ROBUST SEMI-EXPLICIT MODEL PREDICTIVE CONTROL FOR HYBRID AUTOMATA

Yan Pang *, **Hao Xia** *, **Michael P. Spathopoulos** *

* *University of Strathclyde, Glasgow, G1 1QE, U.K.*
{*yan.pang, hao.xia, michael.spathopoulos*}*@strath.ac.uk*

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 (robust) model predictive control (MPC) technique is used when the current state is not within a guard set, otherwise the (robust) 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.

1. INTRODUCTION

In this paper, a computationally efficient solution is presented for the supervisory target control of hybrid automata (Trontis and Spathopoulos, 2003). The controller design can be considered as a two stage optimization. At the first stage, for any given initial state, we solve a ***high priority*** optimization problem which minimizes the total discrete transition cost to the target set. This is cast as a reachability problem from a given initial set to a target set. With this reachability problem solved, the initial set can be partitioned into a number of disjoint subsets, and any state contained in a given subset will have the same discrete switching path (weighted shortest path) of the minimized discrete transition cost. At the second stage, for a given initial state, a hybrid controller can be derived based on the results from stage one. The design is formulated as a ***low priority*** optimization problem which minimizes a continuous transition performance index subject to the constraint that the weighted shortest path will be followed. Due to the constraints imposed on the system state and control input, a constrained optimization needs to be solved. A hybrid controller is calculated on-line using model predictive control (MPC) techniques.

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 on-line choices for the switching part, by selecting the shortest discrete path offline. It is then shown that the shortest path can be used to derive a semi-explicit algorithm for hybrid automata. The design proposed in this paper is computationally efficient due to the fact that the global optimization problem has been decomposed into several consecutive local optimization problems. However, the price to be paid for the computational saving is that the result is sub-optimal.

The rest of the paper is organized as follows. In the next section the basic definitions and concepts related to hybrid automata are given. The MPC and MIPC problems are stated and addressed in section 3, In section 4, a semi-explicit algorithm for the problem stated is derived. In section 5, we discuss the effect of bounded disturbances and how the nominal design can be applied in the presence of these disturbances. A CPU application of the algorithm is given in section 6.

2. MODELLING AND PROBLEM FORMULATION

2.1 *Hybrid automaton*

The discrete-time hybrid automaton used in this paper is defined as follows:

Definition 1. (Pang and Spathopoulos, 2005), A **linear discrete-time hybrid automaton** is a collection $A = (Q, X, f, U, D, \Sigma, Inv, E, G, c)$ where $Q = \{q_1, ..., q_N\}$ is a set of discrete states; $X \subseteq \mathbb{R}^n$ is the continuous state space; $f : Q \times X \times U \to 2^X$ assigns every discrete state a Lipschitz continuous evolution function which is described by the linear difference equation (1):

$$x(t+1) = A_q x(t) + B_q u(t) + d(t) \qquad (1)$$

The control input $u(t) \in U$ and disturbance $d(t) \in D$ that both contain the origin as an interior point. Σ is the set of discrete inputs; 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, 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. MPC AND MIPC PROBLEMS

In this section, it is assumed that there is no additive disturbance in linear discrete-time model defined in equation 1.

3.1 *The Model Predictive Control (MPC) Problem*

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:

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

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 *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:

$$\sum_{i=1}^{|Q|} \lambda_i(t) = 1 \tag{3}$$

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:

$$\lambda_i(t) + \lambda_{i+1}(t) = 1 \tag{4}$$

where $\lambda_i(t)$ and $\lambda_{i+1}(t)$ are the binary variables associated with the discrete states q_i^{π} and q_{i+1}^{π} 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

$$\min_{U_t^{N'-1}} J_i'(U_t^{N'-1}, x(t)) \tag{5}$$

$$s.t \begin{cases} x(t+k+1|t) = \lambda_i(t+k|t)[A_{q_i^{\pi}} x(t+k|t) + \\ \quad B_{q_i^{\pi}} u(t+k|t) + c_{q_i^{\pi}}] + \lambda_{i+1}(t+k|t)[A_{q_{i+1}^{\pi}} \\ \quad x(t+k|t) + B_{q_{i+1}^{\pi}} u(t+k|t) + c_{q_{i+1}^{\pi}}] \\ u(t+k|t) \in U \\ x(t+k|t) \in G^{\pi}_{q_i,q_{i+1}} \\ \lambda_i(t+k|t), \lambda_{i+1}(t+k|t) \in \{0,1\} \\ \lambda_i(t+k|t) + \lambda_{i+1}(t+k|t) = 1 \\ \lambda_i(t+k|t) - \lambda_i(t+k-1|t) \leq 0 \end{cases} \tag{6}$$

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 (6) guarantees that there is only one jump from q_i^{π} to q_{i+1}^{π}. For any $l = 0, ..., N'$, if $\lambda_i(t+l|t) = 0$, the system is switched to the next discrete location q_{i+1}^{π} since it is impossible to have another $l' > l$ such that $\lambda_i(t+l'|t) = 1$. $\square$.

The computational tools of MPC are linear programming (LP) or quadratic programing (QP), while the tools for MIPC are mixed integer linear programming (MILP) or mixed integer quadratic programming (MIQP), see (Pang *et al.*, 2005) for more details.

4. 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.

5. ROBUST MPC AND MIPC

In the previous sections, it is assumed that there is no additive disturbance in the continuous dynamics. Since MPC and MIPC are both receding horizon *state feedback* laws, the inherent robustness of deterministic MPC and MIPC applied to nominal system is usually enough if the disturbance is sufficiently small (Scokaert and Rawlings, 1995). On the other hand, it is well known that the action of a bounded disturbance can destabilize a predictive controller which is stabilized for the nominal case. A straightforward solution is to treat the disturbance explicitly and carry out a min-max optimization as proposed in (Scokaert and Mayne, 1998). However, there are two major drawbacks associated with this "worst-case" formulation. The first is that the resulting optimization procedure is computationally expensive and the second is that the optimizing performance for the "worst-case" disturbance represents an unrealistic scenario and may yield poor performance whenever the disturbance realization gets close to zero. For the above reasons, a more sensible approach is to minimize the nominal performance index while imposing constraint fulfillment for all admissible disturbances. This idea has been pursued in (Chisci *et al.*, 2001; Langson *et al.*, 2004; Mayne *et al.*, 2005). In this paper, we will adapt a similar strategy, so that all results introduced before can be used.

Feedback model predictive control in which the decision variable is a *policy* c, was advocated in (Lee and Yu.Z., 2005; Scokaert and Mayne, 1998). The *policy* is a sequence $\{\mu_0(\cdot), \mu_1(\cdot), \dots, \mu_{N-1}(\cdot)\}$ of control laws. Determination of a control policy is usually prohibitively difficult, as first introduced in (Lee and Kouvaritakis, 2000). Thus, a suboptimal control policy in which $\mu_i(x) = v_i + Kx$ will be employed here. This state feedback law transforms the decision variable from a *policy* to a sequence of control actions $\{v_0, v_1, \dots, v_{N-1}\}$. The inherent feedback via the time-invariant K reduces the spread of trajectories due to disturbance and it is often very effective.

Before introducing the main result of this section, a few set notations need to be introduced. Given two sets A, B, then $A \oplus B \triangleq \{a+b \mid a \in A, b \in B\}$ (set addition) and $A \ominus B \triangleq \{a \mid a \oplus B \subseteq A\}$ (set subtraction).

For the discrete time system defined in (1), its corresponding nominal model is defined by

$$x(t+1) = Ax(t) + Bu(t) \tag{7}$$

For the nominal model (7) and an initial condition $\bar{x}_0$, let $\bar{\mathbf{u}} \triangleq \{\bar{u}_0, \dots, \bar{u}_{N-1}\}$ be the optimal control sequence for the cost function. By applying $\bar{\mathbf{u}}$ to (7), the optimal nominal state trajectory can be obtained as $\bar{\mathbf{x}} = \{\bar{x}_0, \bar{x}_1, \dots, \bar{x}_N)$. For the perturbed plant (1), the feedback policy c is defined as:

$$\mu_i(x, \bar{x}_i, \bar{u}_i) \triangleq \bar{u}_i + K(x - \bar{x}_i) = (\bar{u}_i - K\bar{x}_i) + Kx, \quad i = 0, 1, \dots, N-1 \tag{8}$$

Suppose the sequences $\{x_i\}$ and $\{u_i\}$ are the solutions of the perturbed system (1) with feedback policy c, *i.e.* $\{x_i\}$ and $\{u_i\}$ satisfy:

$$\begin{aligned} x_{i+1} &= Ax_i + Bu_i + d_i \\ u_i &= \bar{u}_i + K(x(i) - \bar{x}(i)) \end{aligned}$$

with initial condition $x_0 = \bar{x}_0$. A simple inductive argument yields

$$x_i \in \bar{x}_i + \sum_{j=0}^{i-1} A_K^j D \qquad i = 1, \dots, N \tag{9}$$

$$u_i \in \bar{u}_i + K\sum_{j=0}^{i-1} A_K^j D \qquad i = 1, \dots, N-1 \tag{10}$$

with $A_K = A + BK$ and $\sum$ denoting set addition. Clearly, the feasibility of the feedback policy c depends on whether x_i and u_i satisfy the original state and control constraint respectively. Based on equations (9) and (10), with initial state $x_0 \in X$, in order to guarantee the feasibility of c, $(\bar{x}, \bar{u})$ has to satisfy tighter constraints:

$$\bar{x}_i \in Inv^*(q) \ominus \sum_{j=0}^{i-1} A_K^j D, i = 1, \dots, N \tag{11}$$

$$\bar{u}_0 \in U,\ \bar{u}_i \in U \ominus K\sum_{j=0}^{i-1} A_K^j D, i = 1, \dots, N-1 \tag{12}$$

From the above two equations, it is clear that the state feedback gain K should be chosen such that $\sum_{j=0}^{i-1} A_K^j D,\ \ K\sum_{j=0}^{i-1} A_K^j D,\ \ i = 1, \dots, N-1$ are minimized. A computational technique for the minimum over approximation of these sets has been presented in (Rakovic *et al.*, 2005).

By replacing the original state and control constraints with tighter constraints defined in (11) and (12), it is straightforward to re-formulate the MPC and MIPC problems defined above.

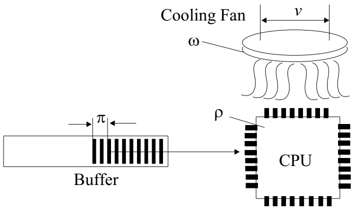

Fig. 1. The CPU model

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:

$$\begin{cases} \dot{\pi} = -K_1 c \\ \dot{\rho} = -K_2\rho + K_3 c - K_4\omega \\ \dot{\omega} = -K_5\omega + K_6 v \end{cases} \tag{13}$$

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.

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

Table 1. Continuous dynamics.

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

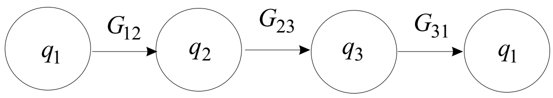

Fig. 2. The discrete path π.

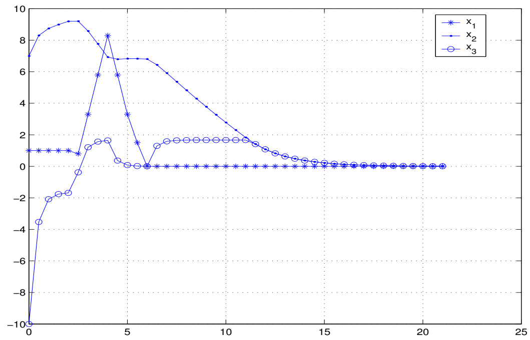

Fig. 3. Trajectories of $x_1(t), x_2(t)$ and $x_3(t)$.

each location are shown in Table 1. The discrete path considered in this example is described in figure 2. where the guard sets are:

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

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

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

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. 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 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

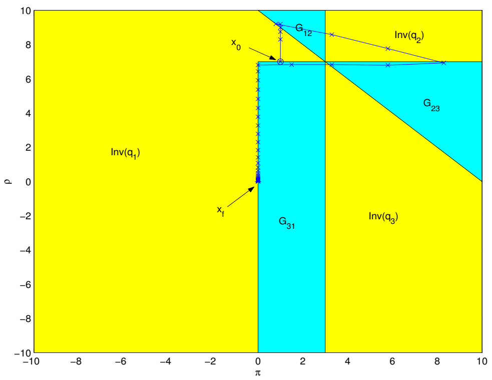

Fig. 4. The trajectory on $\pi - \rho$ space.

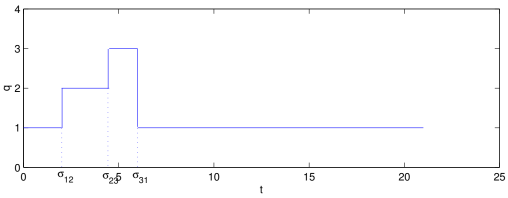

Fig. 5. Discrete states along the optimal trajectory and optimal discrete inputs.

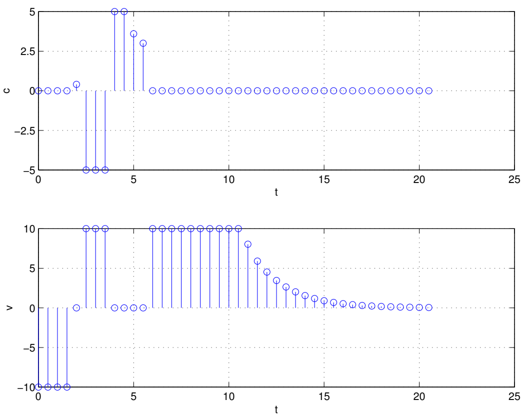

Fig. 6. Optimal continuous inputs.

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, 2005). MPC and MIPC of hybrid systems has been extended to systems in the face of persistent disturbances. This is achieved by imposing tighter state and control constraints to the nominal system. Then, the feedback predictive controller, based on the nominal state and control trajectories, provides a suboptimal solution to the target control problem.

8. REFERENCES

Azuma, S. and J. Imura (2003). Optimal control of sampled-data piecewise affine systems and its application 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.

Chisci, L., J.A. Rossiter and G. Zappa (2001). Systems with persistent disturbances: predictive control with restricted constraints. *Automatica* **37**, 1019–1028.

Langson, W., I. Chryssochoos, S.V. Rakovic and D.Q. Mayne (2004). Robust model predictive control using tubes. *Automatica* **40**, 125–133.

Lee, J.H. and Yu.Z. (2005). Worst-case formulations of model predictive control for systems with bounded parameter. *Automatica* **41**, 125–133.

Lee, Y.I. and B. Kouvaritakis (2000). Robust receding horizon control for systems with uncertain dynamics and input saturation. *Automatica* **36**, 1497–1504.

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

Mayne, D.Q., M.M. Seron and S.V. Rakovic (2005). Robust model predictive control of constrained linear system with bounded disturbances. *Automatica* **41**, 125–133.

Pang, Y. and M.P. Spathopoulos (2005). Time-optimal control for discrete-time hybrid automata. *International Journal of Control* **78**(11), 847–863.

Pang, Y., M. P. Spathopoulos and J. Raisch (2005). On suboptimal control design for hybrid automata using predictive control techniques. In: *Proceedings of the 16th IFAC World Congress on Automatic Control*. Prague.

Rakovic, S.V., E.C. Kerrigan, K.I. Kouramas and D.Q. Mayne (2005). Invariant approximations of the minimal robust positively invariant set. *IEEE Trans. on Automatic Control* **50**(3), 406–410.

Scokaert, P.O.M. and D.Q. Mayne (1998). Min-max feedback model predictive control for constrained linear systems. *IEEE Trans. on Automatic Control* **43**, 1136–1142.

Scokaert, P.O.M. and J.B. Rawlings (1995). Stability of model predictive control under perturbantions. In: *Proceedings of IFAC symposium on nonlinear control systetms design*. Lake Taboe, CA.

Trontis, A. and M.P. Spathopoulos (2003). Supervisory target control for hybrid systems. *International Journal of Control* **76**(11), 1142–1158.