ADAPTIVE CRUISE CONTROLLER DESIGN: A COMPARATIVE ASSESSMENT FOR PWA SYSTEMS

Daniele Corona * **Bart De Schutter** *

* *Delft Center for System and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands*

Abstract: We propose the design of an *adaptive cruise controller* (ACC) of a Smart vehicle as a benchmark set up for methods developed for piecewise affine (PWA) systems based on model predictive control (MPC) arguments. The control law of the system aims at achieving a trade-off between tracking and fuel consumption while guaranteeing specific *constraints*, related to physical limitations, safety/comfort issues, environment protection and energy saving. In this paper we consider some PWA MPC control design methods, an on-line, an off-line and a robust on-line and compare them with an on-line linear approximation and an off-line gain scheduling approach. The results will be briefly described and the algorithms will be tested in terms of key issues for real implementation and accuracy of the solution.

Keywords: piecewise affine systems, model predictive control, engine control, multi-parametric mixed integer linear programming.

1. INTRODUCTION

Recently, significant efforts of the control system and computer science communities have been devoted to the study and design of hybrid systems. Within this class particular attention was paid to piecewise affine (PWA) systems, namely, a finite set of affine system and a switching signal that switches, internally or externally forced, from one affine mode to another.

PWA systems arise from modeling processes that integrate integer/logical behavior with continuous variables or from quantized inputs (Elia and Mitter, 1999), or from the linear spline approximation of nonlinearities (Sontag, 1981). The discontinuities implicitly hidden in the discrete behavior of these systems make the control design a nontrivial task, the complexity of which is additionally increased if constraints are considered.

Nevertheless several methods that aim to design the control law for this class were proposed in the literature and some of them are MPC based. An equivalent model, the mixed logical dynamical (MLD) model (Bemporad and Morari, 1999), is used to compute the control law implicitly or

[1] Supported by (1) the European 6th Framework Network of Excellence "HYbrid CONtrol: Taming Heterogeneity and Complexity of Networked Embedded Systems (HYCON)", contract number FP6-IST-511368, (2) the BSIK project "Transition Sustainable Mobility (TRANSUMO)", and (3) the Transport Research Centre Delft program "Towards Reliable Mobility", and (4) the Dutch Science Foundation (STW), Grant "Model Predictive Control for Hybrid Systems" (DMR. 5675).

Table 1. *Entries of equation* (1).

m: Mass of vehicle	800 kg
c: Viscous coefficient	0.5 kg/m
μ: Friction coefficient (dry asphalt)	0.01
b: Traction force	3700 N
g: Gravity acceleration	9.8 m/s^2
α: Switching velocity	18.75 m/s

explicitly (Borrelli, 2003). Several variants that consider robustness (Kerrigan and Mayne, 2002) or stability properties (Lazar *et al.*, 2005) were also considered. Methods based on the construction of a piecewise Lyapunov function have been developed in (Hedlund and Rantzer, 1999).

Despite the presence of several methods, an applicative comparison test bed that highlights their main features is, to our best knowledge, missing. Therefore in this paper a benchmark set up for the design of MPC for a PWA system is proposed, arising from an application that is focused on the design of an *adaptive cruise controller* (ACC) for a Smart vehicle within physical/safety constraints.

The paper is organized as follows: we first define the PWA model of the system and the problem. This is transformed into a minimization problem with a mixed integer objective function. We then describe the control methods that we use and we present, in the last section, a comparison table.

2. MODEL AND PROBLEM DESCRIPTION

Model. In a basic ACC application 2 cars are driving one after the other (see Figure 1.a). The aim of an ACC is to ensure a minimal separation between the vehicles and a speed adaptation. We assume that the front vehicle communicates its speed and position to the rear vehicle, which has to track them. For the control design purpose we only consider the dynamics of the rear vehicle, the model of which is:

$$m\ddot{s}(t) + (c\dot{s}^2(t) + \mu\, mg)\text{sgn}(\dot{s}(t)) = bu(t), \quad (1)$$

where $s(t)$ is the position, b is the traction force, $u(t)$ is the normalized throttle/brake. Viscous and static frictions are considered, braking will be simulated by applying a negative u. Numerical values are given in Table 1.

Model (1) is valid as long as the speed is significantly different from zero, hence we impose that the velocity is always above a minimum value.

A least square approximation (Figure 1.b) of the nonlinear friction curve $V = cv^2$ leads to a PWA system:

$$m\ddot{s}(t) + c_i\dot{s}(t) + f_i = bu(t), \quad i = 1, 2.$$

Table 2. *Values of the constraints.*

$x_{2,\min}$: Min. velocity	5.0 m/s
$x_{2,\max}$: Max. velocity	37.5 m/s
d_{safe}: Max. position overshoot	5.0 m
a_{acc}: Max. acceleration	2.5 m/s^2
a_{dec}: Min. acceleration	-1 m/s^2
ξ: Comfort jerk	2.0 m/s^3
$u_{\max}$: Max. throttle/brake	1
Δ_u: Max. throttle/brake variation	0.2

Mode $i = 1(2)$ is active when $\dot{s}(t) < (\geq)\alpha$. The coefficients c_i, f_i are derived using the data shown in Figure 1.b [2].

The discrete-time state-space representation (sampling time $T = 1s$, *zero order hold*) is

$$x(k+1) = A_i x(k) + B_i u(k) + F_i, \quad x_2(k) < \alpha \quad (2)$$

with

$$A_1 = \begin{bmatrix} 1 & 0.97 \\ 0 & 0.99 \end{bmatrix}, \; B_1 = \begin{bmatrix} 2.31 \\ 4.61 \end{bmatrix}, \; F_1 = -\begin{bmatrix} 0.05 \\ 0.10 \end{bmatrix},$$
$$A_2 = \begin{bmatrix} 1 & 0.98 \\ 0 & 0.96 \end{bmatrix}, \; B_2 = \begin{bmatrix} 2.28 \\ 4.54 \end{bmatrix}, \; F_2 = \begin{bmatrix} 0.22 \\ 0.44 \end{bmatrix},$$

where $x_1(k)$ is the position and $x_2(k)$ is the velocity of the rear vehicle.

Constraints. Safety, comfort and economy or environmental issues, as well as limitations on the model, constrain the performance of the system. Since some methods require bounded variables we assume a minimum ($x_{1,\min} = 0\,m$) and a maximum ($x_{1,\max} = 2000\,m$) position. In particular we consider limitations on the state $x(k)$ and on the input $u(k)$. More precisely, $\forall\, k$,

$$\begin{array}{c} x_{\min} \leq x(k) \leq x_{\max} \\ x_1(k) \leq \eta_1(k) + d_{\text{safe}} \\ a_{\text{dec}}T \leq x_2(k+1) - x_2(k) \leq a_{\text{acc}}T \\ |x_2(k+1) - 2x_2(k) + x_2(k-1)| \leq \xi T^2. \end{array} \quad (3)$$

The above equations express, respectively, the operative range of the state, the tracking of the leading vehicle trajectory $\eta(k) = [\eta_1(k), \eta_2(k)]^{\mathrm{T}}$ within a given overshoot d_{safe}, bounds on acceleration and jerk. We have, $\forall\, k$, for the control input:

$$\begin{array}{rl} |u(k)| & \leq u_{\max} \\ |u(k+1) - u(k)| & \leq \Delta_u. \end{array} \quad (4)$$

Numerical values are listed in Table 2. Note that although some of these constraints may be violated without causing major damages, i.e., collision or engine breakdown, we decided to consider all of them as *hard*.

[2] For the sake of simplicity we only consider one breakpoint, leading to a PWA composed of two operating modes. A finer approximation is also possible, by setting more than one breakpoints on the nonlinear curve.

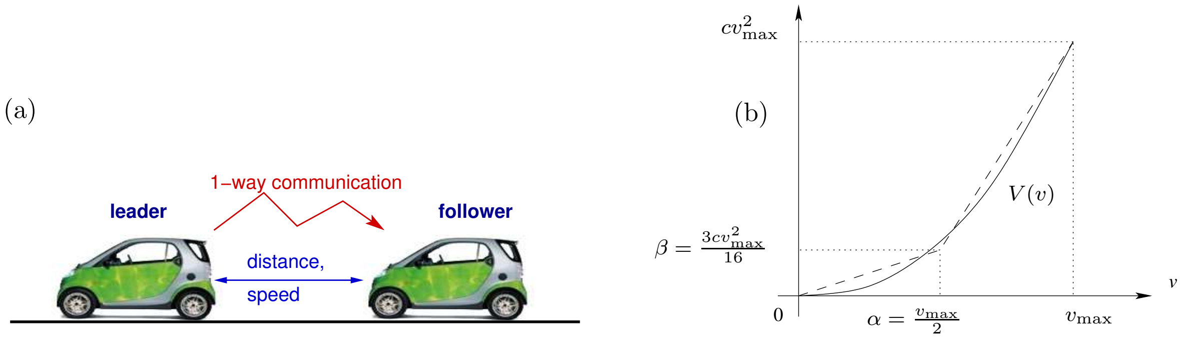

Fig. 1. *(a) ACC set up and (b) nonlinear to PWA approximation.*

Optimal control problem. The control signal $u(k)$ is designed by minimizing the following criterion, in an MPC receding horizon fashion.

$$\min_{\tilde{u}(N_{\mathrm{p}})} J(\theta(k), \tilde{u}(N_{\mathrm{p}})) \triangleq \sum_{j=1}^{N_{\mathrm{p}}} ||Q\varepsilon(k+j)||_1 + ||Ru(k+j-1)||_1, \tag{5}$$

s.t., (2), (3) and (4). Here $\varepsilon(k+j) = x(k+j) - \eta(k+j)$ is the *tracking error*, $\tilde{u}(N_{\mathrm{p}}) = [u(k), \ldots, u(k+N_{\mathrm{p}}-1)]$ the sequence of control inputs, Q, R are weight matrices, $\theta(k) \triangleq [u(k-1), x(k-1)^{\mathrm{T}}, x(k)^{\mathrm{T}}, \eta(k+1)^{\mathrm{T}}, \ldots, \eta(k+N_{\mathrm{p}})^{\mathrm{T}}]^{\mathrm{T}}$ is a set of parameters. In this framework the front vehicle communicates the prediction of N_{p} samples ahead of its trajectory.

Note that an appropriately tuned shorter control horizon $N_{\mathrm{c}} < N_{\mathrm{p}}$ may also be studied, i.e., $u(k+j) = u(k+N_{\mathrm{c}}-1)$, $j = N_{\mathrm{c}}, \ldots, N_{\mathrm{p}}-1$. This choice has the general advantage of reducing the number of variables and of providing a smoother solution. Nevertheless here we only consider $N_{\mathrm{p}} = N_{\mathrm{c}}$ because a tuning issue was beyond the scope of this paper. The ℓ_1-norm allows the use of (mixed integer) linear programming and usually results in better performance than the ℓ_∞-norm (Borrelli, 2003).

3. DESIGN METHODS

In this section we describe the design methods that we have used on the problem above. For all methods we employed the weight matrices $Q = \mathtt{diag}(0.8, 0.1)$ and $R = 0.01$.

Since some of these methods require the transformation of the PWA system into a *mixed logical dynamical* (MLD) form, we show for completeness how this was performed in our setup.

PWA to MLD transformation. The PWA system (2) is transformed into an MLD system by the introduction of a binary variable $\delta(k)$ (Bemporad and Morari, 1999). The value of $\delta(k)$ equals 0(1) when the active moden is system 1(2). Hence the new model of the system is:

$$x(k+1) = A_1 x(k) + Lv(k) + F_1, \tag{6}$$

where $L = [A_2 - A_1 | B_2 - B_1 | F_2 - F_1 | B_1]$ and $v(k) = [z(k)^{\mathrm{T}}, y(k), \delta(k), u(k)]^{\mathrm{T}}$ (with $z(k) = x(k)\delta(k)$, $y(k) = u(k)\delta(k)$, $\delta(k) \in \{0,1\}$) is the auxiliary *mixed logical* control input. The variables $z(k), y(k)$ are nonlinear, but they can be converted into equivalent mixed-integer linear inequalities (Bemporad and Morari, 1999):

$$\begin{gathered} x_{\min}\delta(k) \le z(k) \le x_{\max}\delta(k) \\ -x_{\max}(1-\delta(k)) \le z(k) - x(k) \le -x_{\min}(1-\delta(k)) \\ |y(k)| \le u_{\max}\delta(k) \\ |y(k) - u(k)| \le u_{\max}(1-\delta(k)). \end{gathered} \tag{7}$$

The switching condition leads to

$$\begin{gathered} -\delta(k)(v_{\min} - \alpha) \le x_2(k) - v_{\min} \\ \delta(k)(\alpha - v_{\max}) \le -x_2(k) + \alpha. \end{gathered} \tag{8}$$

Method 1: on-line direct approach. This method (Bemporad and Morari, 1999) solves problem (5) s.t. the MLD model (6), the additional constraints (7) and (8). This is appropriately converted into a mixed integer linear program (MILP) as it follows:

$$\begin{aligned} J(\theta(k))^* = &\min_{\tilde{y}} \; c'\tilde{y} \\ &\text{s.t.} \;\; E\tilde{y} \le G + E_\theta \theta(k), \end{aligned} \tag{9}$$

where the variable $\tilde{y}$ includes the prediction of the control variable $\tilde{v}$ and the dummy variables introduced to convert the ℓ_1 problem into a linear problem. In a receding horizon fashion, the problem is solved on-line and the best control action is computed at step k. At the next step $k+1$ the set of parameters is updated and a new optimization problem is formulated.

Method 2: off-line explicit approach. Problem (9) can be solved parametrically, as proposed in (Bemporad *et al.*, 2000), based on the theoretical results described in (Dua and Pistikopoulos, 2000). This leads to an mp-MILP (multi parametric mixed integer linear programming) as described in (Borrelli, 2003), Chapter 8.

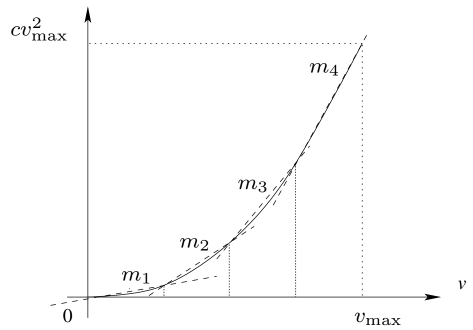

Fig. 2. *Approximation of the nonlinear curve via a gain scheduling approach; m_1, m_2, m_3, m_4 indicate the different affine models.*

The mp-MILP solver provides a partition of convex polyhedra in the parameter space θ, and a PWA control law of the form $\tilde{y}(\theta) = F_i\theta + G_i$, that serves as a *look-up table* during the real time evolution. The software we used for this case is part of the `Multi-Parametric-Toolbox release 2.5` (Kvasnica *et al.*, 2004).

Method 3: on-line linear approximation. In alternative to the hybrid optimization we also considered a locally linear approximation approach. The method consists in using a *locally* linearized prediction model. As detailed in (Beccuti *et al.*, 2005), this technique is appealing for relatively slow processes. The manipulated variables are computed on the base of the linear model, that approximates the nonlinear friction force $V = cv^2$ with its tangent line, i.e., $V = cv^2 \approx 2cv_tv - cv_t^2$.

Method 4: off-line linear approximation. The previous method also suggests an off-line version, in a gain scheduling fashion. The nonlinear curve depicted in Figure 1.b, is approximated into $M = 4$ affine models m_1, m_2, m_3, m_4 in point to point secant approximation, as illustrated in Figure 2.

For each affine model m_i we solve an off-line mp-LP problem of the form (9). More precisely we construct $M = 4$ look-up tables, each valid for a given range of velocity. In the simulation the controller selects the table according to the current value of the speed.

Method 5: Positively invariant set. This on-line approach (Lazar *et al.*, 2005) is aimed to determine a positively invariant set $\mathbb{X}_{N_\mathrm{p}}$ for the PWA system, within which a strictly linear control action may be used. The cost function of the form

$$J(\theta(k), \tilde{u}(N_\mathrm{p})) \triangleq \sum_{j=1}^{N_\mathrm{p}} ||Q_j\varepsilon(k+j)||_1 + ||Ru(k+j-1)||_1, \quad (10)$$

where $Q_j = Q\ \forall\, j = 1, \ldots, N_\mathrm{p} - 1$, is employed as a candidate Lyapunov function to construct (a) an

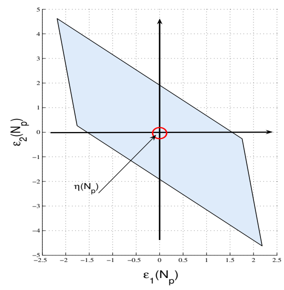

Fig. 3. *Positively invariant set, used as end point constraint in method 5, centered on the last point of the predicted reference trajectory.*

additional end point constraint and (b) a terminal cost matrix Q_{N_p}. The method (Lazar *et al.*, 2005) is conceived for the ℓ_∞-norm, but it can be easily adapted to the ℓ_1-norm.

The reformulation of problem (5) with the additional conditions (a) and (b) has the important advantage of providing a *stabilizing* control law for the hybrid system. Hence we compute an on-line MPC controller over the MLD model (6), enriched with an end point constraint.

In this specific tracking problem, the end point constraint (Figure 3) computed off-line, is *centered* in the N_p-th point of the reference trajectory. The terminal cost matrix is $Q_{N_\mathrm{p}} = \begin{bmatrix} 4.58 & 0.45 \\ 5.14 & 4.15 \end{bmatrix}$.

4. COMPARISON ISSUES AND RESULTS

The five methods are implemented in `Matlab 7`, `Linux 2.4.22` *OS* on an `INTEL pentium 4, 3GHz`. All optimizations, LP and MILP, are performed via the solvers embedded in `TOMLAB v5.1`, except for method 2 (`CDD Criss Cross` solver).

The prediction model for methods 3 and 4 is a linear approximation and an MLD model for methods 1, 2 and 5. Simulation is carried out over the nonlinear continuous time system (1). Methods 3 and 4 are really competitive as long as the system and prediction model are close [3]. On the other hand, methods 1, 2 and 5, based on PWA approximation of the nonlinearity, are not significantly affected by the model mismatch. The integration of equation (1) is done with `Matlab ode45` with `relative` and `absolute tolerances` 1.0×10^{-8}, `max step size` 1.0×10^{-3}.

We have considered two references: (a) a smooth trajectory, with constant velocity and (b) irregular trajectory, with variable velocity, depicted in Figure 4. Some other features of the common setup

[3] The sampling time plays a fundamental role in this specific application.

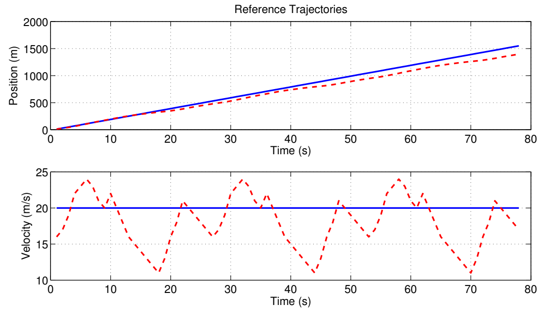

Fig. 4. *Reference trajectories used in simulation, constant (solid) and disturbed (dashed) velocity.*

Table 3. *General data and initial conditions.*

Prediction horizon N_{p}	3
Control horizon N_{c}	3
Sample time T	$1\,s$
Simulation time	$75\,s$
Number of variables	MLD: 24, Linear: 12
of which dummy	9
of which integer	MLD: 3, Linear: –
Number of parameters	11
Number of constraints (MLD)	85 (89 for method 5)
Number of constraints (Linear)	78
Input initial condition	$u(-1) = 0$
State initial condition	$x(0) = [0, 5]^{\mathrm{T}}$
State past condition	$x(-1) = [-5, 5.3]^{\mathrm{T}}$

and initial condition of the simulations are listed in Table 3.

Tables 4 and 5 collect the data obtained by running the methods for both references, constant and time varying velocity. In particular Table 4 shows the performance of the methods with the former and Table 5 with the latter.

We believe that the most interesting aspect (from an application point of view) in Tables 4 and 5 is the computational time, both on-line (average and maximum along the simulation period) and off-line, as well as the memory requirements. These entities are related to the size of the optimization problem, in our particular case to the length of the prediction horizon.

Let us observe that the on-line methods based on the MLD transformation are reasonably fast for low values of N_{p}, and their velocity is comparable with the purely linear method. For higher values of the horizon, the computational time of method 3 is not significantly affected as it is for methods 1 and 5, due to their inner mixed integer structure.

In terms of on-line memory usage we point out that the off-line methods perform quite poorly. In fact the number of the faceted polyhedra grows exponentially with N_{p} (as the number of parameters depends on N_{p}) and each number should be in double precision. However it should be remarked that off-line methods do not require the optimizer on board, which can be relevant in an *integrated design* that also takes into account budget issues.

We also provide the number of times when infeasibility occurred during the simulation period. This index is related to the length of the prediction horizon, and it should decrease as the value of N_{p} increases. The next row of the tables report the cost of the evolution of the obtained control law.

For the on-line methods the item *Maximum tractable* N_{p} is obtained by choosing the smallest N_{p} for which the computational time required by the optimizer is lower than the sampling time and the solution is 100% feasible for the whole simulation period. We observe that this is possible for PWA approximations, but not for methods 3 and 4, for which increasing the prediction horizon does not provide significant improvement on the feasibility of the solution.

Method 5 contains an additional terminal constraint, and it is completely infeasible for low values of N_{p}. The minimum N_{p} that guarantees 100% feasibility is $N_{\mathrm{p}} = 17(19)$ (reference b), but unfortunately for this size the computation time becomes longer than the sample time ($T = 1\,s$).

For the off-line methods, namely 2 and 4, the value of *Maximum tractable* N_{p} is based on reasonable on-line computational time and data storage requirements. As it is shown in Tables 4 and 5 the look-up process to the partition table requires more time than the sampling time T. Nevertheless, the look-up process may be improved by use of appropriate binary search algorithms (Tøndel *et al.*, 2003), the complexity of which is logarithmic.

One major limitation of off-line methods is that the look-up table is only valid for a specific configuration of the system. Thus these methods are not able to handle possible variation of the system parameters values, and a recomputation of the table may be required.

5. CONCLUSIONS

Some methods conceived for computing control laws of PWA systems were used to design an ACC for a Smart vehicle. We have made a comparison among several methods. In particular, methods based on MLD transformation (1 and 5) appear to be very reliable and accurate although they require to perform an on-line optimization. On the other side, off-line methods seem to be more efficient in terms of feasibility, but they suffer of high complexity of the data structure, and their exponential complexity is practically prohibitive

Table 4. *Methods applied on the ACC design, with a constant velocity reference.*

Method	1	2	3	4	5
Time on-line (avg) (s)	0.09	0.90	0.19	1.16	0.09
Time on-line (max) (s)	0.57	1.64	0.16	8.31	0.18
Time off-line (s)	0.16	6.84×10^3	0.002	1.13×10^5	0.21
Memory usage on-line (Mb)	4.43+optimizer	28.5	3.5+optimizer	62	4.53+optimizer
Memory usage off-line (Mb)	0.05	28.5	0.01	62	0.06
Infeasibility (%)	62	1.3	60	27	100
Cost of evolution	337.71	5.38×10^3	474.15	365.32	–
Maximum tractable N_p	18	2	> 30	2	17
Number of regions	–	4541	–	19311	–

Table 5. *Methods applied on the ACC design, with time-varying velocity reference.*

Method	1	2	3	4	5
Time on-line (avg) (s)	0.09	1.13	0.19	1.79	0.07
Time on-line (max) (s)	0.17	1.91	0.39	8.56	0.44
Time off-line (s)	0.09	6.84×10^3	0.003	1.13×10^5	0.21
Memory usage on-line (Mb)	4.45+optimizer	28.5	3.5+optimizer	62	4.45+optimizer
Memory usage off-line (Mb)	0.05	28.5	0.01	62	0.05
Infeasibility (%)	84	1.3	77	56	100
Cost of evolution	334.68	1.59×10^3	456.83	391.41	–
Maximum tractable N_p	15	2	> 30	2	19
Number of regions	–	4541	–	19311	–

for a high number of variables. In a further development we intend to consider also a model with discrete input, the gear shift. We will also consider the performance of some other interesting techniques, (Hedlund and Rantzer, 1999; Raković *et al.*, 2004), that appear to be suitable for the described problem.

Acknowledgment. Authors are thankful to M. Kvasnica (ETH Zürich) for his important advices with the minimization tools and to M. Lazar (TU Eindhoven) for providing the end point constraint for Method 5.

REFERENCES

Beccuti, A.G., T. Geyer and M. Morari (2005). A hybrid system approach to power systems voltage control. In: *Proc. 44th IEEE Conf. on Dec. and Contr.*, Seville, Spain. pp. 6774–6779.

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

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

Borrelli, F. (2003). *Constrained Optimal Control of Linear and Hybrid Systems*. LNCIS 290. Springer–Verlag. Berlin.

Dua, V. and E.N. Pistikopoulos (2000). An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Op. Research* **99**(1–4), 123–139.

Elia, N. and S. Mitter (1999). Quantization of linear systems. In: *Proc. 38th IEEE Conf. on Dec. and Contr.*, Phoenix, Arizona USA. pp. 3428–3435.

Hedlund, S. and A. Rantzer (1999). Optimal control of hybrid systems. In: *Proc. 38th IEEE Conf. on Dec. and Contr.*, Phoenix, USA. pp. 3972–3976.

Kerrigan, E.C. and D.Q. Mayne (2002). Optimal control of constrained, piecewise affine systems with bounded disturbances. In: *Proc. 41th IEEE Conf. on Dec. and Contr.*, Las Vegas, USA. pp. 1552–1557.

Kvasnica, M., P. Grieder, M. Baotić and F.J. Christophersen (2004). *Multi-Parametric Toolbox MPT: User's Manual*. (ETH) Zurich. See: `http://control.ee.ethz.ch/~mpt`.

Lazar, M., W.P.M.H. Heemels, S. Weiland, A. Bemporad and O. Pastravanu (2005). Infinity norms as Lyapunov functions for model predictive control of constrained PWA systems. In: *LNCS: Hybrid Systems: Computation and Control*. number 3414. Springer Verlag. Zürich, Switzerland. pp. 417–432.

Raković, S., E.C. Kerrigan and D.Q. Mayne (2004). Optimal control of constrained piecewise affine systems with state and input–dependent disturbances. In: *Proc. Math. Theory of Networks and Sys.*, Leuven, Belgium.

Sontag, E.D. (1981). Nonlinear regulation: the piecewise affine approach. *IEEE Trans. Automatic Contr.* **26**(2), 346–357.

Tøndel, P., T.A. Johansen and A. Bemporad (2003). Evaluation of piecewise affine control via binary search tree. *Automatica* **39**(3), 945–950.