Tutorial introduction to the modelling and control of hybrid systems

Luisella Balbis*

Industrial Control Centre,
Department of Electronic and Electrical Engineering,
University of Strathclyde,
50 George Street,
Glasgow G1 1QE, Scotland
E-mail: lbalbis@eee.strath.ac.uk
*Corresponding author

Andrzej W. Ordys

Kingston University in London,
Friars Avenue, Roehampton Vale,
London SW15 3DW, UK
E-mail: A.Ordys@kingston.ac.uk

Michael J. Grimble and Yan Pang

Industrial Control Centre,
Department of Electronic and Electrical Engineering,
University of Strathclyde,
50 George Street,
Glasgow G1 1QE, Scotland
E-mail: m.grimble@eee.strath.ac.uk
E-mail: yan.pang@strath.ac.uk

Abstract: A tutorial introduction is provided to the relatively new subject of hybrid systems. The modelling of hybrid systems is assuming ever greater importance for systems where the combination of continuous control and with logical decision making is required. This arises in some of the most critical operating regions where systems are under start-up, shutdown or are undergoing major planned changes. There is recognition that separate independent design of these functions will reduce achievable performance and cause unpredictable behaviour in some of the most safety critical areas of operation. The introduction provided is not exhaustive but it introduces some of the main concepts and motivates the use in applications of this relatively new area of control design. The theoretical results are illustrated using engineering examples.

Keywords: hybrid systems; discrete-event systems; hybrid automata; Petri nets; optimal control.

Reference to this paper should be made as follows: Balbis, L., Ordys, A.W., Grimble, M.J. and Pang, Y. (2007) 'Tutorial introduction to the modelling and control of hybrid systems', *Int. J. Modelling, Identification and Control*, Vol. 2, No. 4, pp.259–272.

Biographical notes: Luisella Balbis received an MSc in Computer Engineering from the Universita' degli Studi di Genova, Italy in 2002. Currently, she is working towards a PhD in System Control at the University of Strathclyde, Glasgow. Her current research interest includes predictive control, hybrid system and real time control.

Andrzej W. Ordys is a Professor of Automotive Engineering at Kingston University, London. Previously, he worked at the University of Strathclyde as the British Energy Senior Lecturer in Control Systems in the Department of Electronic and Electrical Engineering. He has been contributing to the theoretical development of stochastic and predictive control theory and the theory of benchmarking control algorithms. He has been involved in the management of many research projects especially with large international participation. He has contributed to a range of industrial application projects including those for the automotive, power, oil and gas, chemical, defence and metal processing industries. He is a Member of IEE and Chartered Engineer and Senior Member of IEEE. Currently, he is Chairman of EC&I Section of IEE Scotland. He has been Member of International Programme Committee for many international conferences, most recently for IEEE American Control Conference 2005, IASTED Control Applications Conference 2005 and IASTED Modelling Identification and Control Conference 2005, IEEE Conference on Control Applications, Taiwan, June 2004. He is a Member of the Organising Committee of UKACC Control 2006 Conference.

Michael J. Grimble received a BSc (Coventry) in Electrical Engineering in 1971, an MSc, a PhD and a DSc from the University of Birmingham in 1971, 1978 and 1982, respectively. Whilst in industry he took an Open University Degree studying Mathematics and received a BA in 1978. The University of Strathclyde, Glasgow, appointed him to the Professorship of Industrial Systems in 1981. He is also the Director of the Industrial Control Centre and the Technical Director of ISC Ltd. His interests in control theory are focussed upon: robust control theory, multivariable design techniques, optimal control, adaptive control, estimation theory, non-linear control, performance assessment and condition monitoring.

Yan Pang received her BS from Harbin Engineering University, PR China in 2000 and his PhD from the University of Strathclyde, UK in 2004. From 2004 to 2005, she was a Research Associate with the Department of Electronic and Electrical Engineering, Sheffield University, UK. Since 2005, she has been a Research Fellow with the Industrial Control Centre, University of Strathclyde.

1 Introduction

Traditional control loop design has been concerned with feedback loops and either continuous or discrete-time control of the loop. Any switching that has been involved or reconfiguration has been treated as a separate issue, left to operators or even automatic decision making systems. In some systems the control in such operating regions is some of the most difficult and critical and yet an integrated approach has not been available. The feature that makes these problems different is the importance of the discrete events that occur and that may require major operational changes. Such events may of course occur at random times when say loading conditions require new sources of power or reconfiguration of a system. These changes are therefore event driven and a so-called hybrid control strategy is needed.

No common definition for a hybrid system is available. A general definition suggests that a dynamical system should be considered a hybrid system if (and only if) it is impossible to treat the system either as a purely continuous-variable system or as a purely discrete-event system (Lunze, 2002). It is evident that hybrid dynamical systems have existed for a long time but in the past discrete events have mainly been treated separately from the control of the continuous dynamics of the system.

Continuous systems theory often assumes that the system is described by continuous-time differential equations or discrete-time differential equations. On the other hand discrete event theory considers systems whose dynamics are characterised by the asynchronous occurrence of discrete events. These systems are described by a traditional Discrete Event Systems (DES) approach such as Automata, Petri nets and Markov chains.

The main reason *hybrid systems* is becoming such an active area of research in automotive and aerospace systems is that advances are needed in systems with automated decision making/switching, if companies are to gain a competitive advantage. There are also cases where considering systems as purely continuous or as purely discrete is not sufficient to provide even adequate control. The interaction between event driven (discrete) and time driven (continuous) dynamics can result in very complex behaviour that requires methods in which both continuous and discrete systems theory are used simultaneously. Hybrid behaviour is of course exhibited in most technical systems that are composed of physical components with continuous dynamics, where the reaction to external events plays a major role. The hybrid aspects of a system can also arise from internal causes, like operating mode changes and failures. Hybrid features can be identified in many applications such as:

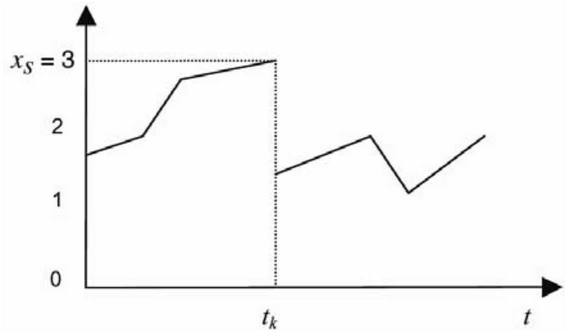
- automotive (Borrelli et al., 2001; Möbus et al., 2003; Vasak et al., 2004)
- power generation plant (Ferrari-Trecate et al., 2000, 2002)
- aerospace (Pritchett et al., 2000)
- manufacturing (Gallestey et al., 2003; Pepyne and Cassandras, 2000)
- traffic control (Tomlin et al., 1998)
- chemical process control (Lennartson et al., 1996).

This paper is organised as follows. Sections 2 and 3 provide an introduction to the dynamic behaviour and to existing modelling frameworks for hybrid system respectively. In Section 4 the most promising techniques for control of hybrid systems are described. Section 5 illustrates the application of predictive control techniques to a case study. The tutorial includes a variety of approaches, but the coverage is only partial and does not address the problem of hybrid systems analysis. The main aim is to give an introduction to a field that has been the subject of extensive research over the past few years. For surveys on stability analysis the reader can refer to Decarlo et al. (2000) and Davrazos and Koussoulas (2001).

2 Hybrid systems behaviour

Hybrid systems involve both continuous-valued and discrete-valued variables and their evolution is described by an equation of motion that generally depends on both. For example, consider the state trajectory of an autonomous first-order system shown in Figure 1 (Lunze, 2002).

Figure 1 State trajectory of a hybrid system



Note that when the state reaches the threshold value x_s its trajectory exhibits a jump. As the system state does not move continuously in a *Lipschitz sense* from one real value to another its dynamics cannot be described by the method of purely continuous theory. Conversely, discrete event theory provides a suitable method to represent the autonomous jump between the instants t_k^- and t_k^+ but not the behaviour of the system between the changes for $t \neq t_k$.

A detailed description of hybrid phenomena can be found in Branicky et al. (1998) and Lunze (2002). Generally such systems can be characterised by a differential equation:

$$\dot{x}(t) = \xi(x,u,t), \quad t \geq 0 \tag{1}$$

where the function ξ depends on the continuous state trajectory $x(t)$, the continuous part of the control $u(t)$ and the discrete phenomena.

Four phenomena exhibiting hybrid features can be particularly highlighted:

1 *Autonomous switching*: the state trajectory cannot be represented by a unique expression. The vector field $\xi(\cdot)$ is composed of different fields $\xi_i(\cdot)$ defined inside a region of validity $h(\cdot)$ and changes discontinuously if the state reaches a given bound.

2 *Autonomous impulses*: the state of a system changes impulsively or jumps after it has reached a threshold x_s.

3 *Controlled switching*: this case is similar to autonomous switching, but here the changes in the vector $\xi(\cdot)$ is governed by the control input u with an associated cost.

4 *Controlled impulses*: the state of the system may change impulsively or jump if the control input u with an associated cost reaches a given bound u_s.

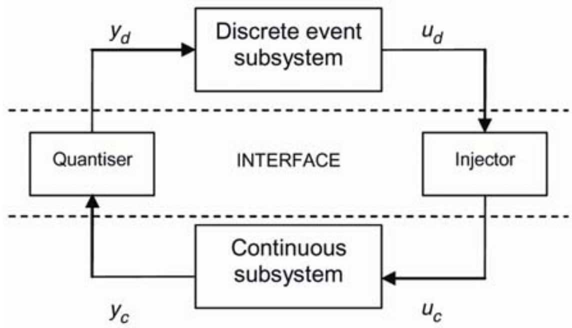
The modelling of hybrid phenomena is extremely challenging. The models proposed for this kind of system have to describe both the continuous and the discrete-time dynamics of the system. At the same time they have to be simple enough for the analysis and solution of synthesis problems.

3 Hybrid system modelling

3.1 Decomposition of hybrid systems

The decomposition of hybrid systems involves the use of a discrete event model and a continuous model and a means of coupling them by suitable interfaces. These convert continuous-valued measurements into discrete event signals and vice versa (Antsaklis et al., 1993). This approach has been used widely in hierarchical systems and supervisory control systems (Branicky et al., 1998; Lemmon et al., 1999; Stiver et al., 1996). A hybrid control system may have the structure depicted in Figure 2 (Lunze, 2002). In this case the plant constitutes the physical part of the system whose behaviour is governed by the laws of physics or chemistry and can be approximated by differential equations modelled in continuous time. On the other hand, the controller is a discrete-event system driven by external state events. The state-space of the discrete part is a discrete set and the state trajectory is a piecewise constant function that switches from one value to another when an event occurs. A number of frameworks exist for describing DES, such as *automata, Petri nets, finite state machine, semi-Markov process, max-plus-algebra* and so on (Cassandras and Lafortune, 1999). Suitable interfaces called quantisers and injectors map the measurements of continuous variables from the plant into a set of discrete valued signals or symbols and vice versa.

Figure 2 Decomposed hybrid system



The complexity due to the interaction between the continuous and discrete aspects and the synchronisation of the two models requires adequate simulation and modelling toolboxes. An overview of the hybrid simulation and software packages supporting hybrid features is presented in Mosterman (1999). Some of the main packages include:

- *gPROMS* is designed for process modelling, simulation and optimisation. It has discrete-event modelling facilities and can solve large sets of equations efficiently, partial differential equations and partially determinable systems (Van Beek and Rooda, 2000).

- *Dymola*, *Omola* and *Smile* are object-oriented modelling languages, in which the structure of the system is represented in the hierarchical formulation of the model. Submodels can be connected in different ways that express the physical nature of the connections (Elmqvist et al., 1993).
- *Modelica* is a language for physical modelling that aims to unify several different languages, building on non-causal modelling with true ordinary differential and algebraic equations and using object-oriented constructs to facilitate reuse of modelling knowledge (Elmqvist and Mattsson, 1997).
- χ is a hybrid specification language developed for modelling and simulation of discrete-event, continuous-time and hybrid systems, especially for manufacturing plant. The simulation of a hybrid model consists of a sequence of continuous phases that alter with discrete phases at certain time points (Van Beek and Rooda, 2000).
- *Simulink/Stateflow* (1999) is block diagram-based for modelling and simulating event-driven systems. It provides a valid solution for designing embedded systems that contain supervisory logic, combining graphical modelling and animated simulation (www.mathworks.com). Stateflow is a tool integrated into the Matlab environment that uses a variant of the conventional finite state machine notation and state diagram, called a State charts approach (Harel, 1997; Pascal and Sahbani, 2000).

A simulation can involve a detailed model and description of the plant. However, the analysis of large hybrid systems, by means of simulation tools, involves problems and limits due to the computational complexity and the dependency on the model parameters.

3.2 Representation of hybrid systems by extended discrete event models

The extended discrete event models technique is based on formal graph theoretic models well known in the computer science community, such as finite state machines and Petri nets. This approach consists of extending classic DES theories by continuous variables, whose behaviour is governed by differential equations connected to discrete states (Kowalewski, 2002). The resulting frameworks used in this area are hybrid automata and Hybrid Dynamical Nets (Hummel and Fengler, 2001; Svadova, 2001).

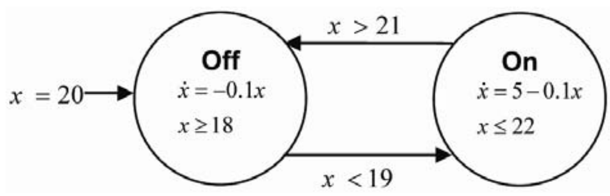
3.2.1 Hybrid automata

A hybrid automata model integrates discrete finite automata with time-dependent continuous variables. It can be represented as a tuple: (N, X, init, inv, flow, jump, Σ, event), where (Henzinger, 1996):

- N is a finite directed graph (V, E) that denotes all possible states that a discrete-event system can occupy. Here, V is a set of vertices called control modes and $E \subset V \times V$ is a set of directed arcs between vertices called control switches.
- X is a finite set $\{x_1,\ldots, x_n\}$ of continuous-valued variables called timers, that represent the continuous dynamics of the hybrid system.
- Init, inv and flow are three labelling functions that assign each vertex $v \in V$ three predicates. Init(v) determines an initial condition in P^n for the time and the continuous states. Inv(v) formulates conditions which have to be true while the system remains in a discrete state and when an invariant evaluates to false, the discrete state must be left or must not be entered, respectively. Flow(v) are differential equations assigned to each discrete state that describe the evolution of continuous states of the system.
- Jump(e) is a labelling function that assigns to each edge $e \in E$ a predicate formed from guard atomic event, that is, equations of the form $[a'x_i R b'x_j]$ or $[a'x_i R c]$, where a and b are real vectors and c is a constant. These equations mean that the product $a'x_i$ stands in relation R (< or >) with $b'x_i$ or c.
- Σ is a finite set of events and event is an edge labelling function that assigns to each edge an event event: $E \to \Sigma$.

Thermostat example: a simple example for hybrid automaton is shown in Figure 3 (Henzinger, 1996). This automaton provides a rough model for a thermostat. The continuous variable x represents the temperature and it is controlled by the heater that can assume two discrete states. In control mode *Off*, the heater is off, and the temperature falls according to the flow condition $\dot{x} = -0.1x.$ In control mode *On*, the heater is on, and the temperature rises according to the flow condition: $\dot{x} = 5 - 0.1x.$ As an initial condition the heater is assumed off and the temperature is 20°. When the temperature falls below 19°, the heater may go on, according to the jump condition x < 19. According to the invariant condition $x \geq 18$, the heater must go on when the temperature falls to 18°.

Figure 3 Thermostat automaton



In the literature of hybrid systems there are different classes of hybrid automata, depending on the type of continuous dynamics of the system. Among them we can distinguish:

- *Timed automata*: in a timed automaton the dynamics of the continuous variable x increases with a rate of one. In other words, the flow conditions satisfy the differential equation.

- *Rectangular automata*: in this case the flow conditions are specified by an interval, and therefore the dynamics of each continuous variable x varies according to the non-deterministic differential equation $\dot{x} \in [a,b]$.

- *Linear automata*: in this case the derivatives of the continuous variables satisfy linear relationships.

- *Input/output automata*: in this case the concept of hybrid automata is refined introducing input and output variables and input and output actions (Lynch et al., 2003).

Hybrid automata have been used mainly for the analysis and verification of hybrid systems. Algorithmic verification is a procedure that aims to check whether the hybrid systems satisfied the desired specifications. This is achieved using a computer algorithm that receives an input to the model of the system and has the desired specifications, and then verifies by means of search techniques if the behaviour of the system model in all possible circumstances satisfies the requirements (Kowalewski, 2002). The analysis could be addressed to safety specifications, in order to check if the system is able to avoid undesirable regions and satisfy reachability issues. This aims to verify if a desired state is reachable from a given state. The algorithmic approach for the analysis of hybrid systems involves computational issues and it is only possible for classes for which the search procedure terminates in a finite number of steps. It can be demonstrated that verification problems are feasible for restricted classes of hybrid systems (Henzinger et al., 1995). For the other classes the problem is not totally solved.

In order to show that the analysis problem is solvable for a class of hybrid systems, abstraction techniques could be applied mapping the original system into an hybrid automaton, in which both the quantitative measurement and the qualitative information related to the dynamics of the system are preserved.

3.2.2 *Hybrid dynamical nets*

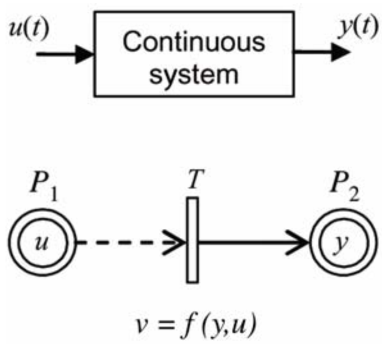
The main idea of the Hybrid Dynamical Nets approach consists of modelling the discrete and the continuous behaviour of the hybrid system combining discrete Petri Net with continuous Petri Net (Drath, 2002; Drath et al., 1999). The Petri Net is a mathematical and graphical modelling tool constituted by a bipartite oriented graph. Place P and transition T represents the nodes of the graph connected by arcs. There are directed arcs from the input places (*preconditions*) of a transition to the transition, and directed arcs from the transition to its output places (*post-conditions*). A place can be connected with several input/output transitions and a transition can be connected with several input/output places. The *marking* M of a Petri Net denotes the current state of the network places, and is governed by a marking function $m(P_i)$ that assigns a positive integer number of tokens to each place of the net. Moreover, a function W: $F \to R^+$ assigns each arc of the network a weight w.

A transition is *enabled* if all its input places contain at least as many tokens as is the weight of arc from the place to the transition. An enabled transition may be *fired*. During firing the transition removes w_i tokens (where w_i is the weight of the arc whose endpoint is the transition) from its starting (input) place and adds w_j tokens (where w_j is the weight of the arc starting at the transition) to its ending (output) place.

The classical Petri Net approach is suitable for a variety of systems including concurrent, distributed, asynchronous, parallel, deterministic and non-deterministic, but not for continuous systems. This type of system could be described by the continuous Petri Net approach, in which the marking of a place is a real positive number. A firing of a transition is carried out like a continuous flow (Svadova, 2001). In a continuous Petri Net a transition T_j is always active and the instantaneous firing speed function $v = f(u, y)$ at time t indicates the quantity of marks transferred from input place to output place of transition T_j per unit of time. In the field of hybrid systems, the continuous net with constant speed (CCPN) is usually used in modelling hybrid Petri nets. In CCPN the marking M is a real positive number, whose value is function of the time. The maximum firing speed V_j associated with transition T_j is constant and marking independent, whereas the instantaneous firing speed $v_j(t)$ at time t indicates the quantity of marks transferred from input place to output place of transition T_j per unit of time.

A continuous system represented by means of continuous Petri Net is showed in Figure 4 (Drath, 2002).

Figure 4 Continuous Petri net system



Hybrid systems can be modelled combining the discrete and continuous PN described above, by means of discrete or continuous transitions. The result is a Hybrid Dynamic Net, in which the discrete and continuous elements in Figure 5 are simultaneously adopted (Drath, 2002).

The firing rules for these transitions in Hybrid Dynamic Nets are the following (Svadova, 2001):

- a discrete transition is enabled if each input place has $M^n(P_i, T_j) \geq \text{Pre}(P_i, T_j)$
- a continuous transition is enabled if for each input discrete place P_i $M^n(P_i, T_j) \geq \text{Pre}(P_i, T_j)$ and $M^n(P_i, T_j) > 0$.

Three main interactions between continuous and discrete subsystems are defined (Drath, 2002).

Discrete control of continuous processes: discrete control places govern the firing of continuous transitions according to the rules above. In the example in Figure 6 the transition T is enabled only if P_3 is marked.

Figure 5 Modelling element in Hybrid Dynamical Nets

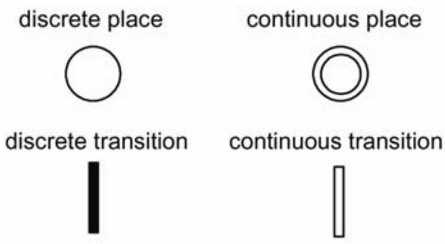
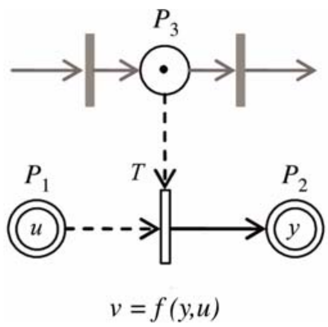
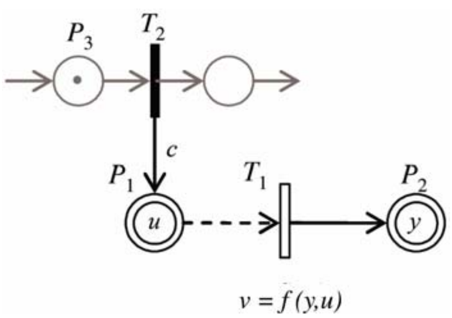


Figure 6 Discrete control of continuous processes



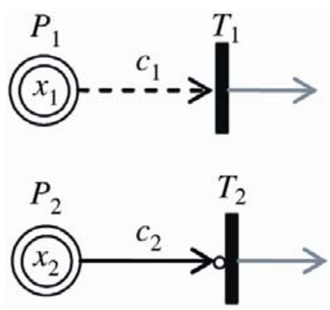
Generation of step functions: the firing of discrete transitions generates an immediate change in continuous valued variables, as shown in Figure 7 when T_2 fires, the value of u changes instantly to $u + c$.

Figure 7 Generation of step functions



Event generation from continuous state variables: in this case continuous state variables govern the firing of discrete transition in two alternative ways illustrated in Figure 8. Transition T_1 is activated if $x_1 > c_1$ and T_2 is activated if $x_2 < c_2$. The last condition is graphically denoted by an inhibitor arc that connects place P_2 with transition T_2.

Figure 8 Event generation from continuous state variables



The Hybrid Dynamical Nets is a powerful method of formulating hybrid system problems that allows integrating continuous and discrete dynamics. However, the complexity of hybrid system can result in highly complicated structures difficult to manage using this paradigm.

3.3 Representation of hybrid systems by extended continuous models

This approach represents an extension of continuous models based on differential equation systems to include discrete variables that give discontinuous behaviour. The resulting modelling form for a discrete time hybrid system is the following (Bemporad et al., 2002b):

$$\begin{aligned} &x(k+1) = f(x(k),u(k),w(k)) \\ &y(k) = g(x(k),u(k),w(k)) \\ &0 \le h(x(k),u(k),w(k)) \end{aligned} \qquad (2)$$

where $u(k) \in R^m$, $x(k) \in R^n$ and $y(k) \in R^l$ denote the input, the state and the output and $w(k) \in R^r$ is a vector of auxiliary variables. The form of the functions f: $R^n \times R^m \times R^r \to R^n$, g: $R^n \times R^m \times R^r \to R^l$, h: $R^n \times R^m \times R^r \to R^q$ determines different classes of hybrid systems such as Piecewise Affine (PWA) systems, Mixed Logical Dynamical (MLD) systems and Linear Complementary (LC) systems. The following systems are assumed subsets of the general class of hybrid systems represented by Equation (2).

3.3.1 Piecewise affine systems

PWA systems are a class of hybrid systems characterised by a straightforward structure defined as a set of affine dynamics. This structure is obtained by dividing the state space into polyhedral regions, each of them linked to a linear state-update equation (Morari et al., 2003).

$$\begin{aligned} &x(k+1) = A_i x(k) + B_i u(k) + f_i \\ &y(k) = C_i x(k) + D_i u(k) + g_i \\ &\text{if} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \Omega_i \triangleq \left\{ \begin{bmatrix} x \\ u \end{bmatrix} : H_i x + J_i u \le K_i \right\} \quad i = 1,\ldots,s \end{aligned} \qquad (3)$$

where $x \in R^{n_c} \times \{0,1\}^{n_l}$, $u \in R^{m_c} \times \{0,1\}^{m_l}$, Ω_i is a polyhedral partition of the sets of the state plus input space $\Omega \subset R^{n+m}$, $n = n_c + n_l$, $m = m_c + m_l$.

This model is suitable to describe a large number of physical processes like discrete-time linear systems with static piecewise-linearities, discrete-time linear systems with logic states and inputs or switching systems. A drawback of the PWA formulation is in the fact that PWA systems are a particular class of non-linear systems, therefore linear control theory can not be applied directly to design the controllers (Morari et al., 2003). For this reason in Bemporad and Morari (1999) a MLD framework is presented.

3.3.2 Mixed logical dynamical systems

MLD systems are described by linear dynamic equations subject to inequalities involving continuous and binary variables. The key idea consists of describing the evolution of continuous variables through linear dynamic equations and discrete variables through propositional logic statements. The Boolean variables characterising the logic part are transformed into 0–1 integers and embedded in the state equations by expressing them as mixed-integer linear inequalities. The general MLD form is the following (Bemporad and Morari, 1999):

$$\begin{aligned} x(k+1) &= Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \\ y(k) &= Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \\ E_1x(t) &+ E_2u(t) + E_3\delta(k) + E_4z(k) \le E_5 \end{aligned} \tag{4}$$

where $x = [x_c\ x_l]$ is the state of the system, whose component $x_c \in R^{n_c}$ represents the continuous part of the state and $x_l \in \{0,1\}^{n_l}$ represents the logical or discrete part of the state, $n = n_c + n_l$, $y = [y_c\ y_l]$ is the output vector with $y_l \in \{0,1\}^{l_l}$, $l = [l_c\ l_l]$, $u = [u_c\ u_l]$ is the input signal with $u_c \in R^{m_c}$, $u_l \in \{0,1\}^{m_l}$, $m = m_c + m_l$, $\delta \in \{0,1\}^{r_l}$ is the binary variable that represents auxiliary logic and $z \in R^{r_c}$ is a continuous variable given by:

$$z(k) = \delta(k)x(k) \tag{5}$$

Finite state machines, piecewise linear systems, systems with mixed discrete/continuous inputs and state are examples of systems that could be described by mean of MLD formalism.

3.3.3 Linear complementarity systems

The dynamic behaviour of a LC system evolves in time following a sequence of continuous phases and discrete events which cause a jump in the state vector. It is possible to find analogies between the inequalities that appear in the LC problem of mathematical programming and the inequalities that regulates the incidence of the events in LC systems. Formally, a LC system is governed by the equations (Bemporad et al., 2002b):

$$\begin{aligned} x(k+1) &= Ax(k) + B_1u(k) + B_2w(k) \\ y(k) &= Cx(k) + D_1u(k) + D_2w(k) \\ v(k) &= E_1x(t) + E_2u(t) + E_3w(k) + E_4 \\ 0 \le v(k) &\perp w(k) \ge 0 \end{aligned} \tag{6}$$

where $v(k)$, $w(k)$ $\in R^s$ are called *complementary variables* and $\perp$ signifies the orthogonality of $v(k)$ and $w(k)$.

The study of complementary systems can be motivated by a whole range of interesting applications, such as electrical networks with (ideal) diodes, piecewise linear systems switching control systems, variable structure systems, hydraulic processes with one-way valves and so on (Heemels et al., 2000). A more general complementarity modelling framework that uses methods from LC problem is described in Van der Schaft and Schumacher (1998).

It has been demonstrated previously that PWA systems, LC systems and MLD systems are an equivalent representation for different hybrid systems (Heemels et al., 2001). In theory, their simple structure can be exploited to extend the analysis and control design methodologies developed for linear systems to hybrid systems.

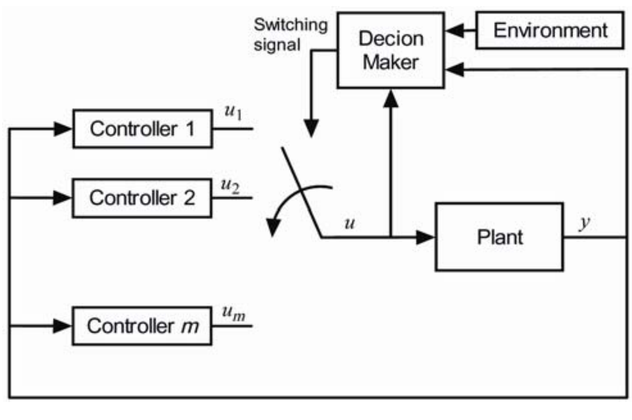
4 Control strategies for hybrid systems

4.1 Switching control

In control theory, the idea of switching between several distinct modes depending on the range of operation has been applied for many years. For example, in non-linear control it is common practice to approximate the non-linear dynamics with a set of linear models defined for various operating regimes and switch among these models. Switching control schemes are also common basis for different control strategies such as gain scheduling, adaptive control, sliding mode control and fuzzy control. As mentioned before, switched systems are a class of hybrid systems that can be described by a family of time driven (continuous time or discrete time) subsystems and a transition system that is either time-based or event-driven. The transition mechanism is usually represented by a set of logic expressions or a DES described with finite automata or Petri nets formalism.

Switched systems that appear in practical applications, especially in an adaptive system, are continuous plants controlled by switching among a family of controllers. Figure 9 shows a graphical representation of this multicontroller architecture. For each model a controller satisfying the performance requirements is designed. At each sampling time only one of the controllers is chosen based on the switching norm and its output is used to drive the plant.

Figure 9 Switching control architecture



The relevance of switching systems arises not only from the fact that they represent a suitable modelling formalism for many complex systems but also because theoretical properties (e.g. existence of a solution, stability, observability, controllability) have been widely explored (Branicky, 1998; Ezzine and Haddad, 1989; Hou et al., 1996; Peleties and DeCarlo, 1991; Wicks et al., 1994).

4.2 Optimal control

Optimal control for hybrid systems is closely related to optimal control of continuous or discrete time systems. The purpose of optimal control is to synthesise a control law with certain properties. These properties are specified in an optimisation criterion or a cost function. For hybrid systems, the cost function in general involves an integral cost accumulated along continuous evolution and switching costs associated with discrete transitions. The cost function can be used to penalise various quantities such as energy consumption, deviation from a desired set point, areas of the state space that are not considered safe. The switching costs can prevent the system from taking *Zeno execution*[1] since infinite mode changes in infinite time would mean infinite cost. Once a cost function for the problem is specified, the control synthesis is transformed to an optimisation problem. In this way, it would be possible to formally state once and for all what good control for a certain process is, and then apply suitable mathematical tools to find the controller that best meets the specifications. However, one problem is that it is often difficult to find a good cost function a priori. Therefore it is common practice to choose a cost function, compute the controller, run experiments, evaluate the results, go back to the first step and adjusting the cost function. Thus, the design procedure has to be iterative. Another potential obstacle is the resulting mathematical problem. It often involves non-convex optimisation of a non-linear function subject to dynamic constraints. This obstacle may be particularly visible in optimal control of hybrid systems where the constraints of combined time-driven and event-driven dynamics in general are very complex. In practical systems, it is often desirable to reduce complexity by optimising a selection of subcomponents of the process to make the overall result 'almost optimal'. It is suggested to use the suboptimal solutions for the actual implementation (Johansen et al., 2000). A different approach to compute upper and lower bounds on the optimal cost is presented in Hedlund and Rantzer (1999). The optimal control problem for a finite set of switching surfaces has been investigated in Sussmann (1999), Xu and Antsaklis (2002) and Shaikh and Caines (2002). Homogeneous regions in the state space that control the switching for a time optimisation problem is presented in Giua et al. (2001). The problem of optimal design of switching surfaces has been addressed in Seatzu et al. (2006). Below various methods for optimal control of hybrid systems are presented.

4.2.1 Maximum principle

The maximum principle was formulated for continuous systems in 1962 (Pontryagin et al., 1962). Given an initial state, it is used to find an optimal trajectory by calculus of variation. The maximum principle gives similar necessary constraints on a trajectory to minimise a cost function in a dynamical system. Each trajectory that satisfies the constraints of the maximum principle is a candidate for optimal trajectory. The candidates can sometimes be found analytically, but often have to be computed by numerical methods, while the optimal one have to be selected by other means. As an important method for optimal control of continuous systems, there have been recent efforts to extend the maximal principle to hybrid systems (Sussmann, 1999). However, the local optimisation of the maximum principle relies on comparisons between neighbouring trajectories. The maximum principle can still be used on continuous evolution of trajectory between switches.

4.2.2 Dynamic programming

The term dynamic programming was introduced by Bellman (1957). The basic idea is the principle of optimality that is, in any state along an optimal trajectory, the remaining part must constitute an optimal trajectory when that state is considered as an initial state. The *value function* or the *cost-to-go function* is central in dynamic programming. It is a function that maps every state onto the cost for a trajectory starting in that state. The principle of optimality can be translated to mathematics in terms of constraints on the optimal value function and the corresponding control signal. The constraint equation, named Hamilton-Jacobi-Bellman (HJB) equation, is a partial differential equation for continuous system case and a differential equation for the discrete systems case. One of the difficulties in applying the HJB equation for systems with continuous dynamics is that the value functions of many optimal control problems are not differentiable. The HJB equation still makes sense, however, if non-classical interpretation of solutions to differential equations are used, such as viscosity solutions (Bardi and Capuzzo-Dolceta, 1997; Bensoussan and Menaldi, 1997) Versions of HJB equation for optimal control of hybrid systems have been formulated in Bensoussan and Menaldi (1997) and Branicky and Mitter (1995).

4.2.3 Model predictive control

Model Predictive Control (MPC) is an advanced optimisation control technique widely applied in many sectors of the process industries (Linkens and Mahafouf, 1994; Qin and Badgwell, 1997; Richalet, 1993; Wen et al., 1997) that has recently been employed for hybrid systems control. Predictive Control methods are characterised by the explicit use of a model to predict the process output over a future prediction horizon, the minimisation of a cost function in order to calculate an optimal control sequence and the adoption of a receding horizon strategy, which means that only the first element of the computed control sequence is applied to the plant. At the next time step a new sequence is computed to replace the previous one. The minimisation problem can be solved either with Mixed Integer Quadratic Programming (MIQP) techniques if $p = 2$ or as a Mixed Integer Linear Program (MILP) if $p = 1$ or ∞ (Bemporad et al., 2002a). The Predictive Control formulation can also explicitly handle constraints, processes with dead times and delays and multivariable interactions.

The predictive control paradigm provides the maximum output from a given plant and takes into account various constraint violations, minimising costs and maximising the efficiency of operations. Nevertheless, MPC involves solving an optimisation problem online at each sampling interval and for this reason it is not suitable for fast and complex systems. It is clear that an efficient implementation of online optimisation tools depends on a quick and repetitive online computation of the optimal control actions (Pistikopoulos et al., 2002).

In order to avoid the online optimisation, a multiparametric programming approach can be used to obtain the optimal state-feedback control law $u(x(k))$ as an explicit function of the state variables $x(k)$ and therefore online optimisation breaks down to simple function evaluations, at regular time intervals, for the given state of the plant to compute the corresponding control actions. The main classes of multiparametric programming are represented by *multiparametric* MILP for performance indexes based on 1-norm or ∞-norm (Borrelli et al., 2003) and *multiparametric* MIQP for performance indexes based on 2-norm (Bemporad et al., 2002a; Borrelli et al., 2003; TØndel et al., 2001). The importance of multiparametric programming in predictive control has huge benefits, as it allows moving off-line the computation of the next command action significant computational savings (Bemporad et al., 2000).

4.2.4 Other methods

Various methods exist that utilise the structure of certain subclass of hybrid systems. A common approach is to split the optimisation algorithm into a discrete part and a continuous part rather than to optimise simultaneously over discrete and continuous variables. An outer loop of the algorithm takes care of discrete event by clever choice of different number of switching and order of discrete modes. The inner loop finds the optimal continuous evolution given the switching scheme dictated by the outer loop. Examples include optimal control of switched systems (Xu and Antsaklis, 2000) and optimal control in the manufacturing model (Pepyne and Cassandras, 2000).

5 Case study: hybrid control design for optimal load sharing

The case study described here analyses the optimal load distribution in a power generation system with several gas turbines (Balbis and Ordys, 2005). The objective of the control problem is to select the optimal number of working units and the optimal load sharing between them in order to minimise the fuel consumption in the system. A similar problem is described in Hansen et al. (1998). However, in that case starting and stopping costs of each unit were not considered. The approach proposed here uses a MLD modelling in a MPC framework.

5.1 Process description

The power generation system here described is powered by gas turbines. In our example, simple cycle gas turbines are adopted. They have the ability to start up fast, producing electrical power in 10 to 30 min. Moreover, gas turbine efficiency and emissions performance improves as load increases. The dynamic equations describing the behaviour of the gas turbine at various operating points are based on the models reported in Ferrari-Trecate et al. (2002) and Ordys et al. (1994). Inputs to the gas turbine are:

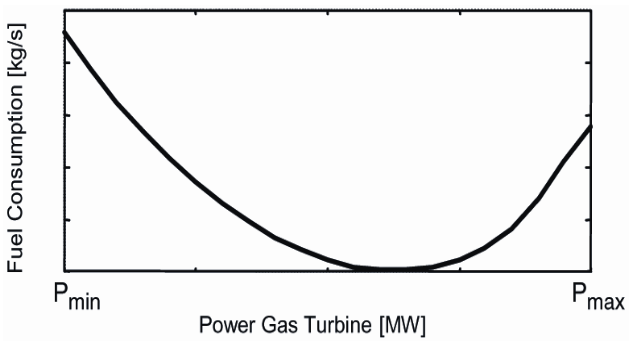
- fuel flow w_f to the combustor [kg/s]
- air flow w_a to the compressor [kg/s]
- the 'on/off' command u_t for the gas turbine.

The outputs of the system are:

- mechanical power delivered P [MW]
- fuel consumption F_c [kg/s].

The fuel consumption of each generation unit is a non-linear function of the power production P and varies according to the relationship $F_c = g(P)$ shown in Figure 10 (Ordys et al., 1994).

Figure 10 Expected fuel consumption versus power



The power production is subject to lower and upper limits:

$$P_{\min} \le P(t) \le P_{\max} \tag{7}$$

The efficiency of operation of a gas turbine depends on the operating mode, with nominal load operation giving the highest efficiency. The efficiency deteriorates rapidly approaching the lower and upper limit of the available produced power. The non-linear function g can be defined in each region by a different affine state-update equation as follows:

$$g_i\left(P(t)\right) = \begin{cases} A_1 P(t) + f_1 & \text{if } P_{\min} \le P(t) \le K_1 \\ A_2 P(t) + f_2 & \text{if } K_1 < P(t) \le K_2 \\ \vdots & \vdots \\ A_s P(t) + f_s & \text{if } K_{s-1} < P(t) \le P_{\max} \end{cases} \tag{8}$$

In addition starting and stopping costs are considered. The increasing operating costs due to fuel consumption during non-dispatched times suggest a review of the traditional start up/shut down scheme. In other words, if the load is expected to change from high value to low value and back again, it may not be optimal to shut down one of the units but rather to reduce the load on all of them. The duration of the start up can vary from a minimum time of 7 min to a maximum time of 14 min. This corresponds to a downtime assumption of 3 hr (normal) or more than 2 hr (cold) shown in Table 1.

Table 1 Start up duration

	Time off [h]	*Start up duration [min]*
Normal start up	[0, 2)	7
Cold start up	[2,+∞)	14

The several start up modes are modelled introducing continuous state variables $x_{\text{off}}(t)$ and $x_{\text{su}}(t)$ that define respectively the uninterrupted time during which the turbine is switched off and the duration of the start up procedures (Ferrari-Trecate et al., 2002). The dynamics of these variables is expressed by the relation:

$$\begin{cases} x_{\text{off}}(t+1) = x_{\text{off}}(t)+1 & \text{if } u_l(t)=0 \\ x_{\text{off}}(t+1) = 0 & \text{if } u_l(t)=1 \end{cases} \tag{9}$$

$$\begin{cases} x_{\text{su}}(t+1) = x_{\text{su}}(t)-1 & \text{if } u_l(t)=1 \\ x_{\text{su}}(t+1) = \begin{cases} 7 & \text{if}\left(0 \le x_{\text{off}}(t) \le 2\right)\wedge\left(u_l(t)=0\right) \\ 14 & \text{if}\left(2 < x_{\text{off}}(t)\right)\wedge\left(u_l(t)=0\right) \end{cases} \end{cases} \tag{10}$$

The power production dynamics are expressed as:

$$P(t) = \begin{cases} k_1 w_f(t) + k_2 w_a(t) + k_3 & \text{if}(x_d(t)<0)\wedge(u_l(t)=1) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The fuel consumption is assumed to be constant during the start up and the shut down modes, and to be a function of the mechanical power during the operative condition:

$$F_c(t) = \begin{cases} K_{\text{su}} & \text{if}\left(u_l(t)=1\right)\wedge\left(x_{\text{su}}(t)\ge 0\right) \\ K_{\text{sd}} & \text{if}\left(u_l(t+1)-u_l(t)<0\right) \\ g_i\left(P(t)\right) & \text{otherwise} \end{cases} \tag{12}$$

The continuous dynamics depending on time, that is, mechanical power production, fuel and air flow and fuel consumption interacting with discrete-valued dynamics, that is, the binary command u_l and the piecewise relations (8)–(12), originate from the hybrid characteristic of the systems.

5.2 Optimisation problem

The optimal load sharing between several generation units can be solved minimising a quadratic function weighting the future errors between the load demand and the predicted power production and the fuel consumption. Given n generators and a load demand P_l [MW] find P_i such that minimises the cost function:

$$\min_{P_i} J = \sum_{k=0}^{N_p} \left\| P_j\left(\tfrac{t+k}{t}\right) - P_l \right\|_{Q_1}^2 + \left\| F_j\left(\tfrac{t+k}{t}\right) \right\|_{Q_2}^2 \tag{13}$$

where N_p is the prediction horizon, N_u is the control horizon, $Q_{1,2}$ is a positive definite error weighting matrices, $P_j(t)$is the total power production and $F_j(t)$ is the total fuel consumption.

The quantities $P_j(t)$ and $F_j(t)$ are given by the equations:

$$P_j(t) = \sum_{i=1}^{n} P_i(t) \tag{14}$$

$$F_j(t) = \sum_{i=1}^{n} F_{c,i}(t) \tag{15}$$

The choice of high values for the matrix Q_1 helps to satisfy the requirement:

$$\sum_{i=1}^{n} P_i(t) = P_l \tag{16}$$

and the presence of the second term in the cost function forces the unit to work in the optimal region regarding the fuel consumption. The overall problem of minimising the cost function subject to the MLD dynamic can be solved using Mixed Integer Quadratic solvers.

5.3 Design study

A case study of a power generation system with two different gas turbines of different powers is considered. The first gas turbine is assumed to have output power varying between 20 and 40 MW and the second gas turbine between 15 and 35 MW. The total load demand is therefore allowed to vary in a range between 0 and 75 MW. For each of the gas turbines the optimal fuel versus power curve shown in Figure 2 has to be identified. For the two gas turbines the non-linear functions $g_1(P_1(t))$ and $g_2(P_2(t))$ are described by the different affine state-update Equations (17) and (18), respectively:

$$g_1\left(P_1(t)\right) = \begin{cases} \left(-0.012P_1(t)+2.73\right) & \text{if } 20 < P_1(t) \le 25 \\ \left(-0.006P_1(t)+2.58\right) & \text{if } 25 < P_1(t) \le 30 \\ 2.4 & \text{if } 25 < P_1(t) \le 30 \\ \left(0.012P_1(t)+1.98\right) & \text{if } 30 < P_1(t) \le 40 \end{cases} \tag{17}$$

$$g_2\left(P_2(t)\right) = \begin{cases} \left(-0.01P_2(t)+2.61\right) & \text{if } 15 \le P_2(t) \le 20 \\ \left(-0.005P_2(t)+2.51\right) & \text{if } 20 < P_2(t) \le 25 \\ 2.3 & \text{if } 25 < P_2(t) \le 30 \\ \left(0.012P_2(t)+2.02\right) & \text{if } 30 < P_2(t) \le 35 \end{cases} \tag{18}$$

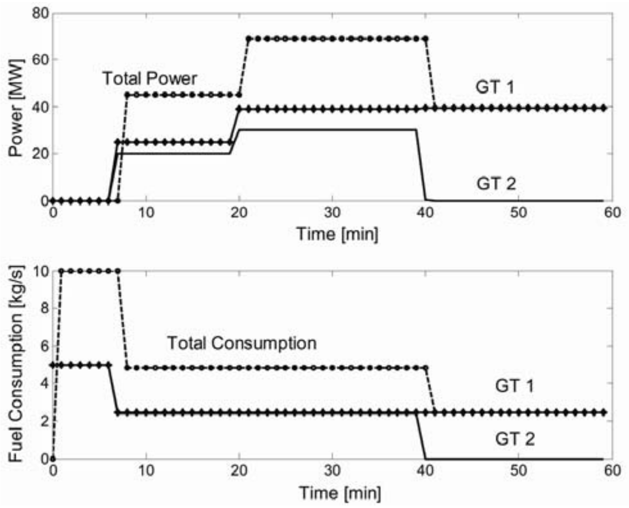
In both units the power production in normal operating conditions is governed by a liner function of air flow and fuel flow. This linear relationship is obtained by linearisation of the model described in Ordys et al. (1994) and is expressed by:

$$P(t) = 2.5w_a(t) + 4.3w_f(t) + 5.5 \tag{19}$$

Other assumptions are that there is no limit on the air flow, the maximum fuel flow is limited to 2.5 kg/s. A simulation horizon of 1 hr and sampling period of 1.2 sec are assumed.

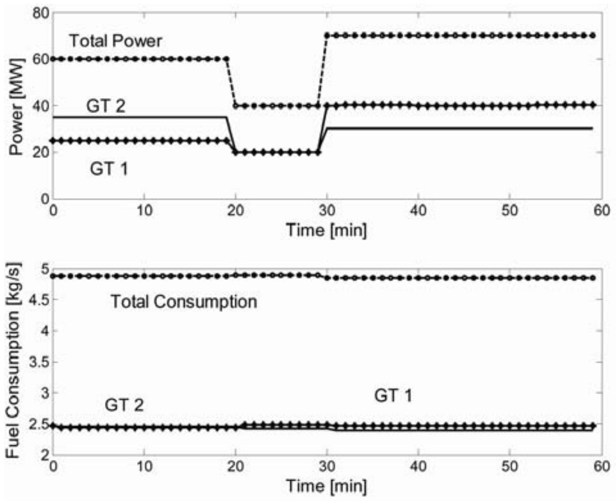
In the first simulation experiment the initial condition for both gas turbines is that they have been off for 2 hr. This means that the start up periods before the production of power has a duration of 7 min, during which the fuel consumption assumes high values. The total load demand is set to vary from 45 to 70 MW and then down to 40 MW. The prediction horizon of 10 min is adopted. The purpose of this first experiment is to show the effect of the redistribution of load between generators when the total load changes (Figure 11).

Figure 11 Optimal load sharing



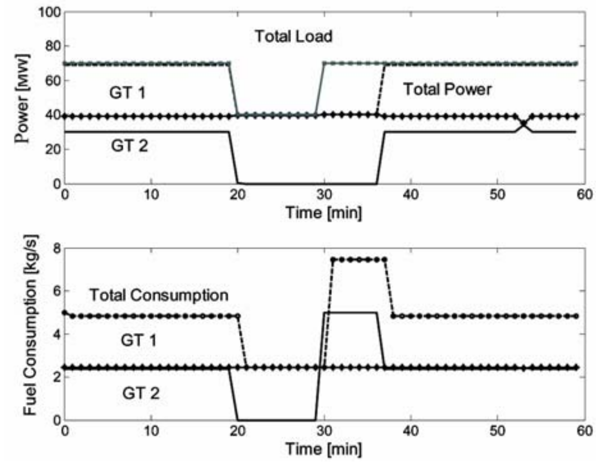
In the second case, at the initial condition the gas turbines are supposed to be working. The total load is set to vary from 60 to 40 MW and then up to 70 MW. As it can be noted, when the load demand goes down to 40 MW the second gas turbine is not switched off. This fact appears to be in contrast with the results obtained in the first simulation, in which the drop in demand corresponded to the shut down of a unit. This behaviour is justified by the adoption of a predictive control strategy, that allows to 'see' the load variation from high value to low value and back again: considering the starting and stopping costs it is not optimal to shut down one of the units but rather to reduce the load on all of them (Figure 12).

Figure 12 Optimal load sharing with $N_p = 10$



In the third simulation experiment a shorter prediction horizon of 8 min is adopted. The total load is set to vary from 70 to 40 MW and then back again to 70 MW. In contrast with the previous case, when the load demand goes down to 40 MW the second gas turbine is switched off. This is due to the fact that the choice of the prediction horizon shorter than the period of time during which the load has a value of 40 MW does not permit to foresee the next increase in the power demand and therefore to take into account the cost of the start up procedure (Figure 13).

Figure 13 Optimal load sharing with $N_p = 8$



6 Conclusions

In this paper some of the methods that can be used to address the problems of modelling hybrid systems have been presented. These include:

- approaches such as hybrid automata and hybrid Petri nets extending discrete formalisms
- approaches merging continuous and discrete event theories
- approaches integrating continuous dynamic components with logic and discrete values, such as MLD formalism, PWA and LC Systems.

The three modelling techniques listed above can be considered equally valuable. The choice of one of the approaches depends on the application to which the model is dedicated. For example, models that involve finite automata or Petri nets are suitable for understanding the behaviour of the system through simulation, but not for analysis and design purpose. Therefore the first approach has been preferred by computer scientists whereas the second and third approaches have been adopted by the control community.

Perhaps the third modelling technique is the one that has reached the highest level of development in both analysis and controller synthesis.

In particular, the MLD framework seems very promising, both for its capacity to describe a wide set of models, ranging from finite state machines to non-linear systems, approximated by piecewise linear functions, and for the equivalence with PWA systems, LC systems and Max–Min-Plus Scaling (MMPS) systems.

A key advantage of the MLD framework is that several problems like control, verification, state estimation and fault detection, observabilty and stability can be formulated and solved as mixed integer linear or quadratic programs. The popularity of this approach has also been increased by the availability of a compiler HYSDEL (Torrisi and Bemporad, 2004) that generates MLD models, given the textual-based descriptions of the systems.

In the above, the MLD modelling associated with Mixed Integer Programming and a Predictive Control strategy were used to solve the problem of minimal fuel consumption for a power system involving several gas turbines. The costs relating to the operating conditions and the start up/shut down periods have all been considered.

As outlined in Section 4, MPC has computational issues due to the online optimisation but the multiparametric approach is an effective technique to reduce complexity issues. However, this approach has limitations in case of large scale systems. The multiparametric mixed integer programming paradigm involves solving an optimisation problem in an extended space, which increases the complexity of the problem considerably, and for many applications might not be possible to find the explicit solution (Baric et al., 2005).

The root of the problem is that there are not sufficiently powerful computers for the analysis and design of extremely complex systems (Morari and Baric, 2006).

Future research will therefore focus on finding a concise description of complex systems in order to avoid exponential growth of data and to provide numerical algorithms which reduce the computational complexity of the optimisation problems.

References

Antsaklis, P., Stiver, J. and Lemmon, M. (1993) 'Hybrid system modeling and autonomous control systems', *Lecture Notes in Computer Science*, Vol. 736, Springer-Verlag, pp.366–392.

Balbis, L. and Ordys, A. (2005) 'Hybrid predictive control design for optimal load sharing', *Proceedings of the 11th IEEE International Conference on Methods and Models in Automation and Robotics*, Miedzyzdroje, Poland.

Bardi, M. and Capuzzo-Dolceta, I. (1997) *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*, Birkhauser Ed.

Baric, M., Grieder, P., Baotic, M. and Morari, M. (2005) 'Optimal control of PWA systems by exploiting problem structure', *IFAC World Congress*, Prague, Czech Republic.

Bellman, R. (1957) *Dynamic Programming*, Princeton University Press.

Bemporad, A. and Morari, M. (1999) 'Control of systems integrating logic, dynamics, and constraints', *Automatica*, Vol. 35, pp.407–427.

Bemporad, A., Borrelli, F. and Morari, M. (2002a) 'On the optimal control law for linear discrete time hybrid systems', *HSCC*, pp.105–119.

Bemporad, A., Heemels, W.P.M.H. and De Schutter, B. (2002b) 'On hybrid systems and closed-loop MPC systems', *IEEE Transactions on Automatic Control*, Vol. 47, No. 5, pp.863–869.

Bemporad, A., Morari, M., Dua, V. and Pistikopoulos, E.N. (2000) 'The explicit solution of model predictive control via multiparametric quadratic programming', *Proceedings of the American Control Conference*, Chicago, Illinois, June.

Bensoussan, A. and Menaldi, J.L. (1997) 'Hybrid control and dynamic programming', *Dynamics of Continuous, Discrete and Impulsive Systems*, Vol. 3, pp.395–442.

Borrelli, F., Bemporad, A., Fodor, M. and Hrovat, D. (2001) 'A hybrid approach to traction control', *Hybrid Systems: Computation and Control*, Vol. 2034 of Lecture Notes in Computer Science, Springer Verlag, pp.162–174.

Borrelli, F., Bemporad, A. and Morari, M. (2003) 'A geometric algorithm for multiparametric linear programming', *Journal of Optimization Theory and Applications*, Vol. 118, No. 3, pp.515–540.

Branicky, M.S. (1998) 'Multiple Lyapunov functions and other analysis tools for switched and hybrid systems', *IEEE Transactions on Automatic Control: Special Issue on Hybrid Systems*, Vol. 43, No. 4, pp.475–482.

Branicky, M.S., Borkar, V.S. and Mitter, S.K. (1998) 'A unified framework for hybrid control: model and optimal control theory', *IEEE Transaction on Automatic Control*, Vol. 43, No. 1, pp.31–45.

Branicky, M.S. and Mitter, S.K. (1995) 'Algorithms for optimal hybrid control', *Proceedings of the 34th IEEE Conference on Decision and Control*, New Orleans, LA.

Cassandras, C.G. and Lafortune, S. (1999) *Introduction to Discrete Event Systems*, Kluwer Academic Publishers.

Davrazos, G. and Koussoulas, N.T. (2001) 'A review of stability results for switched and hybrid systems', *Proceedings of 9th Mediterranean Conference on Control and Automation*.

DeCarlo, R.A., Branicky, M.S., Pettersson, S. and Lennartson, B. (2000) 'Perspectives and results on the stability and stabilizability of hybrid systems', *Proceedings of the IEEE*, Vol. 88, No. 7, pp.1069–1082.

Drath, R. (2002) 'Description of hybrid systems by modified petri nets', in S. Engell, G. Frehse and E. Schnieder (Eds). *Modelling, Analysis and Design of Hybrid Systems*, LNCS 279, pp.15–36.

Drath, R., Engmann, U. and Schwuchow, S. (1999) 'Hybrid aspects of modeling manufacturing systems using modified Petri nets', *Fifth Workshop on Intelligent Manufacturing Systems*, Granado, Brazil.

Elmqvist, H., Cellier, F.E. and Otter, M. (1993) 'Object-oriented modeling of hybrid systems', *Proceedings of the European Simulation Symposium*, pp.31–41.

Elmqvist, H. and Mattsson, S.E. (1997) 'MODELICA – the next generation modeling language. An international design effort', *Proceedings of the 1st World Congress on System Simulation (WCSS'97)*, September, Singapore.

Ezzine, J. and Haddad, A.H. (1989) 'Controllability and observability of hybrid systems', *International Journal of Control*, Vol. 49, pp.2045–2055.

Ferrari-Trecate, G., Gallesty, E., Letizia, P., Spedicato, M., Morari, M. and Antoine, M. (2002) 'Modelling and control of power plants under consideration of lifetime consumpion: a hybrid system approach', *Proceeding of 15th Triennial World Congress of IFAC*, July, Barcelona, Spain.

Ferrari-Trecate, G., Mignone, D., Castagnoli, D. and Morari, M. (2000) 'Hybrid modeling and control of a hydroelectric power plant', *Technical Report AUT00-11*, Automatic Control Laboratory, ETH Zürich, Switzerland.

Gallestey, E., Stothert, A., Castagnoli, D., Ferrari-Trecate, G. and Morari, M. (2003) 'Using model predictive control and hybrid systems for optimal scheduling of industrial processes', *Automatisierungstechnik*, Vol. 51, No. 6, pp.285–294.

Giua, A., Seatzu, C. and Van Der Mee, C. (2001) 'Optimal control of switched autonomous linear systems', *Proceedings of 40th IEEE Conference on Decision and Control*, Orlando, FL, USA, pp.2472–2477.

Hansen, J.F., Ordys, A.W. and Grimble, M.J. (1998) 'A toolbox for simulation of multilayer optimization system with static and dynamic load distribution', *IFAC Symposium on Large Scale Systems: Theory and Applications*, July, Patras, Greece.

Harel, D. (1997) 'Statecharts: a visual formalism for complex systems', *Science of Computer Programming*, Vol. 8, pp.231–274.

Hedlund, S. and Rantzer, A. (1999) 'Optimal control of hybrid systems', *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, Arizona.

Heemels, W.P.M.H., De Shutter, B. and Bemporad, B. (2001) 'Equivalence of hybrid dynamical models', *Automatica*, Vol. 37, No. 7.

Heemels, W.P.M.H., Schumacher, J.M. and Weiland, S. (2000) 'Linear complementarity systems', *SIAM Journal on Applied Mathematics*, Vol. 60, No. 4, pp.1234–1269.

Henzinger, T.A. (1996) 'The theory of hybrid automata', *Proceedings of the 11th Annual Symposium on Logic in Computer Science (LICS)*, IEEE Computer Society Press, pp.278–292.

Henzinger, T., Kopke, P., Puri, A. and Varaiya, P. (1995) 'What's decidable about hybrid automata?' *27th Annual ACM Symposium on Theory of Computing (STOCS)*, pp.372–382.

Hou, L., Michel, A.N. and Yie, H. (1996) 'Stability analysis of switched systems', *Proceedings of the 35th Conference of Decision and Control.*

Hummel, T. and Fengler, W. (2001) 'Design of embedded control system using hybrid petri net', *International Workshop on Discrete-Event System Design*, 27–29 June, Przytok near Zielona Gora, Poland.

Johansen, T., Petersen, I. and Slupphaug, O. (2000) 'Explicit suboptimal linear quadratic regulation with input and state constraints', *Proceedings of IEEE Conference Decision and Control.*

Kowalewski, S. (2002) 'Introduction to analysis and verification of hybrid systems', in S. Engell, G. Frehse and E. Schnieder (Eds). *Modelling, Analysis and Design of Hybrid Systems*, LNCS 279, pp.153–171.

Lemmon, M.D., He, K.X. and Markovsky, I. (1999) 'Supervisory hybrid systems', *IEEE Control Systems*, Vol. 19, No. 4, pp.42–55.

Lennartson, B., Tittus, M., Egardt, B. and Petterson, S. (1996) 'Hybrid systems in process control', *Control Systems Magazine*, Vol. 16, pp.45–56.

Linkens, D.A. and Mahafouf, M. (1994) *Generalized Predictive Control (GPC) in Clinical Anesthesia*, Department of Automatic Control and System Engineering, Sheffield University, UK.

Lunze, J. (2002) 'What is a hybrid system?' in S. Engell, G. Frehse and E. Schnieder (Eds). *Modelling, Analysis and Design of Hybrid Systems*, LNCS 279, pp.3–14.

Lynch, N., Segala, R. and Vaandrager, F. (2003) 'Hybrid I/O automata', *Information and Computation*, Vol. 185, No. 1, pp.103–157.

Möbus, R., Baotic, M. and Morari, M. (2003) 'Multi-object adaptive cruise control', *Hybrid systems: Computation and Control*, Vol. 2623 of Lecture Notes in Computer Science, Springer Verlag, pp.359–374.

Morari, M. and Baric, M. (2006) 'Recent developments in the control of constrained hybrid systems', *Computers and Chemical Engineering*, Vol. 30, Nos. 10–12, pp.1619–1631.

Morari, M., Baotic, M. and Borrelli, F. (2003) 'Hybrid systems modeling and control', *European Journal of Control*, Vol. 9, Nos. 2/3, pp.177–189.

Mosterman, P.J. (1999) 'An overview of hybrid simulation phenomena and their support by simulation packages', *Hybrid Systems: Computation and Control*, NCS1569, Springer, pp.165–177.

Ordys, A.W., Pike, A.W., Johnson, M.A., Katebi, R.M. and Grimble, M.J. (1994) 'Modelling and Simulation of Power Generation Plants', *Advances in Industrial Control*, Springer-Verlag.

Pascal, J.C. and Sahbani, A. (2000) 'Simulation of hybrid systems using stateflow', *14th European Simulation Multiconference (ESM'2000)*, May, Ghent, Belgique.

Peleties, P. and DeCarlo, R. (1991) 'Asymptotic stability of m-switched systems using Lyapunov-like functions', *Proceedings of American Control Conference*, Boston, pp.1679–1684.

Pepyne, D.L. and Cassandras, C.G. (2000) 'Optimal control of hybrid systems in manufacturing', *Proceedings of the IEEE*, Vol. 88, No. 7, pp.1108–1123.

Pistikopoulos, E.N., Dua, V., Bozinis, N.A., Bemporad, A. and Morari, M. (2002) 'On-line optimization via off-line parametric optimization tools', *Computers and Chemical Engineering*, Vol. 26, No. 2, pp.175–185.

Pontryagin, L.S., Boltyanskii, V.G., Gamkrelidze, R.V. and Mishchenko, E.F. (1962) *The Mathematical Theory of Optimal Processes*, New York: Interscience Publishers, John Wiley and Sons.

Pritchett, A.R., Lee, S., Huang, D. and Goldsman, D. (2000) 'Hybrid system simulation for national airspace system safety analysis', in J.A. Joines, R.R. Barton, K. Kang and P.A. Fishwick (Eds). *Proceedings of the 2000 Winter Simulation Conference*.

Qin, S.J. and. Badgwell, T.F. (1997) 'An overview of industrial model predictive control technologies', *Proceedings CPC-V, CACHE. AIChE, AICHE*, pp.232–256.

Richalet, J. (1993) 'Industrial application of model based predictive control', *Automatica*, Vol. 29, No. 5, pp.1271–1274.

Seatzu, C., Corona, D., Giua, A. and Bemporad, A. (2006) 'Optimal control of continuous-time switched affine systems', *IEEETAC*, Vol. 51, No. 5, pp.726–741.

Shaikh, M.S. and Caines, P.E. (2002) 'On trajectory optimization for hybrid systems: theory and algorithms for fixed schedules', *Proceedings of 41th IEEE Conference on Decision and Control*, Las Vegas, USA, pp.1997–1998.

Simulink/Stateflow (1999) The MathWorks Inc., Available at: http://www.mathworks.com.

Stiver, J.A., Antsaklis, P.J. and Lemmon, M.D. (1996) 'A logical DES approach to the design of hybrid control systems', *Mathematical and Computer Modelling*, Vol. 23, Nos. 11/12, pp.55–76.

Sussmann, H.J. (1999) 'A maximum principle for hybrid optimal control problems', *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, Arizona.

Svadova, M. (2001) 'Modelling hybrid dynamic system using hybrid petri nets', *Proceedings of 13th International Conference on Process Control*, June, Štrbské Pleso, Slovakia.

Tomlin, C., Pappas, G. and Sastry, S. (1998) 'Conflict resolution for air traffic management: a study in multiagent hybrid systems', *IEEE Transactions of Automatic Control*, Vol. 43.

TØndel, P., Johansen, T.A. and Bemporad, A. (2001) 'An algorithm for multiparametric quadratic programming and explicit MPC solutions', *Proceedings of 40th IEEE Conference on Decision and Control*, Orlando, Florida.

Torrisi, F. and Bemporad, A. (2004) 'HYSDEL – a tool for generating computational hybrid models', *IEEE Transactions on Control Systems Technology*, Vol. 12, No. 2, pp.235–249.

Van Beek, D.A. and Rooda, J.E. (2000) 'Languages and applications in hybrid modeling and simulation: Positioning of Chi', *Control Engineering Practice*, Vol. 8, No. 1, pp.81–91.

Van der Schaft, A.J. and Schumacher, J.M (1998) 'Complementarity modelling of hybrid systems', *IEEE Transactions on Automatic Control, Special Issue on Hybrid Systems*, Vol. 43, No. 4, pp.483–490.

Vasak, M., Baotic, M., Petrovic, I. and Peric, N. (2004) 'Electronic throttle state estimation and hybrid theory based optimal control', *International Symposium on Industrial Electronics*, ISIE 2004, Ajaccio, France.

Wen, J.T., Seereeram, S. and Bayard, D.S. (1997) 'Nonlinear predictive control applied to spacecraft attitude control', *Proceedings of American Control Conference*, June, Albuquerque, NM.

Wicks, M.A., Peleties, P. and DeCarlo, R.A (1994) 'Construction of piecewise Lyapunov functions for stabilizing switched systems', *Proceedings of IEEE Conference on Decision and Control*, December, Lake Buena Vista, FL.

Xu, X. and Antsaklis, P.J. (2000) 'Optimal control of switched systems: new results and open problems', *Proceedings of the American Control Conference*, Chicago, Illinois.

Xu, X. and Antsaklis, P.J. (2002) 'An approach to optimal control of switched systems with internally forced switchings', *Proceedings of the American Control Conference*, Anchorage, USA, pp.148–153.

Note

[1]An execution of a hybrid system is *Zeno* if infinitely many transitions occur in finite time *T*.