ON SUBOPTIMAL CONTROL DESIGN FOR HYBRID AUTOMATA USING PREDICTIVE CONTROL TECHNIQUES

Yan Pang [*], **Michael P. Spathopoulos** [*], **Joerg Raisch** [**]

[*] *Department of Mechanical Engineering, University of Strathclyde, Glasgow G1 1JX,UK*
{*yan.pang,michael.spathopoulos*}*@strath.ac.uk*
[**] *Otto-von-Guericke Universitaet Magdeburg and Max-Planck-Institut fuer Dynamik komplexer technischer Systeme*
raisch@mpi-magdeburg.mpg.de

Abstract: In this paper we propose an on-line design technique for the target control problem of hybrid automata. First, we compute off-line the shortest path, which has the minimum discrete cost, from an initial state to the given target set. Next, we derive a controller which successfully drives the system from the initial state to the target set while minimizing a cost function. The model predictive control (MPC) technique is used when the current state is not within a guard set, otherwise the mixed-integer predictive control (MIPC) technique is employed. An on-line, semi-explicit control algorithm is derived by combining the two techniques and applied on a high-speed and energy-saving control problem of the CPU processing. *Copyright© 2005 IFAC*

Keywords: Hybrid Automata, Optimization, Target Control, Predictive Control

1. INTRODUCTION

Model predictive control (MPC) exhibits attractive features that makes it interesting and relevant to hybrid systems. The model predictive control (MPC) and the mixed integer predictive control (MIPC) for hybrid systems have been studied by a number of researchers *e.g.* (Bemporad *et al.*, 2001; Lazar and Heemels, 2003). The hybrid models adopted by them are piecewise affine (PWA) systems or mixed logical dynamic (MLD) systems. At the same time the MPC for switched hybrid systems has been discussed in (Stursberg and Engell, 2002).

Here, we use the hybrid automata formalism in order to model hybrid systems. Therefore, the transition guards are full dimensional sets rather than hyper-planes. This kind of modelling introduces non-determinism, *e.g.* within the guard set the controller can make the choice between switching or idling.

Considering the high computational complexity of the MPC on-line algorithm presented in (Bemporad *et al.*, 1999), we formulate a semi-explicit(sub-optimal) method that reduces the computational burden. For this, we remove the off-line choices for the switching part, by selecting the shortest discrete path. It is then shown that the shortest path can be used to derive a semi-explicit algorithm for hybrid automata, instead of solving the mixed integer programming problem (NP hard) for each discrete-time instant.

The paper is organized as follows. In the next section the basic definitions and concepts related to hybrid automata are given. In section 3, the MPC problem and associated computational issues are stated. Similarly, in section 4, the MIPC problem is addressed. In section 5, a semi-explicit algorithm for the problem stated is derived. A CPU application of the algorithm is given in section 6. Finally, section 7 contains conclusions.

2. MODELLING AND PROBLEM FORMULATION

2.1 *Hybrid automaton*

Let $X \in \mathbb{R}^n$ be the continuous state space and let $Q = \{q_1, ..., q_N\}$ be the finite set of discrete states. The continuous state space specifies the possible values of the continuous states for each $q \in Q$ where the continuous dynamics is modelled by differential or difference equations. The general nonlinear formulation is non-trivial involving optimization of nonlinear equations. Thus, a linear discrete-time model is used as follows:

$$x(t+1) = f_q(x(t), u(t)) = A_q x(t) + B_q u(t) + c_q \tag{1}$$

where $A_q \in \mathbb{R}^{n \times n}$, $B_q \in \mathbb{R}^{n \times n_u}$, $c_q \in \mathbb{R}^n$, and $u \in U \subset \mathbb{R}^{n_u}$ is the system input: $U = \{u | C_U u \leq h_U, C_U \in \mathbb{R}^{l \times n_u}, h_U \in \mathbb{R}^l\}$. The discrete-time hybrid automaton is defined, similarly to (Pang and Spathopoulos, 2004), as follows:

Definition 1. A **linear discrete-time hybrid automaton** is a collection $A = (Q, X, f, U, \Sigma, Inv, E, G, c)$ where $Q = \{q_1, ..., q_N\}$ is a set of discrete states; $X \subseteq \mathbb{R}^n$ is the continuous state space; $f : Q \times X \times U \rightarrow 2^X$ assigns every discrete state a Lipschitz continuous evolution function which is described by the linear difference equation (1); U and Σ are the sets of continuous and discrete inputs respectively; let $\epsilon \in \Sigma$ denote the situation where no discrete command is issued; $Inv : Q \rightarrow 2^X$ assigns each $q \in Q$ an invariant set; $E \subseteq Q \times \Sigma \times Q$ is a collection of discrete transitions; $G : E \rightarrow 2^X$ assigns each $e = (q, \sigma, q') \in E$ a guard; $c : (Q \times Q) \rightarrow \mathbb{R}^+$ assigns a positive cost to each transition.

All the sets involved above are considered as polytopes. The guard set $G_{q,q'}(\sigma)$ is the subset of the state space where the system can switch from location q to q'. The moment at which the transition takes place is a design variable. An external system (controller) orders an appropriate discrete input when a certain condition, subject to design, is satisfied.

Definition 2. A hybrid controller is a map: $C : Q \times X \rightarrow 2^{\Sigma \times U}$. The controller issues both discrete inputs $C_d(q(t), x(t)) \in 2^\Sigma$ and continuous inputs $C_c(q(t), x(t)) \in 2^U$.

2.2 *Problem Statement*

Let $\Pi = \{\pi\}$ denote the set of all discrete paths from q_0 to q_F:

$$\Pi = \{\pi | \exists \sigma, \exists N \in \mathbb{N}, j = 0, ..., N-1, q_N = q_F : e_j = (q_j, \sigma, q_{j+1}) \in E \wedge \pi = (q_0, ..., q_N)\}$$

Essentially, the discrete paths are derived by abstracting the continuous dynamics away *i.e.* considering reachability on the discrete graph. Let $l(\pi)$ be the number of discrete transitions in a path $\pi \in \Pi$. Therefore, $\pi = (q_0^\pi, q_1^\pi, ..., q_{l(\pi)}^\pi)$, with $q_0^\pi = q_0$, $q_{l(\pi)}^\pi = q_F$ and the cost of path π is defined as: $c(\pi) = \sum_{i=1}^{l(\pi)} c(q_{i-1}^\pi, q_i^\pi)$. This function represents the transition cost along π from an initial state (q_0, x_0) to a final state $(q(t_f), x(t_f)) \in F$.

Given a hybrid automaton A and a target set $F = (q_F, X_F)$, for a state (q_0, x_0), the control problem defined here can be cast as follows: Design the sequence of control inputs such that all trajectories will reach the target set while minimizing associated cost functions. This is formulated in two steps:

(1) find the shortest discrete path with the minimal discrete cost $c(\pi)$;
(2) compute optimal (continuous and discrete) control inputs for each discrete state (location) on-line. Here optimality is addressed locally and therefore the overall design is suboptimal.

For the first step, we utilize a generalization of Dijkstra's shortest path algorithm on weighted graphs (Martins *et al.*, 1998), and find the shortest path with the minimum cost $c(\pi)$ from q_0 to q_F.

For the second step, depending on whether the current state belongs to a guard set or not, we have two cases. If the current state is not in the desired guard set, then the standard MPC method is employed to drive the current state to the guard set where the system may be switched to the next discrete state along the path π. On the other hand, if the current state has already reached the guard set, then the MIPC method is used to drive the current state to the next guard set along the path π. This procedure is repeated until the target set is reached without violating any constraint.

3. THE MODEL PREDICTIVE CONTROL (MPC) PROBLEM

3.1 *Constrained Optimal Control*

For the shortest path π, the aim is to compute a suboptimal controller which successfully drives the system from (q_0, x_0) into (q_F, X_F). Let $\pi = (q_0^\pi, q_1^\pi, ..., q_{l(\pi)}^\pi)$ be the path, and $G_{q_i, q_{i+1}}^\pi = \{x | C_i^\pi x \leq h_i^\pi\}$ be the transition guards from discrete state q_i^π to q_{i+1}^π, with $i = 0, 1, ..., l(\pi) - 1$, where $C_i^\pi \in \mathbb{R}^{n_c \times n}$, $h_i^\pi \in \mathbb{R}^{n_c}$. Also, let $X_F = \{x | C_F x \leq h_F\}$, with $C_F \in \mathbb{R}^{n_f \times n}$, $h_F \in \mathbb{R}^{n_f}$. Given a state $x(t) \in Inv(q_i^\pi)$, we define the following optimal control problem:

Problem 1

First define the following cost function:

$$J_i(U_t^{N-1}, x(t)) =: \omega_1 \cdot \| x(t+N|t) - T_i \|_p + \sum_{k=0}^{N-1} \omega_2 \cdot \| x(t+k|t) - T_i \|_p + \sum_{k=0}^{N-1} \omega_3 \cdot \| u(t+k|t) - u_e \|_p$$

The factors $\omega_1, \omega_2, \omega_3 \in \mathbb{R}$ are appropriate weights for the contributions of these three terms. Also, $U_t^{N-1} =: [u^T(t+0|t), u^T(t+1|t), ..., u^T(t+N-1|t)]^T$. At each time t, $x(t+k|t)$ and $u(t+k|t)$ denote the predicted state and input at time $t+k$. $\|x(t+k|t) - T_i\|_p$ describes the distance between the current state and (the nearest boundary) of the guard (target) set:

$$T_i = \begin{cases} G^{\pi}_{q_i, q_{i+1}} & if \ i \in \{0, 1, ..., l(\pi)-1\} \\ X_F & if \ i = l(\pi) \end{cases} \quad (2)$$

with a norm $\| \cdot \|_p$, $p = \infty, 2, 1$. $\|u(t+k|t) - u_e\|_p$ contains the deviation of $u(t+k|t)$ from a reference input u_e. N is the prediction horizon.

The finite-time optimal control problem is defined as:

$$\min_{U_t^{N-1}} J_i(U_t^{N-1}, x(t))$$

$$s.t. \begin{cases} x(t+k+1|t) = \\ \quad A_{q_i^{\pi}} x(t+k|t) + B_{q_i^{\pi}} u(t+k|t) + c_{q_i^{\pi}} \\ u(t+k|t) \in U \\ x(t+k|t) \in Inv(q_i^{\pi}) \end{cases}$$

The main idea of predictive control is to use the model of the plant to *predict* the future evolution of the system. Based on this prediction, at each time step t the controller selects a sequence of future command inputs through an on-line optimization procedure, which aims at minimizing the distance from the current state to the target set, and enforces fulfillment of the constraints. Only the first sample of the optimal sequence is actually applied to the plant at time t. At time $t+1$, a new sequence is evaluated to replace the previous one. This on-line "re-planning" provides the desired feedback control feature.

3.2 *Computational issues*

The objective of using MPC in a discrete state q is to drive the current state to the guard set (target set) in an optimal way (locally). The terms $\sum_{k=0}^{N-1} \|x(t+k|t) - T_i\|_p$ and $\sum_{k=0}^{N-1} \|u(t+k|t) - u_e\|_p$ are less important comparing with the $\|x(t+N|t) - T_i\|_p$. In the following, we use an objective function that only involves the position of the final state $x(t+N|t)$.

Let $T_i = \{x | C_{T_i} x \leq H_{T_i}, C_{T_i} \in \mathbb{R}^{n_t \times n}, H_{T_i} \in \mathbb{R}^{n_t}\}$. It is clear that if $C_{T_i} x(t+N|t) \leq H_{T_i}$, then the state $x(t+N|t)$ is in the set T_i. Let $E_T = C_{T_i} x(t+N|t) - H_{T_i}$, where E_j is the j-th element of the vector $E_T \in \mathbb{R}^{n_t}$. We re-formulate problem 1 as:

$$\min_{U_t^{N-1}} \max_j \quad E_j \quad (3)$$

$$s.t \begin{cases} x(t+k+1|t) = A_{q_i^{\pi}} x(t+k|t) + B_{q_i^{\pi}} u(t+k|t) + c_{q_i^{\pi}} \\ u(t+k|t) \in U \\ x(t+k|t) \in Inv(q_i^{\pi}) \end{cases}$$

Consider that:

$$x(t+N|t) = A^N x(t) + \sum_{k=0}^{N-1} A^k B u_{N-1-k} \quad (4)$$

The above min-max optimization problem can be equivalently transformed as:

$$\min_{U_t^{N-1}} \quad z$$

$$s.t. \begin{cases} C_{T_i} A^N x(t) + C_{T_i} \sum_{k=0}^{N-1} A^k B u_{N-1-k} - H_{T_i} \leq \mathbf{1_{n_t}} z \\ x(t+k+1|t) = \\ \quad A_{q_i^{\pi}} x(t+k|t) + B_{q_i^{\pi}} u(t+k|t) + c_{q_i^{\pi}} \\ u(t+k|t) \in U \\ x(t+k|t) \in Inv(q_i^{\pi}) \end{cases}$$

where $\mathbf{1_j}$ is a column vector of j length of ones, *i.e.* $\mathbf{1_j} =: [1, ..., 1]^T \in \mathbb{R}^j$. The above problem can be rewritten in the more compact form, such that by treating $x(t)$ as a vector of parameters, the LP becomes a multi-parametric LP:

$$\min_z J(z, x(t)) = z \quad (5)$$

$$s.t. \ \ Gz \leq S + Fx(t) \quad (6)$$

where G, S, F are appropriate matrices. More details about multi-parametric LP can be found in (Bemporad *et al.*, 2002).

The advantage of above formulation is that the objective cost function indicates whether the final state $x(t+N|t)$ has been driven to T_i. *i.e.* if $J(z, x(t)) = z \leq 0$ then $x(t+N|t) \in T_i$, otherwise $x(t+N|t) \notin T_i$. Here, if there exists a minimal N^*, such that $J(z, x(t)) = z \leq 0$, then N^* is the minimal number of steps from $x(t)$ to T_i and the corresponding U_t^* is the optimal control sequence. In this case, online computation is not necessary, since an optimal solution is derived off-line.

4. THE MIXED INTEGER PREDICTIVE CONTROL (MIPC) PROBLEM

Once the state $x(t)$ is driven to a guard set $G^{\pi}_{q_i, q_{i+1}}$ using MPC, it is up to the discrete controller to decide whether to let it idle in state q_i or switch to the next state q_{i+1}. To design the local optimal discrete controller, the logical decisions and the transition structure of A are expressed using relations of binary variables, and the solution is then determined by Mixed Integer Programming (MIP).

The dynamics at t are determined by the current discrete state and input. Let $|Q|$ denote the number of discrete states of A. We introduce $|Q|$ binary variables defined as

$$\lambda_i(t) = \begin{cases} 1 & if \;\; q(t) = q_i \\ 0 & otherwise \end{cases} \qquad i \in \{1, ..., |Q|\}$$

It is clear that:

$$\sum_{i=1}^{|Q|} \lambda_i(t) = 1 \tag{7}$$

Under the assumption that the guard set $G^{\pi}_{q_i,q_{i+1}}$ has no intersection with another guard set $G^{\pi}_{q_i,q_j}$, $j \neq i+1$ along the path, we have for state $x(t) \in G^{\pi}_{q_i,q_{i+1}}$ that:

$$\lambda_i(t) + \lambda_{i+1}(t) = 1 \tag{8}$$

where $\lambda_i(t)$ and $\lambda_{i+1}(t)$ are the binary variables associated with the discrete states q^{π}_i and q^{π}_{i+1} respectively.

The following optimal control problem is solved for the state $x(t) \in G^{\pi}_{q_i,q_{i+1}}$ which can be observed by the system.

Problem 2

Define the cost function

$$J_i'(U_t^{N'-1}, x(t)) =: \omega_1 \cdot \|x(t+N'|t) - T_{i+1}\|_p + \omega_2 \cdot \sum_{k=0}^{N'-1} \|x(t+k|t) - T_{i+1}\|_p + \omega_3 \cdot \sum_{k=0}^{N'-1} \|u(t+k|t) - u_e\|_p$$

where N' is the prediction horizon and consider the finite-time optimal control problem

$$\min_{U_t^{N'-1}} \; J_i'(U_t^{N'-1}, x(t)) \tag{9}$$

$$s.t \begin{cases} x(t+k+1|t) = \lambda_i(t+k|t)[A_{q_i^{\pi}} x(t+k|t) + \\ \quad B_{q_i^{\pi}} u(t+k|t) + c_{q_i^{\pi}}] + \lambda_{i+1}(t+k|t)[A_{q_{i+1}^{\pi}} \\ \quad x(t+k|t) + B_{q_{i+1}^{\pi}} u(t+k|t) + c_{q_{i+1}^{\pi}}] \\ u(t+k|t) \in U \\ x(t+k|t) \in G^{\pi}_{q_i,q_{i+1}} \\ \lambda_i(t+k|t), \lambda_{i+1}(t+k|t) \in \{0,1\} \\ \lambda_i(t+k|t) + \lambda_{i+1}(t+k|t) = 1 \\ \lambda_i(t+k|t) - \lambda_i(t+k-1|t) \leq 0 \end{cases} \tag{10}$$

It should be noted that the set T_{i+1} is different for the set T_i in problem 1 as:

$$T_{i+1} = \begin{cases} G^{\pi}_{q_{i+1},q_{i+2}} & if \;\; i \in \{0, 1, ..., l(\pi) - 2\} \\ X_F & if \;\; i = l(\pi) - 1 \end{cases}$$

The constraint $\lambda_i(t+k|t) - \lambda_i(t+k-1|t) \leq 0$ for all $k = 1, \ldots, N'$ in the last line of equation (10) guarantees that there is only one jump from q^{π}_i to q^{π}_{i+1}. For any $l = 0, ..., N'$, if $\lambda_i(t+l|t) = 0$, the system is switched to the next discrete location q^{π}_{i+1} since it is impossible to have another $l' > l$ such that $\lambda_i(t+l'|t) = 1$. □

In order to derive a linear optimization problem, the products of variables are linearized. The M-formulations (Stursberg and Panek, 2002) transform the products $x_j(t) \cdot \lambda_i(t)$ and $u_j(t) \cdot \lambda_i(t)$ for $j = 1, ...n$ in terms of linear inequalities, see also (Bemporad *et al.*, 1999).

The computational methods of MIPC are different from those of MPC. The tools of MIPC are mixed integer linear programming (MILP) or mixed integer quadratic programming (MIQP) instead of linear programming or quadratic programming for MPC.

5. A SEMI-EXPLICIT ALGORITHM

Given a discrete path $\pi = (q_0, q_1, ..., q_{l(\pi)} = q_F)$ and an initial state (q_0, x_0), the following online predictive control algorithm derives a controlled trajectory from (q_0, x_0) to the target set :

Algorithm 1. (A semi-explicit algorithm).

1. $t = 0, i = 0, x(0) = x_0$;
2. **while** $i \leq l(\pi) - 1$ **do**;
3. **if** $x(t) \in G^{\pi}_{q_i q_{i+1}}$;
4. solve problem 2;
5. **if** $\lambda_i^*(t) = 1 \wedge x(t+1) \in G^{\pi}_{q_i q_{i+1}}$;
6. $C^*(q(t), x(t)) = (\epsilon, u_t^*(0))$, $t := t+1$; go to 3
7. **else**
8. $C^*(q(t), x(t)) = (\sigma_{i,i+1}, u_t^*(0))$; $t := t+1$; $i := i+1$; go to 2
9. **end**
10. **else**
11. solve problem 1;$C^*(q(t), x(t)) = (\epsilon, u_t^*(0))$; $t := t+1$; go to 3
12. **end while**
13. **while** $x(t) \notin X_F$ **do**;
14. solve problem 1;$C^*(q(t), x(t)) = (\epsilon, u_t^*(0))$; $t := t+1$
15. **end while**

The above algorithm contains two *while* loops. The first *while* loop stops when the system reaches the last discrete state q_F of the path. The second *while* loop terminates when $x(t) \in X_F$. In the first *while*, the algorithm first checks whether the continuous state $x(t)$ is in the guard set or not. If yes, it solves the MIPC problem 2. The solution of problem 2. provides both the continuous and discrete inputs. When a discrete switching occurs, the index i is increased by one and the system evolves in the new discrete state. On the other hand, if the continuous state is outside the guard set, the algorithm solves the MPC problem 1 and calculates the continuous input which optimally drives the system to the guard (target) set.

In comparison to MLD models, the guard sets in hybrid automata bring more choices for the controller. Thus, it costs more to compute the switching sequence on-line. Algorithm 1 is called *semi-explicit* (Lazar and Heemels, 2003) since

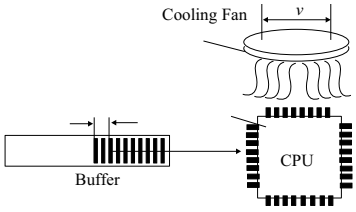

Fig. 1. The CPU model

the shortest discrete path is calculated off-line, while both continuous and discrete inputs are calculated on-line. In this semi-explicit set up the CPU computational time is reduced.

In comparison to the feedback controller of the time-optimal control problem stated in (Pang and Spathopoulos, 2004), the on-line controller of algorithm 1 achieves reduction of storing capacity, since in (Pang and Spathopoulos, 2004) each level set has to be stored. In addition, the on-line controller avoids non-determinism by supplying a sequence of optimal control inputs, instead of sets of control inputs. Thus, controlled trajectories can be simulated.

6. CPU PROCESSING CONTROL

In this section, the above results are applied on the CPU processing control problem (Azuma and Imura, 2003). In order to realize the high-speed and energy-saving computing more effectively, we model the system as a hybrid automaton and apply the semi-explicit algorithm 1 to this system. The state of system when a sufficiently long time has passed after booting the system is defined as equilibrium state of this model, and define the output of the temperature sensor equipped on the motherboard as the CPU temperature. Then from some experimental results, the dynamical behaviors of this model around equilibrium state are given as follows: (a) the time variation of the amount of CPU tasks in the buffer proportionally decreases as clock frequency increases, and (b) the time variation of CPU temperature proportionally increases as the clock frequency increases and the angular velocity of cooling fan decreases.

Thus the state equations of this model around the equilibrium state are expressed as follows:

$$\begin{cases} \dot{\pi} = -K_1 c \\ \dot{\rho} = -K_2 \rho + K_3 c - K_4 \omega \\ \dot{\omega} = -K_5 \omega + K_6 v \end{cases} \tag{11}$$

where $\pi \in \mathbb{R}$, $\rho \in \mathbb{R}$, and $\omega \in \mathbb{R}$ are the deviations of the amount of CPU tasks in the buffer, the CPU temperature and angular velocity of a cooling fan from the equilibrium state, respectively, and $c \in \mathbb{R}$ and $v \in \mathbb{R}$ are deviations of clock frequency and the voltage input of a cooling fan from equilibrium input, respectively. The first and second equations express the dynamics (a) and (b), while the third equation is the dynamics of the DC motor of fan.

Table 1. Continuous dynamics.

State	Dynamics($\dot{x} =$)	Input	Invariant
q_1	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -0.05 & -0.5 \\ 0 & 0 & -3 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0.5 \end{bmatrix} u$	$c = 0$ $v \in [-10, 10]$	$-10 \le \pi \le 3$ $-10 \le \rho \le 10$ $-10 \le \omega \le 10$
q_2	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -0.05 & -0.5 \\ 0 & 0 & -3 \end{bmatrix} x + \begin{bmatrix} -1 & 0 \\ 0.1 & 0 \\ 0 & 0.5 \end{bmatrix} u$	$c \in [-5, 5]$ $v \in [-10, 10]$	$\pi \le 10$ $\rho \le 10$ $\pi + \rho \ge 10$ $-10 \le \omega \le 10$
q_3	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & -0.05 & -0.5 \\ 0 & 0 & -3 \end{bmatrix} x + \begin{bmatrix} -1 & 0 \\ 0.1 & 0 \\ 0 & 0 \end{bmatrix} u$	$c \in [-5, 5]$ $v = 0$	$0 \le \pi \le 10$ $-10 \le \rho \le 7$ $-10 \le \omega \le 10$

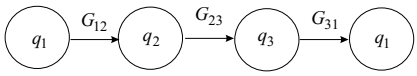

Fig. 2. The discrete path π.

Let c and v can be switched according to the values of π and ρ at the switching times. The policy is that

- the voltage v of cooling fan is the only control input in the usual mode (q_1);
- the clock frequency c is the only control if the amount of CPU tasks is large but CPU temperature is not so high that is called busy mode (q_3);
- both c and v are used as control inputs only in an emergency mode (q_2).

Let $x = [\pi, \rho, \omega]^T$ and $u = [c, v]^T$ be the continuous state and control input. The parameters in each location are shown in Table 1. The discrete path considered in this example is described in figure 2. where the guard sets are:

$$G_{12}(\sigma_{12}) = \left\{ x \,\middle|\, \begin{bmatrix} -1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x \le \begin{bmatrix} -10 \\ 3 \\ 10 \end{bmatrix} \right\}$$

$$G_{23}(\sigma_{23}) = \left\{ x \,\middle|\, \begin{bmatrix} -1 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x \le \begin{bmatrix} -10 \\ 10 \\ 7 \end{bmatrix} \right\}$$

$$G_{31}(\sigma_{31}) = \left\{ x \,\middle|\, \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix} x \le \begin{bmatrix} 3 \\ 0 \\ 7 \\ 10 \end{bmatrix} \right\}$$

The initial and target states are $(q_1, [1, 7, -10]^T)$ and $(q_1, [0, 0, 0]^T)$ and the equilibrium input is $u_e = [0, 0]^T$. Applying the algorithm 1 on the discretized automaton with $T_s = 0.5s$, we have the simulation results shown in figure 3, where the objective functions take the form of 3. The trajectory projected on $\pi - \rho$ space is shown in figure 4. The optimal input of the controller is depicted in figures 5 and 6.The execution time on a Pentium 1GHz with 256MB RAM is 3.435 s.

7. CONCLUSIONS

The main contribution of this paper is the use of hybrid automata models in association with

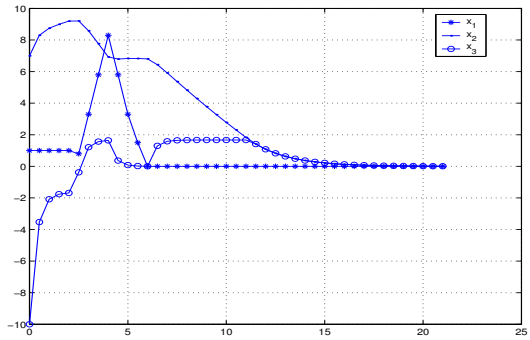

Fig. 3. Trajectories of $x_1(t), x_2(t)$ and $x_3(t)$.

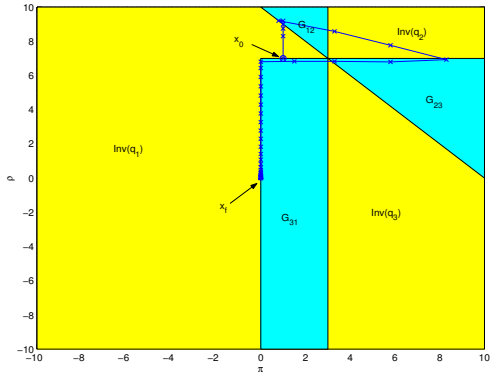

Fig. 4. The trajectory on $\pi - \rho$ space.

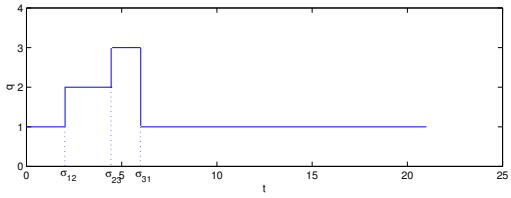

Fig. 5. Discrete states along the optimal trajectory and optimal discrete inputs.

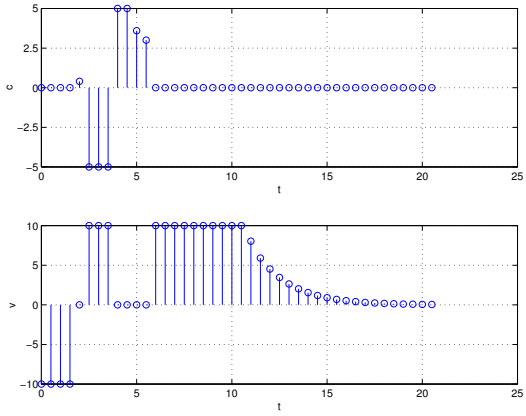

Fig. 6. Optimal continuous inputs.

predictive control techniques in order to derive suboptimal solutions for the target control problem instead of using MLD or PWA models. The difference is that the hybrid automaton model involves guard sets (switching conditions) that introduce non-determinism. Algorithm 1 reduces the on-line computation by deriving off line the shortest discrete path. In addition, the on-line controller avoids non-determinism by supplying a sequence of optimal control inputs, instead of sets of control inputs as in (Pang and Spathopoulos, 2004).

Model predictive and mixed integer predictive control of hybrid systems may be extended to systems in the face of persistent disturbances. The controller with a robust performance for the closed-loop system has to be provided.

8. REFERENCES

Azuma, S. and J. Imura (2003). Optimal control of sampled-data piecewise affine systems and its applicatio to cpu processing control. In: *Proceedings of the 42nd IEEE Conference on Decision and Control*. Maui,Hawaii USA.

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

Bemporad, A., F. Borreli and M. Morari (2000). The explicit solution of constrined lp-based receding horizon control. In: *Proceedings of the 39th IEEE Conference on Decision and Control*. Sydney, Australia.

Bemporad, A., F. Borrelli and M. Morari (2002). Model predictive control based on linear programming - the explicit solution. *IEEE Trans. on Automatic Control* **47**(12), 1974–1985.

Bemporad, A., W.P.M.H. Heemels and B.De Schutter (2001). On hybrid systems and closed-loop mpc systems. In: *Proceedings of the 40th IEEE Conference on Decision and Control*. Orlando, Florida.

Lazar, M. and W.P.M.H. Heemels (2003). A semi-explicit mpc set-up for constrained piecewise affine systems. In: *Proceedings of the European Control Conference*. Cambridge, UK.

Martins, E., M. Pascoal and J. Dos Santos (1998). The k shortest paths problem. Technical report. Department of Mathematics, Coimbra University. Portugal.

Pang, Y. and M.P. Spathopoulos (2004). Time-optimal control for discrete-time hybrid automata. *International Journal of Control (Accepted)*.

Stursberg, O. and S. Engell (2002). Optimal control of continuous systems using mixed-integer programming. In: *Proceedings of the 15th IFAC World Congress on Automatic Control*. Barcelona, Spain.

Stursberg, O. and S. Panek (2002). Control of switched hybrid systems based on disjunctive formulations. In: *Hybrid Systems: Computation and Control*. LNCS 2289. Springer-Ver lag.