Performance Analysis of Piecewise Linear Systems and Model Predictive Control Systems

A. Bemporad†‡, F.D. Torrisi†, M. Morari†
†Automatic Control Laboratory, Swiss Federal Institute of Technology - bemporad,torrisi,morari@aut.ee.ethz.ch
‡ Dipartimento di Ingegneria dell’Informazione, Università degli Studi di Siena

Abstract
In their recent paper [2], the authors provided a tool for obtaining the explicit solution of constrained model predictive control (MPC) problems by showing that the control law is a continuous piecewise affine (PWA) function of the state vector. Therefore, the feedback interconnection between the MPC controller and a linear system, or a PWA system (e.g., a PWA approximation of a nonlinear system), is a PWA system. For discrete-time PWA and hybrid systems, we presented an algorithm for verification/reachability analysis in [5]. In this paper, we formulate the performance analysis problem of closed-loop PWA systems (including MPC feedback loops where the prediction model and the plant model could be different) as a reachability analysis problem, and use our algorithm to obtain a tool for characterizing (i) the set of states for which the evolution is feasible, (ii) the domain of stability, (iii) the performance of the closed-loop.

1 Introduction
Model Predictive Control (MPC) has become the accepted standard for complex constrained multivariable control problems in the process industries. Here at each sampling time, starting at the current state, an open-loop optimal control problem is solved over a finite horizon. At the next time step the computation is repeated starting from the new state and over a shifted horizon, leading to a moving horizon policy. The solution relies on a linear dynamic model, respects all input and output constraints, and optimizes a quadratic performance index. The big drawback of MPC was the relatively formidable on-line computational effort which limited its applicability to relatively slow and/or small problems. For discrete time linear time invariant systems with constraints on inputs and states, in [3] the authors developed an algorithm to determine explicitly the state feedback control law which minimizes a quadratic performance criterion. The control law was shown to be piecewise linear and continuous, thus reducing the on-line computation to a simple linear function evaluation instead of an expensive quadratic program.

Therefore, the feedback connection between a linear model and an MPC controller is a piecewise affine (PWA) system, of the form

$$\begin{align*}
x(t+1) &= Ax(t) + B_1d(t) + B_2\delta(t) + B_3z(t) \\
\delta z(t) &\leq E_2\delta(t) + E_3z(t) + E_5
\end{align*}$$

(2a)

(2b)

where $x \in \mathbb{R}^n \times \{0,1\}^{n_b}$ is a vector of continuous and binary states, $d \in \mathbb{R}^d \times \{0,1\}^{d_b}$ are disturbance inputs, and $\delta \in \{0,1\}^{n_b}, z \in \mathbb{R}^{n_c}$ represent auxiliary binary and continuous variables respectively, which are introduced when transforming logic relations into mixed-integer linear inequalities [2], and $A, B_{1-3}, E_{1-5}$ are matrices of suitable dimensions.

MPC techniques guarantee stability of the nominal linear plant through the introduction of stability constraints, which are often removed in practical MPC schemes as they typically deteriorate performance. Moreover, an important issue is to analyze the behavior of the feedback loop when the nominal model and the actual plant model differ, e.g. because of the presence of nonlinearities. Robust MPC tech-
The basic idea of this paper is to check for reachability, for instance, in the context of formal verification of hybrid automata [10]. The reachability analysis issue is well known to be undecidable in the general case. As for stability analysis, such a reachability analysis problem can be formulated as a verification problem the issue of characterizing a set of unsafe states, or possibly providing a counterexample, certifies that all possible trajectories never enter an invariant set around the origin within a finite time, it puts a limitation on the set of states where the constraints are violated. More precisely, we label as asymptotically stable in T steps the trajectories that enter an invariant set around the origin within a finite time T, or as infeasible in T steps the trajectories which enter $X_u$ within that time. Subsets of $X(0)$ leading to neither of the two previous cases are called non-classifiable in T steps. Such a finite-time verifi-
ation problem is decidable, as in the case for many other undecidable problems that can be meaningfully approximated by decidable ones (e.g., the decidable algorithm shown in [1] for analysis of observability is another example of such a philosophy).

The approach followed in this paper is related to the idea of robust simulation [11], which consists of simulating entire set evolutions rather than single trajectories for stability and performance analysis. In [11] the author tests for finite time stability by computing an outer approximation of the reach set via mathematical programming. In particular, an outer approximation is performed at each time step in order to keep the complexity polynomial. In this paper, we present a robust simulation algorithm that, at the expense of extra computation, provides the exact simulation. Although the worst case complexity is still exponential in the time horizon and the number of guardlines, thanks to a set of heuristics, which exploits the piecewise linear nature of the hybrid system, the performance of the algorithm is comparable to the one of [11], and does not suffer for high state-space dimensions.

2 Model Predictive Control

Consider the problem of regulating to the origin the discrete-time linear time invariant system

$$\begin{align*}
x(t+1) &= Ax(t) + Bu(t) \\
y(t) &= Cx(t)
\end{align*}$$

while fulfilling the constraints

$$x_{\min} \leq x(t) \leq x_{\max}, \quad u_{\min} \leq u(t) \leq u_{\max}$$

at all time instants $t \geq 0$. In (3)–(4), $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$ are the state, input, and output vector respectively, $x_{\min} \leq x_{\max}$ ($u_{\min} \leq u_{\max}$) are finite $n(m)$-dimensional vectors, and the pair $(A,B)$ is stabilizable.

Model Predictive Control (MPC) solves such a constrained regulation problem in the following way. Assume that a full measurement of the state $x(t)$ is available at the current time $t$. Then, the optimization problem

$$\min_{U_{t+Na-1}^t} J(U, x(t)) = ||x_{t+Na-1}||_2^2 + \sum_{k=0}^{Na-1} ||x_{t+k+1}||_2^2 + ||u_{t+k}||_R^2$$

subject to $x_{t+k+1} \leq x_{\max}$, $k = 0, \ldots, Na$, $u_{t+k} \leq u_{\max}$, $k = 0, \ldots, Na$, $x_{t+1} = x(t)$, $x_{t+k+1} = Ax_{t+k} + Bu_{t+k}$, $k \geq 0$, $u_{t+k} = Kx_{t+k}$, $N_u \leq N_y$ is solved at each time $t$, where $x_{t+k+1}$ denotes the predicted state vector at time $t+k$, obtained by applying the input sequence $U_{t+Na-1}^t \triangleq u_t, \ldots, u_{t+Na-1}$ to model (3) starting from the current state $x(t)$ measured at time $t$. The state constraints in (5) are defined also for $k = 0$. Although such a constraint is not affected by $U$, it puts a limitation on the set of states for which (5) has a solution, namely, states which are infeasible for (4) are also infeasible for (5). In (5), we assume that $Q = Q' \preceq 0, R = R' \succ 0, P \succeq 0, (Q^T, A)$ detectable (for instance $Q = C^TC$ with $(C,A)$ detectable),
$N_y \geq N_u \geq N_a$, and $K$ is a linear gain. Frequently, $P$ and $K$ are obtained by solving the Riccati equation with weights $Q$, $R$, which amounts to switching the control to the unconstrained LQR after $N_c$ time-steps. 

Let $U^*(t) = \{u^*_t, \ldots, u^*_{t+N_u-1}\}$ be the optimal solution of (5). Then at time $t$

$$u(t) = u^*_t \quad (6)$$

is applied as input to system (3). The optimization (5) is repeated at time $t+1$, based on the new state $x(t+1)$, yielding a moving or receding horizon control strategy.

The stability of MPC feedback loops was investigated by numerous researchers. Stability is, in general, a complex function of the various tuning parameters $N_a$, $N_y$, $N_c$, $Q$, $R$, $P$, and $K$. For applications it is most useful to impose some conditions on $N_y$, $N_c$, $P$, and $K$ so that stability is guaranteed for all $Q \succeq 0$, $R > 0$. Then $Q$ and $R$ can be freely chosen as tuning parameters to affect performance. Sometimes the optimization problem (5) is augmented with a so called “stability constraint”. This additional constraint imposed over the prediction horizon explicitly forces the state vector either to shrink in some norm or to reach an invariant set at the end of the prediction horizon.

Most approaches for proving stability follow in spirit the arguments of Keerthi and Gilbert [12] who establish the fact that under some conditions the value function $V(t) = J(U^*(t), t)$ attained at the minimizer $U^*(t)$ is a Lyapunov function of the system [3].

### 2.1 MPC Computation

By substituting

$$x_{t+k|t} = A^k x(t) + \sum_{j=0}^{k-1} A^j B u_{t-1-j} \quad (7)$$

in (5), the performance index $J(U, x(t))$ can be rewritten in the form

$$\min_U \quad \sum_{i=0}^{k-1} \frac{1}{2} U^T \Psi U + x^T(t) F U$$

subject to $G U \leq W + L x(t) \quad (8)$

where the column vector $U \triangleq [u^*_t, \ldots, u^*_{t+N_u-1}]$ $\in R^s$, $s \triangleq m N_u$, is the optimization vector, $\Psi = \Psi' \succ 0$, and $G$, $F$, $Y$, $W$, $L$ are easily obtained from $Q$, $R$, and (5)–(7). As only the optimizer $U$ is needed, the term involving $Y$ is usually removed from (8).

The optimization problem (8) is a quadratic program (QP). Because the problem depends on the current state $x(t)$, the implementation of MPC requires the on-line solution of a QP at each time step. Although efficient QP solvers based on active-set methods or interior point methods are available, computing the input $u(t)$ demands significant on-line computation effort. For this reason, the application of MPC has been limited to “slow” and/or “small” processes.

In [3] the authors presented a new approach to implement MPC, where all the computation effort is moved off-line. The idea is based on the observation that in (8) the state $x(t) \in R^n$ can be considered a vector of parameters, and (8) as a multi-parametric Quadratic Program (mp-QP). An algorithm to solve mp-QP problems was presented in [3]. Once the multi-parametric problem (8) has been solved off line, i.e., the solution $U^* = f(x(t))$ of (8) has been found, the model predictive controller (5) is available explicitly, as the optimal input $u(t)$ consists simply of the first $m$ components of $U^*$, $u(t) = [I \ 0 \ldots \ 0] f(x(t))$. In [3] the authors also show that the solution $U^* = f(x)$ of the mp-QP problem is continuous and piecewise affine. Clearly, because of (9), the same properties are inherited by the controller, i.e.,

$$u(t) = F x(t) + g_i \quad \text{for} \quad x(t) \in C_i \triangleq \{x: H_i x \leq s_i\}; \quad i = 1, \ldots, s \quad (9)$$

where $\cup_{i=1}^s C_i$ is the set of states for which a feasible solution to (5) exists. Therefore, the closed MPC loop is of the form (1), where $A_i = A + B F_i$, $f_i = B g_i$, $B_i = 0$ (A vector of polyhedrally-bounded additive disturbances $d(t)$ can be taken into account by considering nonzero matrices $B_i$). Note that the form of the closed-loop MPC system remains PWA also when (i) the matrices $A$, $B$ of the plant model are different from those used in the prediction model, and (ii) the plant model has a PWA form. Typically, the MPC law (5) is designed on a linear model obtained by linearizing the nonlinear model of the plant around some operating condition. When the nonlinear model can be approximated by a PWA system (e.g., through multiple linearizations at different operating points or by approximating nonlinear static mappings into piecewise linear functions), the closed-loop formed by the nonlinear plant model and the MPC controller (5) can be approximated by a PWA system as well.

### 3 Performance Characterization Problem

As mentioned in the introduction, determining the stability of PWA systems can be a complex task. Nevertheless, we aim at estimating the domain of attraction of the origin, and the set of initial conditions from which the state trajectory remains feasible for the constraints (4).

As mentioned in the previous section, the nominal MPC closed-loop is an autonomous PWA system. The origin belongs to the interior of one of the sets of the partition, namely the region where the LQ gain $K$ is asymptotically stabilizing while fulfilling the constraints (4), which by convention will be referred to as $C_0$. Denote by $D_{\infty}(0) \subseteq R^n$ the (unknown) domain of attraction of the origin. Given a bounded set $X(0)$ of initial conditions, we want to characterize $D_{\infty}(0) \cap X(0)$.

By construction, matrix $A_0$, associated with the region $C_0$, is strictly Hurwitz and $f_0 = 0$ (in fact, in $C_0$ the feedback gain is the unconstrained LQR gain $F_0 = K$, $g_0 = 0$ [3]). Then we can compute an invariant set in $C_0$. In particular, we compute the maximum output admissible set (MOAS) $X_{\infty} \subseteq C_0$. $X_{\infty}$ is the largest invariant set contained in $C_0$, which by construction of $C_0$ is compatible with the constraints $u_{\min} \leq K x(t) \leq u_{\max}, \ x_{\min} \leq x(t) \leq x_{\max}$. By [9, Th.4.1], MOAS is a polyhedron with a finite number of facets, and is computed through a finite number of linear programs (LP’s) [9].

\[
1 \text{If the effect of perturbations } d(t) \in U \subseteq R^n, \text{ where } U \text{ is a given bounded set of disturbances and } B_0 \neq 0, \text{ has to be taken into account } X_{\infty} \text{ is the largest invariant set under disturbance excitation, and can be computed as proposed in [8].} \]
In order to circumvent the undecidability of stability mentioned above, we give the following

**Definition 3.1** Consider the PWA system (1), and let the origin $0 \in C_0 \triangleq \{x : H_0 x < S_0\}$, and $A_0$ be strictly Hurwitz. Let $X_\infty$ be the maximum output admissible set (MOAS) in $C_0$, which is an invariant for the linear system $x(t+1) = A_0 x(t)$. Let $T$ be a finite time horizon. Then, the set $X(0) \subseteq R^n$ of initial conditions is said to belong to the domain of attraction in $T$ steps $D_T(0)$ of the origin if $\forall x(0) \in X(0)$ the corresponding final state $x(T) \in X_\infty$.

Note that $D_T(0) \subseteq D_{T+1}(0) \subseteq D_\infty(0)$, and $D_T(0) \rightarrow D_\infty(0)$ as $T \rightarrow \infty$. The horizon $T$ is a practical information about the speed of convergence of the PWA system to the origin.

**Definition 3.2** Consider the PWA system (1), and let $X_{\text{infeas}} \triangleq R^n \setminus \cup_{i=1}^{\infty} C_i$. The set $X(0) \subseteq R^n$ of initial conditions is said to belong to the domain of infeasibility in $T$ steps $T(0)$ if $\forall x(T) \in X(0)$ there exists $t, 0 \leq t \leq T$ such that $x(t) \in X_{\text{infeas}}$.

Given a set of initial conditions $X(0)$, we aim at finding subsets of $X(0)$ which are safely asymptotically stable ($X(0) \cap D_T(0)$), and subsets which lead to infeasibility in $T$ steps ($X(0) \cap T(0)$). Subsets of $X(0)$ leading to none of the two previous cases are labeled as non-classifiable in $T$ steps. As we will use linear optimization tools, we assume that $X(0)$ is a convex polyhedral set (or the union of convex polyhedral sets). Typically, non-classifiable subsets shrink and eventually disappear for increasing $T$.

### 3.1 Switching Sequences

Consider the following simple case of evolution of the PWA system (1), where $u(t) = 0, f_i = 0, \forall i = 0, \ldots, s - 1$,

$$x(t) = A_{i(t-1)} A_{i(t-2)} \cdots A_{i(0)} x(0) \tag{10}$$

where in (10) $i(k) \in \{0, \ldots, s - 1\}$ is the index such that $H_{i(k)} x(k) \leq S_{i(k)}, k = 0, \ldots, t - 1$, is satisfied. The previous questions of practical stability can be answered once all switching sequences $I(t) \triangleq \{i(0), \ldots, i(t - 1)\}$ leading to $X_\infty$ or $X_{\text{infeas}}$ from $X(0)$ are known. In fact, for safe stability in $T$ steps it is enough to check that the reach set at time $T, X(T, X(0)) \triangleq A_{i(T-1)} A_{i(T-2)} \cdots A_{i(0)} X(0)$, satisfies the set inclusion $X(T, X(0)) \subseteq X_\infty$ for all admissible switching sequences $I(T)$. However, the number of all possible switching sequences $I(T)$ is combinatorial with respect to $T$ and $s$, and any enumeration method would be impractical. In the next section we show that a verification algorithm can be used to avoid such an enumeration.

### 4 Reachability Analysis of Hybrid Systems

In this section, we recall the verification algorithm presented in [5]. In order to determine admissible switching sequences $I(t)$, the algorithm exploits the special structure of the PWA system (1). This structure allows an easy computation of the reach set as long as the evolution remains within a single region $C_i$. Whenever the reach set crosses a guardline and enters a new region $C_j$, a new reach set computation based on the $j$-th linear dynamics is computed, as shown in Fig. 1(a).

Let $X(0)$ be a convex polyhedral set, and partition it into subregions $X_i(0) \triangleq X(0) \cap C_i, i = 0, \ldots, s - 1$. For all nonempty sets $X_i(0)$, computing the evolution $X(T, X_i(0))$ requires: (i) the reach set $X(t, X_i(0), C_i)$, i.e., the set of evolutions at time $t$ in $C_i$ from $X_i(0)$; (ii) crossing detection of the guardlines, $P_h \triangleq X(t, X_i(0), C_i) \cap C_i \neq \emptyset, \forall t = 0, \ldots, i - 1, i + 1, \ldots, s - 1$; (iii) elimination of redundant constraints and approximation of the polyhedral representation of the new regions $P_h$ (approximation is desirable, as the number of facets of $P_h$ can grow linearly with time); (iv) detection (1) of emptiness of $X(t, P_h, C_i)$ (emptiness happens when all the evolutions have crossed the guardlines), (2) of safe stability $X(t, P_h, C_i) \subseteq X_\infty$, (3) of full infeasibility $X(t, P_h, C_i) \subseteq X_{\text{infeas}}$ (these three will be referred to as fathoming conditions).

#### 4.1 Reach Set Computation

Let the set of initial conditions be defined by the polyhedral representation $X(0) \triangleq \{x : S_0 x \leq T_0\}$. The subset $S_i(t, X(0))$ of $X(0)$ whose evolution lies in $C_i$ for $t$ steps is given by

$$S_i(t, X(0)) = \{x \in R^n : S_0 x \leq T_0, H_i A_i^t x \leq S_i - H_i \sum_{j=0}^{k-1} A_j^t f_i, k = 0, \ldots, t\} \tag{11}$$

As $S_i(t, X(0))$ is a polyhedral set, the reach set $X(t, X_i(0), C_i)$ is a polyhedral set as well. In the presence of input disturbances, $S_i(t, X(0)) = \{x \in R^n : S_0 x \leq T_0, H_i (A_i^t x + \sum_{j=0}^{k-1} A_j^t [B_i d(k-1-j) + f_i]) \leq S_i, k = 0, \ldots, t\}$, is a polyhedron in the augmented space of tuples $(x, d(0), \ldots, d(t - 1))$.

#### 4.2 Guardline Crossing Detection

Switching detection amounts to finding all possible new regions $C_h$’s entered by the reach set at the next time step, i.e., nonempty sets $P_h \triangleq X(t, X_i(0), C_i) \cap C_h, h \neq i$. Rather than enumerating and checking nonemptiness for all $h = 0, \ldots, i - 1, i + 1, \ldots, s - 1$, we can exploit the equivalence between PWA systems and MLD models (2), and solve the switching detection problem via mixed-integer linear programming. More in detail, in the MLD form the condition $x(t) \in C_h$ is associated with the condition $\delta(t) = \delta_h \in \{0, 1\}^r$, for instance $x(t) \in C_3 \iff \delta(t) = [1 \ 0 \ 1]'$. Switching detection

![Figure 1: Reachability Analysis](image-url)
amounts to finding all feasible vectors \( \delta(t) \in \{0,1\}^{\tau} \) which are compatible with the constraints in (2) plus the constraint \( x(t-1) \in \mathcal{X}(t-1, \mathcal{X}_i(0), \mathcal{C}_i) \). Such a problem is a mixed-integer linear feasibility test (MILFT), and can be efficiently solved through standard recursive branch and bound procedures. Thus, on average the MLD form (through the branch and bound algorithm) requires only a very small number of feasibility tests, while the PWA form would require enumerating and solving a feasibility test for all the possible \( s \) regions.

4.3 Approximation of Intersections
The computation of the reach set proceeds in each region \( \mathcal{C}_i \) from each new intersection \( \mathcal{P}_h \). A new reach set computation is started from \( \mathcal{P}_h \), unless \( \mathcal{P}_h \) is contained in some larger subset of \( \mathcal{C}_i \) which has already been explored. As the number of facets of \( \mathcal{P}_h \) can grow linearly with time, we need to approximate \( \mathcal{P}_h \) so that its complexity is bounded (and therefore the computation of the reach set from \( \mathcal{P}_h \) has a limited complexity with respect to the initial region), and checking for set inclusion is a simple task. Hyper-rectangular approximations are the best candidates, as set inclusion with respect to the initial region, and checking for set inclusion is a simple task. Hyper-rectangular approximations are the best candidates, as set inclusion between hyper-rectangles reduces to a simple comparison of the coordinates of the vertices. On the other hand, a crude rectangular outer approximation of \( \mathcal{P}_h \) can lead to explore large regions which are not reachable from the initial set \( \mathcal{X}(0) \), as they are just introduced by the approximation itself. In [4] the authors propose an iterative method for inner and outer approximation which is based on linear programming, and approximates with arbitrary precision polytopes by a collection of hyper-rectangles, as depicted in Fig. 1(a).

4.4 Fathoming
In Sect. 4.1 we showed how to compute the evolution of the reach set \( \mathcal{X}(t, \mathcal{P}_h, \mathcal{C}_i) \) inside a region \( \mathcal{C}_i \). The computation is stopped once one of the following happens: (i) The set \( \mathcal{X}(t, \mathcal{P}_h, \mathcal{C}_i) \) is empty. This means that the whole evolution has left region \( \mathcal{C}_i \); (ii) \( \mathcal{X}(t, \mathcal{P}_h, \mathcal{C}_i) \subseteq \mathcal{X}_\infty \), i.e., all possible evolutions from \( \mathcal{P}_h \) are safely stable, (iii) \( \mathcal{X}(t, \mathcal{P}_h, \mathcal{C}_i) \subseteq \mathcal{X}_\text{infeas} \), i.e., all possible evolutions from \( \mathcal{P}_h \) have violated the constraints in (4). (iv) time \( t > T \). These conditions can be checked through linear programming.

4.5 Graph of Evolution
The result of the exploration algorithm detailed in the previous sections can be conveniently represented on a graph \( G \) (Fig. 1(b)). The nodes of \( G \) represent sets from which a reach set evolution is computed, and an oriented arc of \( G \) connects two nodes if a transition exists between the two corresponding sets. Each arc has an associated weight which represents the time-steps needed for the transition. The graph has initially no arc, and nonempty initial sets \( \mathcal{X}_i(0) \) and \( \mathcal{X}_\infty, \mathcal{X}_\text{infeas} \) as nodes. When a new intersection \( \mathcal{X}(t, \mathcal{X}_i(0), \mathcal{C}_i) \setminus \mathcal{C}_h \) is detected, it is approximated by a collection of hyper-rectangles, as described in Sect. 4.3. Each hyper-rectangle becomes a new node in \( G \), and is connected by a weighted arc from \( \mathcal{X}_i(0) \).

After the verification algorithm terminates, the oriented paths on \( G \) from initial nodes \( \mathcal{X}_i(0) \) to terminal

**Figure 2:** Example (12)

Consider the system \( y(t) = \frac{s+1}{s^2 + s + 2} u(t) \), and sample the dynamics with \( T = 0.2 \text{ s} \). The task is to regulate the system to the origin while fulfilling the constraints \(-1 \leq u(t) \leq 1 \) and \( x(t) \geq [-0.5,0.5] \). To this aim, we design an MPC controller based on the optimization problem

\[
\begin{align*}
\min_{u(t:t+1)} & \quad ||x_{t+2}||^2 + \sum_{k=0}^{1} ||x_{t+k+1}||^2 + 0.1 ||u_{t+k}||^2 \\
\text{subj. to} & \quad -2 \leq u_{t+k} \leq 2, \quad k = 0, 1 \\
& \quad x_{t+k} \geq x_{min}, \quad x_{min} = [-0.5,0.5], \quad k = 0, 1
\end{align*}
\]

where \( P \) is the solution to the Riccati equation (in this example \( Q = [1 \, 0; \, 0 \, 1], \quad R = 0.1, \quad N_y = N_x = N_c = 2 \)). Note that this choice of \( P \) corresponds to setting \( u_{t+k} = K x_{t+k+1} \) for \( k \geq 2 \), where \( K \) is the LQR gain, and minimizes \( \sum_{k=0}^{\infty} x_{t+k}^T K x_{t+k} + 0.1 u_{t+k}^2 \) with respect to \( u_t, u_{t+1} \). The closed loop response from the initial condition \( x(0) = [1 \, 1]^T \) is shown in Fig. 2(a).

The mp-QP problem associated with the MPC law has the form (8) with

\[
\Psi = \begin{bmatrix} 0.7616 & 0.3029 \\ 0.0355 & 0.6079 \end{bmatrix}, \quad F = \begin{bmatrix} 1.2959 & 1.5010 \\ 0.9835 & 0.5171 \end{bmatrix}, \quad W = \begin{bmatrix} 0.5 \\ 0.5 \\ 0 \end{bmatrix}, \quad L = \begin{bmatrix} 0.7830 & -0.1789 \\ 0.3577 & 0.9628 \\ 1 & 0 \end{bmatrix}
\]

The solution was computed by using the mp-QP solver in [3] in 0.66 s on a PC Pentium III 650 MHz running Matlab 5.3, and the corresponding polyhedral partition of the state-space is depicted in Fig. 2(b). The MPC law is
where region #3 corresponds to the unconstrained LQR controller, #1 and #4 to saturation at -1 and +1, respectively, and #2 is a transition region between LQR regions #1 and #3.

Note that the union of the regions depicted in Fig. 2(b) should not be confused with the region of attraction of the MPC closed-loop. For instance, by starting at $x(0) = [3.5 \ 0]^T$ (for which a feasible solution exists), the MPC controller runs into infeasibility after $t = 5$ time steps.

The reachability analysis algorithm described above was applied to determine the set of safely stable initial states and states which are infeasible in $T = 20$ steps (Fig. 3). The algorithm computes the graph of evolutions in 115 s on a Pentium II 400 running Matlab 5.3.

### 6 Conclusions and Acknowledgments

In this paper we proposed a technique for performance assessment of MPC closed-loop systems which is based on reachability analysis of hybrid systems. The approach can be immediately extended to set-point tracking problems and disturbance rejection, where parametric analysis with respect to set-point/disturbance values can be performed in order to determine the set of initial states which leads to safe evolutions for a given set-point/disturbance, or vice versa all the set-points/disturbance which can be safely commanded from a given set of initial states. The approach also allows the robust analysis of safe stability against norm-bounded disturbances.

This research has been supported by the Swiss National Science Foundation and Esprit Project 26270 VHS (verification of Hybrid Systems).

### References


