OPTIMIZATION-BASED CONTROL OF HYBRID DYNAMICAL SYSTEMS

Alberto Bemporad

Dip. Ingegneria dell'Informazione, Università di Siena, Italy. Email: `bemporad@dii.unisi.it`.

Abstract: Hybrid systems are heterogeneous systems that exhibit both continuous dynamics (difference or differential equations) and discrete dynamics (automata, logic transitions and switching, piecewise linear mappings, quantized commands, etc.). In this paper we revise a modeling framework for hybrid systems, that is directly tailored to the synthesis of stabilizing model predictive controllers based on combinatorial optimization. Through the use of multiparametric programming, the hybrid optimal control laws can be transformed to a piecewise affine state-feedback closed-form, a very attractive feature for fast-sampling applications.

Keywords: Hybrid Systems, Model Predictive Control, Multiparametric Programming, Automotive Applications.

1. INTRODUCTION

While most of the control theory and tools are based on models describing the evolution of continuous signals according to smooth linear or nonlinear state transition functions, typically differential or difference equations, in many applications, the system to be controlled also contains discrete signals satisfying Boolean relations, if-then-else conditions, on/off conditions, and other type of non-smooth nonlinearities, that interact with the continuous signals.

The lack of a general theory and of systematic design tools for systems having such a heterogeneous dynamical discrete and continuous nature led to a considerable interest in the study of *hybrid systems*. After the seminal work (Witsenhausen, 1966), only in the last decade there has been a renewed interest in the study of hybrid systems, pushed by the recent advent of embedded systems, where a logical/discrete decision device is "embedded" in a physical dynamical environment to change the behavior of the environment itself, and by the availability of several software packages for simulation and numerical/symbolic computation that support the theoretical developments.

Hybrid systems arise in a large number of application areas and are attracting increasing attention by researchers in both academia and industry, for instance in the automotive industry. Moreover, many physical phenomena admit a natural hybrid description, like circuits involving relays or diodes, mechanical systems with impacts, biomolecular networks (Alur *et al.*, 2001), and TCP/IP networks in (Hespanha *et al.*, 2001).

Several modelling frameworks for hybrid systems have appeared in the literature. Each class is usually tailored to solve a particular problem, and many of them look largely dissimilar, at least at first sight (Heemels *et al.*, 2001). Today, there is a widespread agreement in defining hybrid systems

[1] Work partially supported by the European Commission under the HYCON Network of Excellence, contract number FP6-IST-511368, and by the Italian Ministry for Education, University and Research (MIUR) under project "Advanced control methodologies for hybrid dynamical systems".

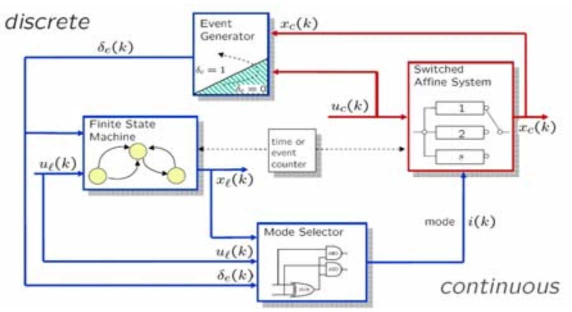

Fig. 1. Discrete-time hybrid automaton (DHA)

as dynamical systems that switch among many operating modes, where each mode is governed by its own characteristic dynamical laws, and mode transitions are triggered by variables crossing specific thresholds (state events), by the elapse of certain time periods (time events), or by external inputs (input events) (Antsaklis, 2000).

The approach described in this paper is based on mixed logical dynamical (MLD) models (Bemporad and Morari, 1999) of hybrid systems, on piecewise affine models (PWA) (Sontag, 1981), and on the quite versatile modeling paradigm of discrete hybrid automata (DHA) (Torrisi and Bemporad, 2004). Such modeling frameworks are briefly reviewed in the following Section 2.

2. MODELS OF HYBRID SYSTEMS

2.1 Discrete Hybrid Automata (DHA)

A DHA (Torrisi and Bemporad, 2004) is the interconnection of an automaton (AUT), which models the discrete dynamics of the hybrid system, with a discrete-time switched affine system (SAS), which models the continuous dynamics of the hybrid system, through an event generator (EG) and a mode selector (MS) (see Figure 1). DHA can be conveniently modeled using the hybrid systems description language HYSDEL (Torrisi and Bemporad, 2004), and converted to MLD or PWA systems in Matlab for analysis and design purposes using the Hybrid Toolbox for MATLAB (Bemporad, 2004*b*).

DHA and PWA are discrete-time deterministic hybrid models. The methodology revised in this paper have been recently extended to *event-based* continuous-time hybrid in (Bemporad *et al.*, 2006), and to *stochastic* hybrid models in (Bemporad and Cairano, 2005).

2.2 Mixed Logical Dynamical (MLD) Systems

DHA can be automatically transformed into the set of linear equalities and inequalities

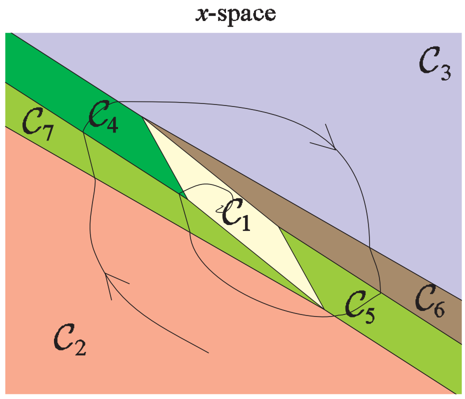

Fig. 2. Piecewise affine system (PWA)

$$x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k) \tag{1a}$$

$$y(k) = Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \tag{1b}$$

$$E_2\delta(k) + E_3z(k) \leq E_1u(k) + E_4x(k) + E_5, \tag{1c}$$

involving both real and binary variables, denoted as the Mixed Logical Dynamical (MLD) model (Bemporad and Morari, 1999), where $x(k) = \begin{bmatrix} x_c(k) \\ x_\ell(k) \end{bmatrix}$ is the state vector, $x_c(k) \in \mathbb{R}^{n_c}$ and $x_\ell(k) \in \{0,1\}^{n_\ell}$, $y(k) = \begin{bmatrix} y_c(k) \\ y_\ell(k) \end{bmatrix} \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$ is the output vector, $u(k) = \begin{bmatrix} u_c(k) \\ u_\ell(k) \end{bmatrix} \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ is the input vector, $z(k) \in \mathbb{R}^{r_c}$ and $\delta(k) \in \{0,1\}^{r_\ell}$ are auxiliary variables, A, B_i, C, D_i and E_i denote real constant matrices, E_5 is a real vector, $n_c > 0$, and p_c, m_c, r_c, n_ℓ, p_ℓ, m_ℓ, $r_\ell \geq 0$. Inequalities (1c) must be interpreted componentwise.

2.3 Piecewise Affine Systems

Piecewise Affine (PWA) systems (Sontag, 1981; Heemels *et al.*, 2001) are a special instance of DHA defined by partitioning the space of states and inputs into polyhedral regions (cf. Figure 2) and associating with each region different linear state-update and output equations:

$$x(k+1) = A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)} \tag{2a}$$

$$y(k) = C_{i(k)}x(k) + D_{i(k)}u(k) + g_{i(k)} \tag{2b}$$

$$i(k) \text{ such that } H_{i(k)}x(k) + J_{i(k)}u(k) \leq K_{i(k)} \tag{2c}$$

where $x(k) \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$ is the state vector at time k, $u(k) \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ is the input vector, $y(k) \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$ is the output vector, $i(k) \in \mathcal{I} \triangleq \{1,\ldots,s\}$ is the current *mode* of the system, the matrices $A_{i(k)}$, $B_{i(k)}$, $f_{i(k)}$, $C_{i(k)}$, $D_{i(k)}$, $g_{i(k)}$, $H_{i(k)}$, $J_{i(k)}$, $K_{i(k)}$ only depend on $i(k)$

($A_{i(k)}$, $B_{i(k)}$, $C_{i(k)}$, $D_{i(k)}$ have zero components in correspondence with $\{0,1\}$ states/outputs), and the inequalities in (2c) should be interpreted component-wise. Each linear inequality in (2c) defines a half-space in $\mathbb{R}^n$ and a corresponding hyperplane, that will be also referred to as *guardline*. Each vector inequality (2c) defines a polyhedron $\mathcal{X}_{i(k)}$ in state+input space $\mathbb{R}^{n+m}$ that will be referred to as *cell*, and the union of such polyhedral cells as *partition*.

PWA systems can model a large number of physical processes, as they can model static nonlinearities through a piecewise-affine approximation, or approximate nonlinear dynamics via multiple linearizations at different operating points.

PWA systems can be modeled as DHA by defining a proper event generator and mode selector, and therefore translated to MLD form. Vice versa, algorithms for translating an MLD model into PWA form were given in (Bemporad, 2004*a*; Geyer *et al.*, 2003).

3. CONTROL OF HYBRID SYSTEMS

Hybrid dynamics are often so complex that a satisfactory feedback controller cannot be synthesized by using analytical tools, and heuristic design procedures usually require trial and error sessions, extensive testing, are time consuming, costly and often inadequate to deal with the complexity of the hybrid control problem properly.

Several control approaches for hybrid systems proposed in the literature rely on the solution of optimal control problems. While algorithms were proposed for continuous-time hybrid models (Xu and Antsaklis, 2004), computations and global solutions are more easily obtained in discrete-time through numerical optimization algorithms.

Model predictive control (MPC) based on discrete-time hybrid dynamical models has emerged as a very promising approach to synthesize feedback controllers for hybrid systems (Bemporad and Morari, 1999). MPC is a widely spread technology in industry for control system design of highly complex multivariable processes (Rawlings, 2000; Camacho and Bordons, 2004; Maciejowski, 2002). The idea behind MPC is to construct an optimal control problem over a finite future horizon, based on an open-loop model of the system, possible constraints on system's variables, and weights on tracking errors and actuator efforts. The problem is translated into an equivalent optimization problem. Then, at each sampling time, the optimization problem is solved by taking the current value of the state vector. The result of the optimization is an optimal sequence of future control moves. Only the first sample of such a sequence is actually applied to the process, as a new optimal control problem based on new measurements is solved over a shifted prediction horizon at the next time step.

Hybrid MPC design is a systematic approach to meet performance and constraint specifications in spite of the aforementioned switching among different linear dynamics, logical state transitions, and more complex logical constraints on system's variables.

The approach consists of modeling the switching open-loop process and constraints using the language HYSDEL (Torrisi and Bemporad, 2004) and then automatically transform the model into the set of linear equalities and inequalities (1).

The associated finite-horizon optimal control problem based on quadratic costs takes the form of a Mixed-Integer Quadratic Programming (MIQP) problem with respect to the optimization variables $U = [u'_0 \ \dots \ u'_{N-1} \ \delta'_0 \ \dots \ \delta'_{N-1} \ z'_0 \ \dots \ z'_{N-1}]'$ and subject to the further restriction that some of the components of U must be either 0 or 1. Both commercial (ILOG, Inc., 2004) and public domain (Bemporad and Mignone, 2000) solvers are available to solve the MIQP problem. When infinity norms are used to specify the performance index for the optimal control problem, the corresponding optimization problem becomes a Mixed-Integer Linear Programming (MILP) problem (Bemporad, 2004*b*), which can be also handled by more efficient public domain solvers such as (Makhorin, 2004), as well as by commercial solvers (ILOG, Inc., 2004).

Regarding complexity, unfortunately MIP's are $\mathcal{NP}$-complete problems. However, the state of the art in solving MIP problems is growing constantly, and problems of relatively large size can be solved quite efficiently. Alternative solution algorithms combining numerical techniques for solving convex programming problems with symbolic techniques for solving constraint satisfaction problems (CSP) were proposed recently in (Bemporad and Giorgetti, 2006), where computational efficiency is gained by taking advantage of CSP solvers for dealing with satisfiability of logic constraints.

3.1 Explicit Hybrid MPC

One of the drawbacks of the hybrid MPC law is the need to solve combinatorial optimization problems on line, with consequent concerns related to computation speed and software complexity.

For MPC based on linear models, to get rid of on-line quadratic programs (QP) an alternative approach to evaluate the MPC law was proposed in (Bemporad *et al.*, 2002a). Rather then solving the QP problem on line for the current vector $x(t)$, the idea is to solve the optimization problem *off line* for all vectors x within a given range and make the dependence of u on x *explicit* (rather than implicitly defined by the optimization procedure). The optimization problem is treated as a multiparametric quadratic programming problem, where $x(t)$ and the reference vector $r(t)$ are the parameters. It turns out that the optimizer of the QP problem is a piecewise affine and continuous function, and consequently the MPC controller can be represented explicitly as

$$u(x) = \begin{cases} F_1 x + g_1 & \text{if } H_1 x \leq k_1 \\ \vdots & \vdots \\ F_M x + g_M & \text{if } H_M x \leq k_M. \end{cases} \tag{3}$$

The controller structure (3) amounts to a look-up table of linear gains (F_i, g_i), where the ith gain is selected according to the set of linear inequalities $H_i x \leq K_i$ that the state vector satisfies. Hence, the evaluation of the MPC controller, once put in the form (3), can be carried out by a very simple piece of control code. In the most naive implementation, the number of operations depends linearly in the worst case on the number M of partitions.

For hybrid MPC problems based on infinity norms, (Bemporad *et al.*, 2000) showed that an equivalent piecewise affine explicit reformulation – possibly discontinuous, due to binary variables – can be obtained through off-line multiparametric mixed-integer linear programming techniques.

The use of linear norms has some practical disadvantages, due to the fact that typically good performance can only be achieved with long time horizons, compared to the horizons requested by quadratic costs. Thanks to the possibility of converting hybrid models (such as those designed through HYSDEL) to their equivalent piecewise affine (PWA) form, an explicit hybrid MPC approach dealing with quadratic costs was proposed in (Borrelli *et al.*, 2005), based on dynamic programming (DP) iterations. A different approach still exploiting the PWA structure of the hybrid model was proposed in (Alessio and Bemporad, 2006), where backwards reachability analysis is exploited (and implemented in the Hybrid Toolbox) to avoid the enumeration of all possible switching sequences over the prediction horizon, and where a post-processing operation eliminate polyhedra (and their associated control gains) that never provide the optimal cost. Typically the DP approach provides simpler explicit solutions when long horizons N are chosen, but on the contrary tends to subdivide the state-space in a larger number of polyhedra than the enumeration approach for short horizons.

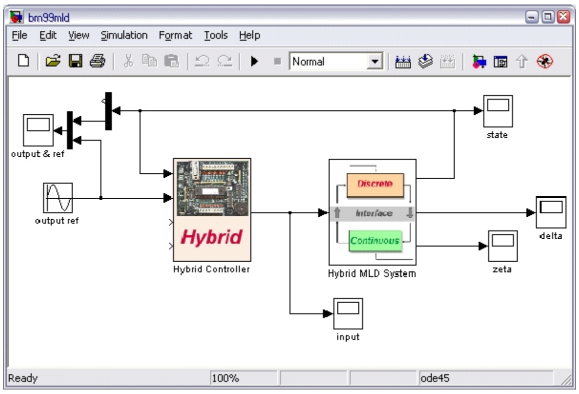

Fig. 3. Simulation of closed-loop hybrid MPC in Simulink using the Hybrid Toolbox

Explicit versions of hybrid MPC are usually useful in practice for systems with few binary variables, one or two inputs, and short prediction horizons, so that the number of polyhedra in the solution (3) is relatively small. Several applications of explicit (hybrid) MPC have been proposed in the automotive domain, see e.g. (Borrelli *et al.*, 2006; Bemporad *et al.*, 2002b; Möbus *et al.*, 2003; Giorgetti *et al.*, 2005).

4. TOOLS FOR DESIGN, EVALUATION AND DEPLOYMENT

The Hybrid Toolbox for MATLAB (Bemporad, 2004b) provides a nice development environment for hybrid MPC. The toolbox can be freely downloaded from `http://www.dii.unisi.it/hybrid/toolbox`. Hybrid dynamical systems described in HYSDEL are automatically converted to MATLAB MLD and PWA objects. MLD and PWA objects can be validated in open-loop simulation, either from the command line or through their corresponding Simulink blocks (see Figure 3). Hybrid MPC controllers based on MILP/MIQP optimization can be designed and simulated, either from the command line or in Simulink.

The Hybrid Toolbox also provides various functions for the design, simulation and code generation of MPC controllers in explicit form. In particular, MPC objects developed through the MPC Toolbox for MATLAB (The Mathworks, Inc.) (Bemporad *et al.*, 2004) can be converted to explicit form through multiparametric quadratic programming. The Hybrid Toolbox also provides functions for manipulation and visualization of polyhedral objects and polyhedral partitions, and contains Simulink blocks to simulate (and possibly rapid prototype) explicit MPC controllers. The Toolbox provides the source C code for evaluating the piecewise affine map (3), whose coefficients are automatically embedded in a header file by appropriate methods. Several demos are available in the Hybrid Toolbox distribution.

5. CONCLUSIONS

Over the last few years an optimization-based methodology has emerged that is able to tackle complex control problems for systems described by hybrid dynamical models. Tools for designing, analyzing, and deploying controllers based on such a methodology are available, and have been extensively used by several students, researchers, and engineers in various application domains. Open problems still remain to come up with numerically viable robust-by-construction schemes, as well as with hybrid state estimation schemes for output-feedback optimization-based control.

REFERENCES

Alessio, A. and A. Bemporad (2006). Feasible mode enumeration and cost comparison for explicit quadratic model predictive control of hybrid systems. In: *2nd IFAC Conference on Analysis and Design of Hybrid Systems*. Alghero, Italy.

Alur, R., C. Belta, F. Ivančić, V. Kumar, M. Mintz, G.J. Pappas, H. Rubin and J. Schug (2001). Hybrid modeling and simulation of biomolecular networks. In: *Hybrid Systems: Computation and Control* (M.D. Di Benedetto and A. Sangiovanni Vincentelli, Eds.). Vol. 2034 of *Lecture Notes in Computer Science*. pp. 19–33. Springer-Verlag.

Antsaklis, P.J. (2000). A brief introduction to the theory and applications of hybrid systems. *Proc. IEEE, Special Issue on Hybrid Systems: Theory and Applications* **88**(7), 879–886.

Bemporad, A. (2004*a*). Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. *IEEE Trans. Automatic Control* **49**(5), 832–838.

Bemporad, A. (2004*b*). *Hybrid Toolbox – User's Guide*. `http://www.dii.unisi.it/hybrid/toolbox`.

Bemporad, A. and D. Mignone (2000). *MIQP.M: A Matlab function for solving mixed integer quadratic programs*. `http://www.dii.unisi.it/~hybrid/tools/miqp`.

Bemporad, A. and M. Morari (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica* **35**(3), 407–427.

Bemporad, A. and N. Giorgetti (2006). Logic-based methods for optimal control of hybrid systems. *IEEE Trans. Automatic Control*. To appear as a regular paper.

Bemporad, A. and S. Di Cairano (2005). Optimal control of discrete hybrid stochastic automata. In: *Hybrid Systems: Computation and Control* (M. Morari and L. Thiele, Eds.). pp. 151–167. Number 3414 In: *Lecture Notes in Computer Science*. Springer-Verlag.

Bemporad, A., F. Borrelli and M. Morari (2000). Piecewise linear optimal controllers for hybrid systems. In: *American Control Conference*. Chicago, IL. pp. 1190–1194.

Bemporad, A., M. Morari and N. L. Ricker (2004). *Model Predictive Control Toolbox for Matlab – User's Guide*. The Mathworks, Inc. `http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/`.

Bemporad, A., M. Morari, V. Dua and E.N. Pistikopoulos (2002*a*). The explicit linear quadratic regulator for constrained systems. *Automatica* **38**(1), 3–20.

Bemporad, A., N. Giorgetti, I.V. Kolmanovsky and D. Hrovat (2002*b*). Hybrid modeling and control of a direct injection stratified charge engine. In: *Symposium on Advanced Automotive Technologies, ASME International Mechanical Engineering Congress and Exposition*. New Orleans, Louisiana.

Bemporad, A., S. Di Cairano and J. Júlvez (2006). Event-based model predictive control and verification of integral continuous-time hybrid automata. In: *Hybrid Systems: Computation and Control*. Santa Barbara, CA.

Borrelli, F., A. Bemporad, M. Fodor and D. Hrovat (2006). An MPC/hybrid system approach to traction control. *IEEE Trans. Contr. Systems Technology* **14**(3), 541–552.

Borrelli, F., M. Baotić, A. Bemporad and M. Morari (2005). Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica* **41**(10), 1709–1721.

Camacho, E.F. and C. Bordons (2004). *Model Predictive Control*. Advanced Textbooks in Control and Signal Processing. 2nd ed.. Springer-Verlag. London.

Geyer, T., F.D. Torrisi and M. Morari (2003). Efficient Mode Enumeration of Compositional Hybrid Models. In: *Hybrid Systems: Computation and Control* (A. Pnueli and O. Maler, Eds.). Vol. 2623 of *Lecture Notes in Computer Science*. pp. 216–232. Springer-Verlag.

Giorgetti, N., A. Bemporad, I.V. Kolmanovsky and D. Hrovat (2005). Explicit hybrid optimal control of direct injection stratified charge engines. In: *Proc. IEEE Int. Symp. on Industrial Electronics*. Dubrovnik, Croatia. pp. 247–252.

Heemels, W.P.M.H., B. De Schutter and A. Bemporad (2001). Equivalence of hybrid dynamical models. *Automatica* **37**(7), 1085–1091.

Hespanha, J. P., S. Bohacek, K. Obraczka and J. Lee (2001). Hybrid modeling of TCP congestion control. In: *Hybrid Systems: Computation and Control* (M.D. Di Benedetto and A. Sangiovanni Vincentelli, Eds.). Vol. 2034 of *Lecture Notes in Computer Science*. pp. 291–304. Springer-Verlag.

ILOG, Inc. (2004). *CPLEX 9.0 User Manual*. Gentilly Cedex, France.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice Hall. Harlow, UK.

Makhorin, A. (2004). *GLPK (GNU Linear Programming Kit) User's Guide*.

Möbus, R., M. Baotić and M. Morari (2003). Multi-objective adaptive cruise control. In: *Hybrid Systems: Computation and Control* (O. Maler and A. Pnueli, Eds.). number 2623 In: *Lecture Notes in Computer Science*. Springer-Verlag. pp. 359–374.

Rawlings, J.B. (2000). Tutorial overview of model predictive control. *IEEE Control Systems Magazine* pp. 38–52.

Sontag, E.D. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Trans. Automatic Control* **26**(2), 346–358.

Torrisi, F.D. and A. Bemporad (2004). HYSDEL — A tool for generating computational hybrid models. *IEEE Trans. Contr. Systems Technology* **12**(2), 235–249.

Witsenhausen, H. (1966). A class of hybrid-state continuous-time dynamic systems. *IEEE Trans. Automatic Control* **11**(2), 161–167.

Xu, X. and P.J. Antsaklis (2004). Optimal control of switched systems based on parameterization of the switching instants. *IEEE Trans. Automatic Control* **49**(1), 2–16.