MILP Subproblem

For each critical region, $\mathrm{CR}^i$, obtained from the solution of the multiparametric LP subproblem in (12), an MILP subproblem is formulated as follows:

$$\begin{aligned} & z = \min_{x,y,\theta} c^T x + d^T y \\ \text{s.t.} \quad & Ax + Ey \le b + F\theta \\ & c^T x + d^T y \le \hat{z}(\theta)^i \\ & \textstyle\sum_{j \in J^{ik}} y_j^{ik} - \sum_{j \in L^{ik}} y_j^{ik} \le |J^{ik}| - 1, \\ & \quad k = 1, \dots, K^i \\ & \theta \in CR^i \\ & x \in \Re^n \\ & y \in \{0,1\}^l, \end{aligned} \tag{13}$$

where $\theta$ is treated as a variable and $\theta \in \mathrm{CR}^i$ indicates that $\theta$ is bounded by the set of inequalities which define $CR^i$; $J^{ik} = (j|y_j^{ik} = 1)$ and $L^{ik} = (j|y_j^{ik} = 0)$, and $|J^{ik}|$ is the cardinality of $J^{ik}$ and $K^i$ is the number of integer solutions that have already been analysed in $\mathrm{CR}^i$. Note that the inequality, $c^T x + d^T y \le \hat{z}(\theta)^i$, excludes integer solutions with higher values than the current upper bound, $\hat{z}(\theta)^i$; the inequality, $\sum_{j \in J^{ik}} y_j^{ik} - \sum_{j \in L^{ik}} y_j^{ik} \le |J^{ik}| - 1$, corresponds to integer cuts prohibiting previous integer solutions from appearing again. The integer solution, $y = \bar{y}^1$, and the corresponding CRs, obtained from the solution of (13), are then recycled back to the multiparametric LP subproblem - to obtain another set of parametric profiles, as graphically depicted in Figure 4, where $\hat{z}(\theta)^5$ and $\hat{z}(\theta)^6$ represent solution in the regions $\mathrm{CR}^5$ and $\mathrm{CR}^6$, respectively (for simplicity in the graphical presentation the same integer solution, $y = \bar{y}^1$, is shown for all critical regions). Note that, in general one may obtain different integer solutions in different critical regions.

If there is no feasible solution to the MILP subproblem (13) in a $\mathrm{CR}^i$, that region is excluded from further consideration and the current upper bound in that region represents the final solution. Note also that the integer solution obtained from the solution of (13) is guaranteed to appear in the final solution, since it represents the minimum of the objective function at the point, in $\theta$, obtained from the solution of (13). The final solution of the MILP subproblem is given by a set of integer solutions and their corresponding $\mathrm{CR}^i$s.

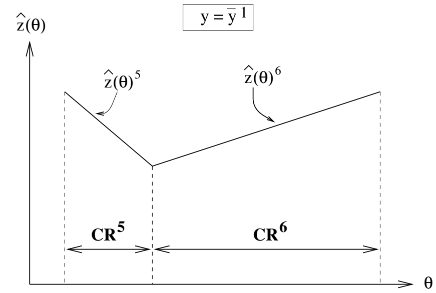


Figure 4: Parametric Solution - Multiparametric LP Subproblem for $y = \bar{y}^1$

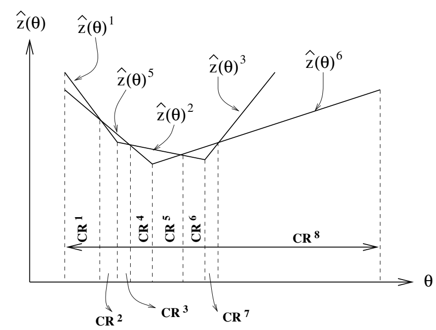


Figure 5: Comparison of Two Parametric Solutions

### Comparison of Parametric Solutions

The set of parametric solutions corresponding to an integer solution, $y = \bar{y}$, which represents the current upper bound are then compared to the parametric solutions corresponding to another integer solution, $y = \bar{y}^1$, in the corresponding $CR$s in order to obtain the lower of the two parametric solutions and update the upper bound. This is achieved by employing the procedure proposed by Acevedo and Pistikopoulos (1997). The result of the comparison procedure is graphically depicted in Figure 5, where $\hat{z}(\theta)^1$ represents the current upper bound in $\mathrm{CR}^2$, $\hat{z}(\theta)^2$ represents the current upper bound in $\mathrm{CR}^3$ and $\mathrm{CR}^6$, $\hat{z}(\theta)^3$ is the current upper bound in $\mathrm{CR}^7$, $\hat{z}(\theta)^5$ the current upper bound in $\mathrm{CR}^1$ and $\mathrm{CR}^4$, and $\hat{z}(\theta)^6$ represents the current upper bound in $\mathrm{CR}^5$ and $\mathrm{CR}^8$.

### Multiparametric MILP Algorithm

Based upon the above theoretical developments, the steps of the algorithm can be stated as follows:

**Step 0** (Initialization) Define an initial region of $\theta$, CR, with best upper bound $\hat{z}^*(\theta) = \infty$, and an initial integer solution $\bar{y}$.

**Step 1** (Multiparametric LP Problem) For each region with a new integer solution, $\bar{y}$:

- **(a)** Solve multiparametric LP subproblem (12) to obtain a set of parametric upper bounds $\hat{z}(\theta)$ and corresponding critical regions CR.

**(b)** If $\hat{z}(\theta) \leq \hat{z}^*(\theta)$ for some region of $\theta$, update the best upper bound function, $\hat{z}^*(\theta)$, and the corresponding integer solutions, $y^*$.

**(c)** If an infeasibility is found in some region CR, go to Step 2.

**Step 2** (Master Subproblem) For each region CR, formulate and solve the MILP master problem in (13) by (i) treating $\theta$ as a variable bounded in the region $CR$, (ii) introducing an integer cut, and (iii) introducing a parametric cut, $c^T x + d^T y \leq \hat{z}^*(\theta)$. Return to Step 1 with new integer solutions and corresponding CRs.

**Step 3** (Convergence) The algorithm terminates in a region where the solution of the MILP subproblem is infeasible. The final solution is given by the current upper bounds $\hat{z}^*(\theta)$ in the corresponding CRs.

The algorithm has been implemented in prototype software environments (PAROS, 2004; Kvasnica et al., 2003).

In the next section modelling of a co-generation power plant as a hybrid system is described in detail. The problem is then recast and solved as an MILP (Ferrari-Trecate et al., 2004).

## An Application to a Co-generation Power Plant

Consider the cogeneration combined cycle power plant comprising four main components: a gas turbine, a heat recovery steam generator, a steam turbine and a steam supply for a paper mill (Figure 6). The plant has two continuous-valued inputs ($u_1$ and $u_2$), and two binary inputs ($u_{l1}$ and $u_{l2}$):

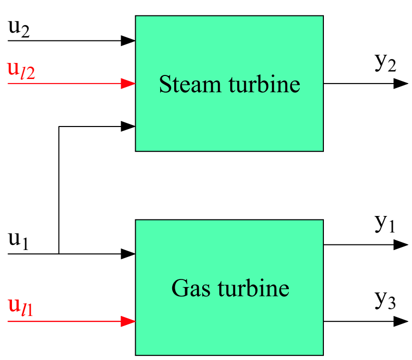


Figure 6: Block diagram of the Island power plant.

- $u_1$ is the set point for the gas turbine load (in percent). The permitted operation range for the gas turbine is in the interval $[u_{1,\min}, u_{1,\max}]$;
- $u_2$ is the desired steam mass flow to the paper mill (in kg/s). The permitted range for the steam flow is in the interval $[u_{2,\min}, u_{2,\max}]$;
- $u_{l1}$ and $u_{l2}$ are, respectively, the on/off commands for the gas and steam turbines; the "on" command is associated with the value one.

We assume that the inputs $u_1$ and $u_2$ are independent and all possible combinations within the admissible ranges are permitted. The binary input variables must fulfill the logic condition

$$u_{l2} = 1 \quad \Rightarrow \quad u_{l1} = 1, \tag{14}$$

which defines a priority constraint between the two turbines: The steam turbine can be switched on/off only when the gas turbine is on, otherwise the steam turbine must be kept off.

The output variables of the model are:

- the fuel consumption of the gas turbine, $y_1$ [kg/s];
- the electric power generated by the steam turbine, $y_2$ [MW];
- the electric power generated by the gas turbine, $y_3$ [MW];

Since we aim at optimizing the plant hourly, we choose a sampling time of one hour and we assume that the inputs are constant within each sampling interval. Due to the long sampling time we may also ignore plant dynamics like temperature changes, controller reaction times, etc. The input/output model of the plant has the form

$$y_1(k+1) = f_1(u_1(k)) \tag{15}$$
$$y_2(k+1) = f_2(u_1(k), u_2(k)) \tag{16}$$
$$y_3(k+1) = f_3(u_1(k)) \tag{17}$$
$$y_4(k+1) = f_4(u_1(k), u_2(k)), \tag{18}$$

where the maps $f_1$, $f_2$, $f_3$ and $f_4$ can be either affine or piecewise affine and are obtained by interpolating experimental data. The use of piecewise affine input/output relations allows to approximate nonlinear behaviours in an accurate way.

### Hybrid Features of the Plant

The main features which suggest modelling the power plant as a hybrid system are the following:

- the presence of the binary inputs $u_{l1}$ and $u_{l2}$;
- the turbines have different start up modes, depending on how long the turbines have been kept off;
- electric power, steam flow and fuel consumption are continuous valued quantities evolving with time.

Furthermore, the following constraints have to be taken into account:

- the operating constraints on the minimum amount of time for which the turbines must be kept on/off (the so-called minimum up/down times);
- the priority constraint (14). This condition, together with the previous one, leads to constraints on the sequences of logic inputs which can be applied to the system;
- the gas turbine load $u_1$ and the steam mass flow $u_2$ are bounded.

### The MLD Model of the Power Plant

All the features of the power plant mentioned can be captured by a hybrid model in the MLD form. In the following we show, as an example, how to derive the MLD description of the different types of start up for the turbines. We focus on the steam turbine. The procedure is exactly the same for the gas turbine.

Typical start up diagrams show that the longer the time for which a turbine is kept off, the longer the time required before producing electric power when it is turned on. This behavior is common to all turbines and is due to the need of heating the materials of the mechanical components in a gradual way, in order to avoid dangerous mechanical stresses. This feature can be modelled, in an approximate way, as a delay between the time instant when the plant is started and the instant when the production of electric power begins.

| | *time spent off* (*h*) | *delay* (*h*) |
|---|---|---|
| normal start up | $[0,8]$ | 1 |
| hot start up | $]8,60]$ | 2 |
| warm start up | $]60,120]$ | 3 |
| cold start up | $]120,+\infty[$ | 4 |

Table 2: Typical types of start up procedures for steam and gas turbines

In the model four different types of start up procedures for the steam and gas turbines (Table 2) are considered. Thus, for instance, if a turbine has been kept off for 70 hours, it will produce electric power with a delay of 3 hours from the instant when the start command is given. The shut down procedure is simpler: When a turbine is turned off, at the next time instant (one hour after!) it will produce zero electric power.

In order to take into account in the MLD model the different start up procedures, it is necessary to introduce three clocks with reset (which are state variables), five auxiliary logic variables $\delta$, and three auxiliary real variables $z$. The clocks are defined as follows:

- $\xi_{on}$ stores the consecutive time during which the turbine produces electric power. If the turbine is producing electric power, $\xi_{on}$ is increased according to the equation
$$\xi_{on}(k+1) = \xi_{on}(k) + 1 \qquad (19)$$
otherwise it is kept equal to zero;
- $\xi_{off}$ stores the consecutive time during which the turbine does not produce electric power. So, if the turbine is off or does not produce electric power (as in a start up phase), $\xi_{off}$ is increased according to the equation
$$\xi_{off}(k+1) = \xi_{off}(k) + 1 \qquad (20)$$
otherwise it is kept equal to zero;
- $\xi_d$, when it is positive, stores the delay that must occur between the turning on command and the actual production of electric power. If the turbine is turned on, $\xi_d$ starts to decrease according to the law
$$\xi_d(k+1) = \xi_d(k) - 1 \qquad (21)$$
and the energy generation will begin only when the condition $\xi_d < 0$ is fulfilled. Otherwise, if the turbine is off, $\xi_d$ must store the delay corresponding to the current type of start up. In view of Table 2, when the turbine is disconnected ($u_{l2} = 0$), the value of $\xi_d$ is given by the following rules:
$$\begin{array}{ll} \xi_{off} \le 8h & \Rightarrow \xi_d = 0 \\ 8h < \xi_{off} \le 60h & \Rightarrow \xi_d = 1 \\ 60h < \xi_{off} \le 120h & \Rightarrow \xi_d = 2 \\ \xi_{off} > 120h & \Rightarrow \xi_d = 3 \end{array} \qquad (22)$$

For instance, if at the time instant $\bar{k}$ the turbine is off ($u_{l2}(\bar{k}) = 0$) and $\xi_{off}(\bar{k}) = 70$, $\xi_d$ will be set equal to 2. If, at the next time instant $\bar{k}+1$ the turbine is switched on ($u_{l2}(\bar{k}+1) = 1$), $\xi_d$ will evolve according to equation (21) and when, at the time instant $\bar{k}+4$, the condition $\xi_d < 0$ is fulfilled, $\xi_{off}$ is reset to zero ($\xi_{off}(\bar{k}+4) = 0$) and $\xi_{on}$ starts to increase according to equation (19).

Since the energy production depends on the condition $\xi_d < 0$, we introduce the logic variable $\delta_d$ defined by the threshold condition

$$\delta_d = 1 \quad \Leftrightarrow \quad \xi_d < 0 \qquad (23)$$

which, written as

$$\delta_d = 1 \quad \Leftrightarrow \quad \xi_d + 0.5 \le 0$$

can be translated into mixed integer linear inequalities using the rule P9 in Table 1.

Then we introduce the logic variable $\delta_{on}$, which represent the condition 'the turbine is on and produces electric power' through the logic statement

$$\delta_{on} = 1 \quad \Leftrightarrow \quad (u_{l2} = 1) \wedge (\delta_d = 1) \qquad (24)$$

In order to find the mixed-integer linear inequalities representing (24) one has two possibilities. The first one is to re-write (24) in Conjunctive Normal Form (CNF) (Cavalier et al., 1990) and then use the rules of Table 1. Alternatively, one can use the algorithm described in (Bemporad et al., 1999) that allows computing the inequalities representing the proposition (24) in an automated way starting from the truth-table of the proposition (24).

The dynamics of the clocks $\xi_{on}$, $\xi_{off}$ can be written as

$$\xi_{on}(k+1) = [\xi_{on}(k)+1]\delta_{on}(k) \qquad (25)$$
$$\xi_{off}(k+1) = [\xi_{off}(k)+1](1-\delta_{on}(k)) \qquad (26)$$

The product between logic variables (as $\delta_{on}$) and continuous variables (as $\xi_{on}$ and $\xi_{off}$) can be translated in the MLD form by introducing the auxiliary real variables $z_{on}$ and $z_{off}$ defined as

$$z_{on}(k) = (\xi_{on}(k)+1)\delta_{on}(k) \qquad (27)$$
$$z_{off}(k) = \xi_{off}(k)\delta_{on}(k) \qquad (28)$$

These relations can be represented through linear inequalities by using the rule P10 of Table 1. Finally the dynamics of the counters $\xi_{on}$ and $\xi_{off}$ in the MLD form is given by the equations:

$$\xi_{on}(k+1) = z_{on}(k)$$
$$\xi_{off}(k+1) = z_{off}(k)$$

In order to represent the dynamics of the counter $\xi_d$, three more auxiliary binary variables and one auxiliary real variable are needed. The binary variables $\delta_h$, $\delta_w$, $\delta_c$ are necessary to distinguish the different types of start up and so their definition depends on the value of $\xi_{off}$. According to the Table 2, we have

$$\delta_h = 1 \quad \Leftrightarrow \quad \xi_{off} \geq 8h \qquad (29)$$
$$\delta_w = 1 \quad \Leftrightarrow \quad \xi_{off} \geq 60h \qquad (30)$$
$$\delta_c = 1 \quad \Leftrightarrow \quad \xi_{off} \geq 120h \qquad (31)$$

Let $z_d$ be defined as

$$z_d = \begin{cases} \xi_d(k) - 1 & if\ u_{l2} = 1 \\ \delta_h(k) + \delta_w(k) + \delta_c(k) & if\ u_{l2} = 0 \end{cases} \qquad (32)$$

Again, (32) can be translated into mixed-integer linear inequalities by using the rules P8 and P10 of Table 1. It is now possible to write the dynamics of the state $\xi_d$ as

$$\xi_d(k+1) = z_d(k)$$

that is compatible with the MLD form (2) and must be complemented with the inequalities representing (29), (30), (31) and (32).

**Remark 1** From the equations (19), (20), (21), it follows that the clocks $\xi_{on}$, $\xi_{off}$ and $\xi_d$ are unbounded, but it is easy to make them bounded by a value $\bar{\xi}_{on}$ by introducing further auxiliary variables modeling rules like 'if $\xi_{on} > \bar{\xi}_{on}$ then $\xi_{on} = \bar{\xi}_{on}$'.

By using the methodology outlined in this section, it is possible to derive an MLD model capturing every hybrid feature of the power plant. The complete model is described in (Spedicato, 2001) and involves 12 state variables, 25 $\delta$-variables and 9 $z$-variables.

The 103 inequalities stemming from the representation of the $\delta$ and $z$ variables are collected in the matrices $E_i$, $i = 1, ..., 5$ of (4) and are not reported here due to the lack of space. Some significative simulations which test the correctness of the MLD model of the power plant are also available in (Spedicato, 2001).

### Cost Functional

The following cost functional is minimized:

$$\begin{aligned} J = C_{dem} + C_{change} + C_{fuel} + C_{start\ up} + \\ + C_{fixed} - E + C_{start\ up\ gas} + C_{fixed\ gas} \end{aligned} \qquad (33)$$

where $C_{dem}$ is the penalty function for not meeting the electric and steam demand, $C_{change}$ is the cost for changing the operation point between two consecutive time instants, $C_{fuel}$ takes into account the cost for fuel consumption, $C_{start\ up}$ is the cost for the start up of the steam turbine, $C_{fixed}$ represents the fixed running cost of the steam turbine, $E$ represents earnings from sale of steam and electricity; this term has to take into account that the surplus production can not be sold, $C_{start\ up\ gas}$ is the start up cost for the gas turbine and $C_{fixed\ gas}$ represents the running fixed cost of the gas turbine.

### Constraints and Derivation of the MILP

The optimization problem can be written as one of minimizing (33) subject to $x(t|k) = x_k$, and for $t = k,...,k+M-1$, to equations of the form (2)-(4) where $k$ and $M$ are the current time instant and the length of the control horizon, respectively. The optimization variables are $\{u(t|k)\}_{t=k}^{k+M-1}$, $\{\delta(t|k)\}_{t=k}^{k+M-1}$, $\{z(t|k)\}_{t=k}^{k+M-1}$. The repetitive solution of this MILP can be avoided by using parametric programming to obtain $\{u(t|k)\}_{t=k}^{k+M-1}$, $\{\delta(t|k)\}_{t=k}^{k+M-1}$, $\{z(t|k)\}_{t=k}^{k+M-1}$ as a function of $x_k$.

## Concluding Remarks

In this paper, an overview of the recent developments in the area of hybrid systems and parametric programming was presented. It was illustrated how optimal control problems related to hybrid systems can be formulated as mixed integer progams.

While solvers for mixed-integer programs are available, parametric programming techniques can be used to recast hybrid systems as multiparametric mixed-integer programs and solved to obtain control variables as an explicit function of the state variables. This reduces the on-line hybrid control problem to simple function evaluations. The key features of hybrid systems were illustrated by a cogeneration power plant example.

## REFERENCES


Acevedo, J. and E. N. Pistikopoulos (1997). A Multiparametric Programming Approach for Linear Process Engineering Problems under Uncertainty. Industrial and Engineering Chemistry Research **36**, 717–728.

Baotic, M., M. Vasak, M. Morari and N. Peric (2002). Hybrid theory based optimal control of electronic throttle. Proceedings of the American Control Conference, Denver.

Bemporad, A. and M. Morari (1999a). Control of Systems Integrating Logic, Dynamics, and Constraints. Automatica **35**(3), 407–427.

Bemporad, A. and M. Morari (1999b). Verification of Hybrid Systems via Mathematical Programming. In: Hybrid Systems: Computation and Control; Second International Workshop - Lecture Notes in Computer Science (F.W. Vaandrager and J.H. van Schuppen, Eds.). Springer Verlag. pp. 31–45.

Bemporad, A., D. Mignone and M. Morari (1999). A Framework for Control, Fault Detection, State Estimation and Verification of Hybrid Systems. Proc. American Control Conference pp. 134–138.

Bemporad, A., F. Borrelli and M. Morari (2000a). Piecewise linear optimal controllers for hybrid systems. Proceedings of the American Control Conference, 1190-1194.

Bemporad, A., G. Ferrari-Trecate and M. Morari (2000b). Observability and Controllability of Piecewise Affine and Hybrid Systems. IEEE Trans. on Automatic Control **45**(10), 1864–1876.

Biegler, L.T. and I.E. Grossmann (2004). Challenges and research issues for product and process design optimization. Foundations of Computer Aided Process Design, Princeton.

Borrelli, F., A. Bemporad and M. Morari (2001). A hybrid approach to traction control. In: Hybrid Systems: Computation and Control (M.D.D. Benedetto and A. Sangiovanni-Vincentelli, Eds.). Springer, Berlin. p. 3.

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

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

Dua, V. and E. N. Pistikopoulos (1999). Algorithms for the solution of multiparametric mixed-integer nonlinear optimization problems. Industrial and Engineering Chemistry Research **38**, 3976–3987.

Dua, V. and E. N. Pistikopoulos (2000). An algorithm for the solution of multiparametric mixed integer linear programming problems. Annals of Operations Research **99**, 123–139.

Dua, V., K. P. Papalexandri and E. N. Pistikopoulos (2003). Global optimization issues in multiparametric continuous and mixed-integer optimization problems. accepted for publication in the Journal of Global Optimization.

Dua, V., N. A. Bozinis and E. N. Pistikopoulos (2002). A multiparametric programming approach for mixed-integer quadratic engineering problems. Computers & Chemical Engineering **26**, 715–733.

Ferrari-Trecate, G., D. Mignone and M. Morari (2002a). Moving Horizon Estimation for Hybrid Systems. IEEE Trans. on Automatic Control **47**(10), 1663–1676.

Ferrari-Trecate, G., E. Gallestey, A. Stothert, P. Letizia, M. Spedicato, M. Morari and M. Antoine (2002b). Modeling and control of cogeneration plants: A hybrid systems approach. In: Proc. of the 5th International Workshop on Hybrid Systems: Computation and Control (C.J. Tomlin and M.R. Greenstreet, Eds.). Springer-Verlag, Berlin.

Ferrari-Trecate, G., E. Gallestey, P. Letizia, M. Spedicato, M. Morari and M. Antoine (2004). Modeling and control of cogeneration power plants: A hybrid system approach. to appear in IEEE Trans. Control Systems Technology.

Ferrari-Trecate, G., F.A. Cuzzola, D. Mignone and M. Morari (2002c). Analysis of discrete-time piecewise affine and hybrid systems. Automatica **38**(12), 2139–2146.

Gal, T. (1995). Postoptimal Analyses, Parametric Programming, and Related Topics. de Gruyter. New York.

Gallestey, E., A. Stothert, D. Castagnoli, G. Ferrari-Trecate and M. Morari (2003). Using model predictive control and hybrid systems for optimal scheduling of industrial processes. Automatisierungstechnik **51**, 285–293.

Kvasnica, M., P. Grieder, M. Baotic and M. Morari (2003). Multi-Parametric Toolbox (MPT). http://control.ee.ethz.ch/∼hybrid/mpt/.

Mignone, D., G. Ferrari-Trecate and M. Morari (2000). Stability and stabilization of piecewise affine and hybrid systems: A LMI approach. Proc. 39th Conference on Decision and Control.

Möbus, R., M. Baotic and M. Morari (2003). Multi-object adaptive cruise control. In: Hybrid Systems: Computation and Control (O. Maler and A. Pnueli, Eds.). Springer, Berlin. pp. 359–374.

Morari, M., M. Baotic and F. Borrelli (2003). Hybrid systems modeling and control. European Journal of Control **9**, 177–189.

Niederberger, D., M. Baotic and M. Morari (2003). Active vibration suppression using switched PZTs. Technical Report AUT03-06, Automatica Control Lab, ETH Zurich.

Pantelides, C.C. (2001). Optimisation of hybrid processes and hybrid controllers. In: Hybrid Systems: Computation and Control (M.D.D. Benedetto and A. Sangiovanni-Vincentelli, Eds.). Springer, Berlin. p. 3.

PAROS (2004). Parametric Optimization Solutions Ltd. http://www.parostech.com.

Piccoli, B. (1999). Necessary conditions for hybrid optimization. IEEE Proceedings of the 38th Conference on Decision and Control, Phoenix, 410-415.

Raman, R. and I.E. Grossmann (1991). Relation between MILP modeling and logical inference for chemical process synthesis. Computers Chemical Engineerings **15**(2), 73–84.

Riedinger, P., C. Zanne and F. Kratz (1999). Time optimal control of hybrid systems. Proceedings of the American Control Conference, San Diego, 2466-2470.

Sakizlis, V., N. Kakalis, V. Dua, J. D. Perkins and E. N. Pistikopoulos (2004). Design of robust model based controllers via parametric programming. Automatica **40**, 189–201.

Sakizlis, V., V. Dua, J. D. Perkins and E. N. Pistikopoulos (2002a). The explicit control law for hybrid systems via parametric programming. Proceedings of the 2002 American Control Conference, Anchorage.

Sakizlis, V., V. Dua, J. D. Perkins and E. N. Pistikopoulos (2002b). The explicit model-based control law for continuous time systems via parametric programming. Proceedings of the 2002 American Control Conference, Anchorage.

Spedicato, M. (2001). Modellizzazione di impianti cogenerativi mediante sistemi ibridi. M. Sc. thesis, Universita' degli Studi di Pavia.

Sussmann, H.J. (1999). A nonsmooth hybrid maximum principle. Stability and Stabilization of Nonlinear Systems Lecture Notes in Control and Information Sciences, 246: 325-354.

Torrisi, F. D., A. Bemporad and D. Mignone (2000). HYSDEL - A Tool for Generating Hybrid Models. Technical report. AUT00-03, Automatic Control Laboratory, ETH Zurich.