Title	Approximate Algorithm for Hybrid Model Predictive Control with Time-Varying Reference
Author(s)	Kobayashi, Koichi; Hiraishi, Kunihiko; Tang, Nguyen Van
Citation	IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E92-A(8): 2046-2052
Issue Date	2009-08-01
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/9173
Rights	Copyright (C)2009 IEICE. Koichi Kobayashi, Kunihiko Hiraishi, and Nguyen Van Tang, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E92-A(8), 2009, 2046-2052. http://www.ieice.org/jpn/trans_online/
Description	

IEICE TRANS. FUNDAMENTALS, VOL.E92–A, NO.8 AUGUST 2009

PAPER

Approximate Algorithm for Hybrid Model Predictive Control with Time-Varying Reference

Koichi KOBAYASHI[†a)], **Kunihiko HIRAISHI**[†], ***Members, and*** **Nguyen Van TANG**[†], ***Nonmember***

SUMMARY In this paper, we propose a new approximate algorithm for the model predictive control (MPC) problem with a time-varying reference of hybrid systems. The proposed algorithm consists of an offline computation and an online computation. In the offline computation, candidates of mode sequences are derived. In the online computation, after the mode sequence is uniquely decided among candidates, the finite-time optimal control problem, i.e., the quadratic programming problem, is solved. So by applying the proposed algorithm, the computational amount of the online computation is decreased. First, the MPC problem with a time-varying reference is formulated. Next, the proposed algorithm is explained, and the accuracy of the obtained approximate solution is discussed. Finally, the effectiveness of the proposed method is shown by a numerical example.

key words: *offline/online computations, time-varying reference, approximate algorithm, model predictive control, hybrid systems*

1. Introduction

The model predictive control (MPC) problem of hybrid systems is reduced to a mixed integer quadratic programming (MIQP) problem, which is a kind of combinatorial optimization problems. For solving the MPC problem of hybrid systems, there are two approaches: online approach and offline approach. In the online approach, the MIQP problem is solved at each time step. In the offline approach, the MPC problem is rewritten as a multi-parametric MIQP (mp-MIQP) problem. In both approaches, the computation time to solve the problem is too long for practical applications. So it is one of the significant works to decrease the computation time to solve the MPC problem of hybrid systems. Then it will be desirable to use both online and offline computations from the computational viewpoint. The algorithms using both online and offline computations have been proposed in [8], [9], but to our knowledge, there are only few results.

On the other hand, in order to overcome the above technical difficulty, it will be important to develop an approximate algorithm of the MPC problem of hybrid systems. Some approximate algorithms for the MPC problem have been proposed in [1], [2], [5]. However, these methods are based on only the online computation or the offline computation.

In this paper, a new approximate algorithm with guaranteed accuracy for the MPC problem of hybrid systems is proposed. The proposed algorithm consists of the offline computation and the online computation. In the offline computation, candidates of mode sequences such that the MPC problem is feasible are derived, and some parameters are computed to derive the upper bound of the optimal value of a given cost function. In the online computation, after a mode sequence is decided by simple calculations using candidates of mode sequences and some parameters derived by the offline computation, the control input is obtained by solving the quadratic programming (QP) problem. So the computational amount of the online computation is similar to that of the QP problem, and the accuracy of the obtained solution can be guaranteed in the sense that the worst value of the optimal cost is given. Furthermore, in this paper, the MPC problem with a time-varying reference (offset) is considered. Since such an MPC problem appears in many practical applications, the problem formulation in this paper is practical. Thus the proposed method is effective from both computational and practical viewpoints.

This paper is organized as follows. In Sect. 2, the finite-time optimal control problem and the MPC problem are formulated. In Sect. 3, as the proposed algorithm, the offline computation algorithm and the online computation algorithm are explained respectively. Also the accuracy of approximate solutions is discussed. In Sect. 4, we illustrate the proposed algorithm by using a numerical example. In Sect. 5, we conclude this paper.

Notation: Let $\mathbf{R}$ denote the set of real numbers, respectively. Let I_n express the $n \times n$ identity matrix. For a given matrix M, let M^T denote the transpose matrix of M.

2. Problem Formulation

Consider the following discrete-time piecewise affine (DT-PWA) system

$$\begin{cases} x(k+1) = A_{I(k)}x(k) + B_{I(k)}u(k) + a_{I(k)}, \\ I(k+1) = I_+ \ \text{ if } x(k+1) \in \mathcal{S}_{I_+} \end{cases} \tag{1}$$

where $x(k) \in \mathcal{X} \subset \mathbf{R}^n$ and $u(k) \in \mathcal{U} \subset \mathbf{R}^m$ are the state and the input, respectively. $\mathcal{X}$ and $\mathcal{U}$ are given as closed and bounded convex sets. $I(k) \in \mathcal{M} := \{1, 2, \ldots, M\}$ is the mode of system, and suppose that mode transition constraints are given by a finite automaton (directed graph). In addition, $I_+ \in \mathcal{M}$ is the mode corresponding to the region, which includes $x(k+1)$. Furthermore, in order to guarantee the well-posedness of the DT-PWA system (1), we assume that $\mathcal{S}_I$ is the bounded convex polyhedron satisfying $\bigcup_{I \in \mathcal{M}} \mathcal{S}_I =$

Manuscript received January 29, 2009.
Manuscript revised March 24, 2009.

[†]The authors are with the Japan Advanced Institute of Science and Technology, Nomi-shi, 923-1292 Japan.
a) E-mail: k-kobaya@jaist.ac.jp

DOI: 10.1587/transfun.E92.A.2046

$\mathcal{X}$ and $\mathcal{S}_I \bigcap \mathcal{S}_J = \emptyset$ for all $I \neq J \in \mathcal{M}$.

For this DT-PWA system, the following finite-time optimal control problem with a time-varying reference, which is a kind of tracking problems, is considered.

Problem 1: Suppose that the DT-PWA system (1), the current time t, the current state $x(t) = x_t \in \mathbf{R}^n$ and the current mode $I(t) = I_t \in \mathcal{M}$ are given. Then for the DT-PWA system (1), find $u(k)$, $k = t, t+1, \ldots, t+N-1$ minimizing the cost function

$$J(x_t, u, x_d) = \sum_{i=t}^{t+N-1} \left\{ \bar{x}^T(i) Q \bar{x}(i) + u^T(i) R u(i) \right\} + \bar{x}^T(t+N) Q_f \bar{x}(t+N) \quad (2)$$

under the state constraints

$$x(L) \in \mathcal{S}_{I_d^r}, \quad x(t+N) \in \mathcal{S}_{I_d^{r+1}} \quad (3)$$

where $Q \geq 0$, $R > 0$, $Q_f \geq 0$, $\bar{x}(i) := x(i) - x_d(i)$, and $x_d(i)$ is the reference (offset) vector given by

$$\begin{aligned} x_d(k) &= x_d^r \in \mathcal{S}_{I_d^r}, \quad k = t, t+1, \ldots, L, \\ x_d(k) &= x_d^{r+1} \in \mathcal{S}_{I_d^{r+1}}, \quad k = L+1, L+2, \ldots, t+N. \end{aligned}$$

$x_d^r, x_d^{r+1} \in \mathbf{R}^n$ are given as constant vectors, and $I_d^r, I_d^{r+1} \in \mathcal{M}$ are the modes corresponding to the regions, which include x_d^r, x_d^{r+1}, respectively.

In the standard formulation, the reference vector $x_d(k)$, $k = t, t+1, \ldots, t+N$ is some constant vector. On the other hand, in this paper, we consider the case that the reference vector is switched between x_d^r and x_d^{r+1}. This case is called here a "switching case." This formulation is frequently appeared in practical plants such as steel plants [6] and chemical plants [10]. Note here that the state constraints (3) are a kind of mode transition constraints, and are not the fixed point constraints. Furthermore, for simplicity of discussion, the following assumption is made:

Assumption 1: The switching number of the reference vector is given by 0 or 1.

In the case that there is no switching of the reference vector, $x_d^r = x_d^{r+1}$ holds, and the switching time L is given by a suitable positive integer in the interval $[1, N]$. This case is called a "non-switching case."

Next, consider the model predictive control (MPC) problem of the DT-PWA system (1). In the online optimization of the MPC problem, Problem 1 must be solved repeatedly at each time step. Then suppose that the reference vector is given by

$$\begin{cases} x_d(k) = x_d^1, & k = t, t+1, \ldots, L_1, \\ x_d(k) = x_d^2, & k = L_1+1, L_1+1, \ldots, L_2, \\ \quad \vdots & \\ x_d(k) = x_d^s, & k = L_{s-1}+1, L_{s-1}+1, \ldots, L_s \end{cases} \quad (4)$$

where L_i is a positive integer satisfying Assumption 1, and

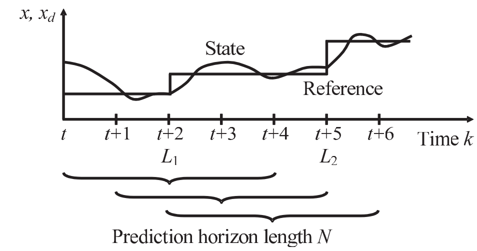

Fig. 1 Illustration of MPC with a time-varying reference.

s is a suitably large positive integer. So x_d^r, x_d^{r+1} of Problem 1 at each time step are given from (4). We remark that depending on L_i and N, it is determined whether Problem 1 at each time step is the switching case or not, and the reference vector is given in offline.

Next, we explain the problem formulation by using a simple example of Fig. 1, where the prediction horizon length is given as $N = 4$. For example, suppose that the current time is t. Then, in the time interval between t and $t+4$, the reference vector is switched only once, and the controlled state trajectory is close to the reference vector. In the MPC problem, after the current time is shifted from t to $t+1$, and the reference vector is updated, Problem 1 is solved again. Thus by using the above problem formulation, we can realize the MPC law such that the state trajectory is close to the time-varying reference vector.

In the standard methods, there are two approaches to solve the MPC problem: online optimization and offline optimization. In the online optimization, Problem 1 is rewritten as a mixed integer quadratic programming (MIQP) problem, and the MIQP problem is solved at each time step. Note here that since the current state and the reference vector are different at each time step, the MIQP problem at each time step is different. However, the MIQP problem has serious weakness, i.e., the computation time to solve the MIQP problem is too long for practical applications. On the other hand, in the offline optimization, Problem 1 is rewritten as a multi-parametric MIQP (mp-MIQP) problem. In this approach, it is not necessary to solve some optimization problem at each time step. However, for the reference vector of Problem 1 at each time step, the mp-MIQP problem must be solved, and this computation is costly. Thus these standard methods have many drawbacks. In this paper, in order to overcome these difficulties, we give up to exactly solve Problem 1, and we propose a new approximate algorithm consisting of the online computation and the offline computation. By using the proposed algorithm, reducing the computation cost of both online and offline computations is achieved.

3. Proposed Algorithm

In this section, a new approximate algorithm to solve the MPC problem is proposed. First, the offline computation algorithm is explained. Next, by using the result of the offline computation algorithm, the accuracy of approximate solutions is discussed. Finally, based on these results, the online computation algorithm is explained.

3.1 Offline Computation Algorithm

In the offline computation algorithm, mode sequences such that Problem 1 in the non-switching cases is feasible are enumerated, and for evaluating the difference in the cost function between the non-switching cases and the switching cases, some parameters are derived.

Consider the following cost function of the non-switching case

$$J_j(x_t, u) = \sum_{i=t}^{t+N-1} \left\{ \bar{x}_j^T(i) Q \bar{x}_j(i) + u^T(i) R u(i) \right\} + \bar{x}_j^T(N) Q_f \bar{x}_j(N), \quad j = 1, 2, \ldots, s \tag{5}$$

where $\bar{x}_j(i) := x(i) - x_d^j$. For all combinations of all mode sequences and j, we compute the upper bound of (5), i.e.,

$$\overline{J}_j^{I_t}(i) = \max_{x_t \in \mathcal{S}_{I_t}} \min_{u} J_j(x_t, u) \ \text{ s.t. } (1), \tag{6}$$
$$i = 1, 2, \ldots, \theta_j^{I_t}, \quad j = 1, 2, \ldots, s,$$
$$I_t = 1, 2, \ldots, M$$

where I_t is the current mode of each mode sequence, and $\theta_j^{I_t}$ is the number of mode sequences such that (6) is feasible for the current mode I_t. Note here that the current state x_t is not given, and is a continuous variable selected among $\mathcal{S}_{I_t}$. Hereafter, for given j and I_t, let $\Theta_j^{I_t}(i)$ express mode sequences such that (6) is feasible. We remark that $\theta_j^{I_t}$ is in general obtained as a large value. Then by imposing further constraints such as mode transition constraints and temporal logic constraints, the value of $\theta_j^{I_t}$ is reduced. See [4] for further details.

Next, consider Problem 1 in the switching case. For given $j = r$, the difference between J of (2) and J_r of (5) is obtained as

$$\begin{aligned} J - J_r &= \sum_{i=L+1}^{t+N-1} \left\{ \bar{x}_{r+1}^T(i) Q \bar{x}_{r+1}(i) - \bar{x}_r^T(i) Q \bar{x}_r(i) \right\} \\ &\quad + \left\{ \bar{x}_{r+1}^T(t+N) Q_f \bar{x}_{r+1}(t+N) \right. \\ &\qquad \left. - \bar{x}_r^T(t+N) Q_f \bar{x}_r(t+N) \right\} \\ &= \sum_{i=L+1}^{t+N-1} \left\{ -2(x_d^{r+1} - x_d^r)^T Q x(i) \right\} \\ &\quad -2(x_d^{r+1} - x_d^r)^T Q_f x(t+N) + \eta \end{aligned} \tag{7}$$

where $x_d^r \neq x_d^{r+1}$ and

$$\eta = (t + N - L) \left\{ (x_d^{r+1})^T Q x_d^{r+1} - (x_d^r)^T Q x_d^r \right\} + \left\{ (x_d^{r+1})^T Q_f x_d^{r+1} - (x_d^r)^T Q_f x_d^r \right\}. \tag{8}$$

Since from (7), $J - J_r$ is expressed as a linear function with respect to $x(i)$, $i = L, L+1, \ldots, t+N$, it is easy to evaluate $J - J_r$ approximately. Noting that η is a constant, we can compute the upper bound of $-2(x_d^{r+1} - x_d^r)^T Q x(i)$, $i = t, t+1, \ldots, t+N-1$ and $-2(x_d^{r+1} - x_d^r)^T Q_f x(t+N)$, i.e.,

$$\overline{\alpha}_I^l = \max_{x \in \mathcal{S}_I} \left\{ -2(x_d^{l+1} - x_d^l)^T Q x \right\}, \tag{9}$$
$$\overline{\beta}_I^l = \max_{x \in \mathcal{S}_I} \left\{ -2(x_d^{l+1} - x_d^l)^T Q_f x \right\}, \tag{10}$$
$$l = 1, 2, \ldots, s-1, \quad I = 1, 2, \ldots, M.$$

Finally, the procedure of the proposed offline computation algorithm is shown as follows.

Procedure of offline computation algorithm:

Step 1: Compute (6) for all combination of all mode sequences and j. Then mode sequences $\Theta_j^{I_t}(i)$ and corresponding upper bounds $\overline{J}_j^{I_t}(i)$, $i = 1, 2, \ldots, \theta_j^{I_t}$ are obtained.

Step 2: Compute $\overline{\alpha}_I^l$ of (9) and $\overline{\beta}_I^l$ of (10).

Remark 1: Feasibility of the optimal control problem using one of mode sequences $\Theta_j^{I_t}(i)$ is in general guaranteed in only a subregion of $\mathcal{S}_{I_t}$. Then by suitably decomposing $\mathcal{S}_{I_t}$ to some regions, i.e., adding new modes, feasibility is guaranteed in the region corresponding to each mode. To decompose $\mathcal{S}_{I_t}$, the bisimulation technique [4] can be used. In this paper, for simplicity of discussion, we assume that there is no region of the state such that feasible mode sequences do not exist.

3.2 Accuracy of Approximate Solutions

In the MPC problem, Problem 1 is solved at each time step. By selecting a mode sequence satisfying (3) among $\Theta_j^{I_t}(i)$, we obtain the approximate solution. Then Problem 1 is rewritten as a quadratic programming (QP) problem, and can be easily solved. If there exist multiple mode sequences satisfying (3), then we must select one mode sequence. In this subsection, by using the computation result of the offline computation algorithm, the accuracy of approximate solutions is discussed, and the upper bound of the optimal value of Problem 1 is derived. The derived upper bound is used in the online computation algorithm. Here, let J_{opt} denote the optimal value of the cost function (2) of Problem 1. Also let J_{app} denote the suboptimal value of the cost function (2) using one mode sequence satisfying (3). In other words, J_{app} is the approximate value of J_{opt}.

3.2.1 Non-switching Case

First, we consider the non-switching cases of the reference vector. Then we derive the following theorem.

Theorem 1: Assume that there exists no switching of the reference vector in Problem 1. Then the following relation holds:

$$J_{\text{opt}} \le J_{\text{app}} \le \overline{J}_r^{I_t}. \tag{11}$$

Proof: It is clear that $J_{\text{opt}} \le J_{\text{app}}$ holds. Since $\overline{J}_r^{I_t}$ is the upper bound of the cost function in Problem 1, (11) is derived straightforwardly. □

From Theorem 1, we see that in the non-switching case, the accuracy of approximate solutions is guaranteed in the sense of (11).

3.2.2 Switching Case

Next, we consider the switching cases of a reference vector. Although the switching cases are more complicated than the non-switching cases, we can prove the following theorem by using some parameters derived by the proposed offline computation algorithm.

Theorem 2: Assume that there exists a switching of the reference vector in Problem 1. Then the following relation holds:

$$J_{\text{opt}} \le J_{\text{app}} \le \overline{J}, \tag{12}$$

$$\overline{J} := \overline{J}_r^{I_t} + \sum_{i=L}^{t+N-1} \overline{\alpha}^r_{I^*(i)} + \overline{\beta}^r_{I^*(t+N)} + \eta \tag{13}$$

where $I^*(k)$ is the mode at time k in a mode sequence, which satisfies (3) and is selected among $\Theta_j^{I_t}(i)$.

Proof: First, the upper bound of $J - J_r$ of (7) is derived. From (9), the upper bound of the first term of the right side in (7) is obtained as

$$\sum_{i=L}^{t+N-1} \left\{-2(x_d^{r+1} - x_d^r)^T Q x(i)\right\} \le \sum_{i=L}^{t+N-1} \overline{\alpha}^r_{I^*(i)}. \tag{14}$$

Furthermore, the upper bound of the second term of the right side in (7) is also obtained as

$$-2(x_d^{r+1} - x_d^r)^T Q_f x(t+N) \le \overline{\beta}^r_{I^*(t+N)}. \tag{15}$$

From (14) and (15), the upper bound of $J - J_r$ is given as

$$J - J_r \le \sum_{i=L}^{t+N-1} \overline{\alpha}^r_{I^*(i)} + \overline{\beta}^r_{I^*(t+N)} + \eta. \tag{16}$$

Next, consider the upper bound of J_{opt}. Since the upper bound of J_r is given by (11), we obtain (12). This completes the proof. □

From Theorem 2, we see that in the switching case, the accuracy of approximate solutions is guaranteed in the sense of (12).

3.3 Onine Computation Algorithm

In this subsection, we explain the online computation algorithm. In the proposed algorithm, after a mode sequence is uniquely decided, the control input is derived by solving the QP problem. Furthermore, this procedure is repeated at each time step. First, the proposed online computation algorithm for the MPC problem of the DT-PWA system is given as follows.

Procedure of online computation algorithm:

Step 1: Suppose that the initial state x_0 and the initial mode I_0 are given. Set $t = 0$, $x(t) = x_0$ and $I(t) = I_0$.

Step 2: From the current time t and the switching time L_i of (4), check whether Problem 1 at time t is the switching case or the non-switching case. If Problem 1 is the switching case, then go to Step 3. If Problem 1 is the non-switching case, then go to Step 5.

Step 3: From (4), decide the switching time L and the reference vectors x_d^r, x_d^{r+1} of Problem 1, and go to Step 4.

Step 4: Decide a mode sequence by using the following procedure:

Step 4-1: Select the mode sequences satisfying $x_d^r \in \mathcal{S}_{I(t+L)}$ and $x_d^{r+1} \in \mathcal{S}_{I(t+N)}$ among $\Theta_r^{I(t)}(i)$, $i = 1, 2, \ldots, \theta_r^{I(t)}$.
Step 4-2: Compute $\overline{J}$ of (13) for each mode sequence selected by Step 4-1.
Step 4-3: Select a mode sequence such that $\overline{J}$ is minimum, and go to Step 6.

Step 5: Decide the reference vector $x_d^r = x_d^{r+1}$ from (4), and select a mode sequence such that $\overline{J}_r^{I(t)}$ is minimum among $\Theta_r^{I(t)}(i)$. Furthermore, go to Step 6.

Step 6: Solve Problem 1, i.e., the QP problem using the mode sequence obtained by Step 4 or Step 5, the current state $x(t)$ and the current mode $I(t)$. Thus the suboptimal control input sequence $u(k)$, $k = t, t+1, \ldots, t+N-1$ is obtained.

Step 7: Apply $u(t)$ to the plant.

Step 8: Set $t = t + 1$, and update $x(t)$ and $I(t)$. Finally, go to Step 2.

In Step 4 or Step 5, if $\theta_r^{I(t)} = 0$ holds, then the MPC problem is infeasible. Also in Step 4-1, if there exists no mode sequence satisfying $x_d^r \in \mathcal{S}_{I(t+L)}$ and $x_d^{r+1} \in \mathcal{S}_{I(t+N)}$, then the MPC problem is infeasible. These conditions can be checked in the offline computation, because the reference vector is given in offline.

Furthermore, in the non-switching cases, based on Theorem 1, we select a mode sequence by using $\overline{J}_r^{I(t)}$. In the switching cases, based on Theorem 2, we select a mode sequence by using $\overline{J}$. Thus, from Step 6, we obtain an approximate solution that guarantees the accuracy in the sense of Theorem 1 or Theorem 2.

In addition, note here that the computation time of Step 4 is very small comparing to that of the QP problem in Step

6, because Step 4 consists of simple calculations. So the computation time of the proposed online computation algorithm at each time step is similar to that of the QP problem. Furthermore, though it is difficult to estimate the computation time of the MIQP problem, the computation time of the QP problem can be estimated. Thus the proposed algorithm is effective from the computational viewpoint.

Remark 2: In the proposed algorithm, a mode sequence is decided using the upper bound of the cost function. On the other hand, we can also consider to use the lower bound of the cost function. In this paper, in order to guarantee the worst case, the upper bound is used.

Remark 3: As for the QP problem, many algorithms have been proposed. So we may select a suitable algorithm depending on the computer environment. For example, from the viewpoint that the computation time is decreased, the ILOG CPLEX solver [11] is one of the powerful solvers. On the other hand, from the viewpoint that the computation time can be estimated, the algorithm proposed in [7] is conventional.

4. Numerical Example

As a numerical example, consider the 2nd-order and 6-mode DT-PWA system where

$$A_1 = \begin{bmatrix} 1.0254 & 0.8109 \\ 0.2169 & 1.6021 \end{bmatrix}, \ A_2 = \begin{bmatrix} 0.1239 & 2.7306 \\ 0.0998 & 0.8010 \end{bmatrix},$$
$$A_3 = \begin{bmatrix} 0.6462 & 0.9246 \\ -0.4168 & 1.6046 \end{bmatrix}, \ A_4 = \begin{bmatrix} 1.0105 & 0.4241 \\ 0.1083 & 0.7903 \end{bmatrix},$$
$$A_5 = \begin{bmatrix} -1.0502 & 0.4840 \\ 0.1854 & 1.3514 \end{bmatrix}, \ A_6 = \begin{bmatrix} 0.8487 & 0.2641 \\ 0.1084 & 1.5251 \end{bmatrix}$$

and

$$B_I = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ a_I = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ I = 1, 2, \ldots, 6.$$

The partition of the state space and the mode transition constraints are given by Fig. 2 and Fig. 3, respectively. Also, the input constraint is given by $-1 \leq u(k) \leq +1$. For this DT-PWA system, let us consider the finite-time optimal control problem (Problem 1). The initial state, the reference vectors, the switching time and the prediction horizon are given by

$$x_t = \begin{bmatrix} -19.7 \\ +5 \end{bmatrix}, \ x_d^r = \begin{bmatrix} +15 \\ +5 \end{bmatrix}, \ x_d^{r+1} = \begin{bmatrix} -15 \\ -5 \end{bmatrix}$$

and $L = 5$, $N = 10$, respectively. The weighting matrices are given by $Q = Q_f = 1000I_2$ and $R = 1$.

First, the proposed offline computation algorithm is explained. From Step 1, we obtain mode sequences $\Theta_r^1(i)$, $i = 1, 2, \ldots, 50$ of Table A· 1 (see Appendix). In Step 2, we obtain

$$\overline{\alpha}_1^r = -0.2 \times 10^6, \ \overline{\alpha}_2^r = +1.0 \times 10^6,$$
$$\overline{\alpha}_3^r = +1.6 \times 10^6, \ \overline{\alpha}_4^r = -0.6 \times 10^6,$$
$$\overline{\alpha}_5^r = +0.6 \times 10^6, \ \overline{\alpha}_6^r = +1.2 \times 10^6.$$

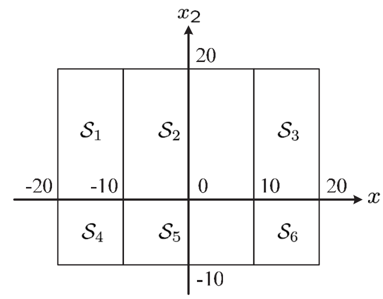

Fig. 2 Partition of the state space.

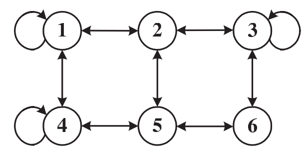

Fig. 3 Mode transition constraints expressed by the directed graph.

From $Q = Q_f = 1000I_2$, $\overline{\alpha}_I^r = \overline{\beta}_I^r$ holds.

Next, Step 4 of the proposed online computation algorithm is explained. In Step 4-1, noting that $t = 0$, $L = 5$, $N = 10$, $x_d^r \in \mathcal{S}_3$, $x_d^{r+1} \in \mathcal{S}_4$ hold, mode sequences $i = 11, 16, 21, 24, 31$ are selected among mode sequences of Table A· 1. In Step 4-2, $\overline{J}$ is computed for each mode sequence selected by Step 4-1. Then we obtain

$$i = 11 : \ \overline{J} = 1.13 \times 10^7, \ \ i = 16 : \ \overline{J} = 1.02 \times 10^7,$$
$$i = 21 : \ \overline{J} = 1.13 \times 10^7, \ \ i = 24 : \ \overline{J} = 1.36 \times 10^7,$$
$$i = 31 : \ \overline{J} = 1.04 \times 10^7.$$

Finally, in Step 4-3, mode sequence $i = 16$ is selected. Thus we obtain one mode sequence among mode sequences of Table A· 1. By solving the MIQP problem, we see that mode sequence $i = 16$ is the optimal mode sequence. Note here that a mode sequence derived by the proposed algorithm is not optimal in general.

The controlled state trajectories are shown in Fig. 4. In Fig. 4, for simplicity, the finite-time optimal control problem (Problem 1) is solved only once. We see that the obtained state trajectory satisfies $x(5) \in \mathcal{S}_3$ and $x(10) \in \mathcal{S}_4$.

Finally, we comment about the computation time to solve the proposed computation algorithms. First, the computation time to derive Table A· 1 was 6.66 [sec], where we used KCLP-HS [3] on the computer with the Intel Pentium M 1.60 GHz processor and the 756 MB memory. Next, the computation time of the online computation algorithm was 0.01 [sec], where we used MATLAB and ILOG CPLEX 11.0 [11] on the computer with the Intel Core 2 Duo 3.0 GHz processor and the 4 GB memory. In such a simple example, the difference in the computation time between the MIQP problem and the QP problem will be small. For reference, the computation time to solve the MIQP problem in this example was 0.22 [sec], where we also used CPLEX. However, we stress that it is hard to estimate the computation time of the MIQP problem. Meanwhile, the computation time of the proposed online computation algorithm can be

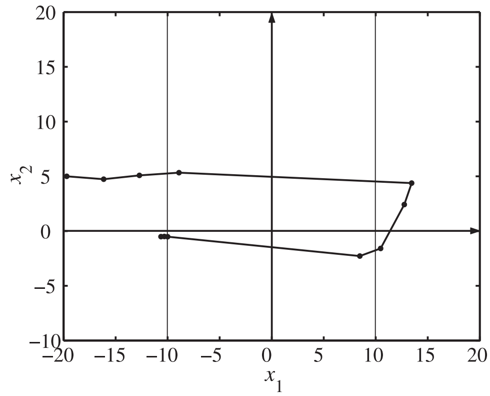

Fig. 4 Controlled state trajectories.

estimated. Furthermore, for large-scale systems, the proposed method will be more effective than the standard approach.

5. Conclusion

In this paper, we have proposed a new approximate algorithm for model predictive control with a time-varying reference of hybrid systems. The proposed algorithm consists of two phases: offline computation algorithm and online computation algorithm. In particular, since the computation time of the online computation algorithm is similar to that of the QP problem, the proposed algorithm will be useful in the practical applications. On the other hand, the evaluation of the accuracy of the approximate solution will be rough. However, for the MPC problem with state/input constraints and mode transition constraints, the proposed method will be effective. This is because candidates of mode sequences are limited.

Finally, in the proposed framework, it will be important to guarantee the stability. In addition, it will be one of future topics to consider the case that the reference vector is generated in online. Then it will be important to clarify the relation between feasibility, the reference vector and plants. Furthermore, it will be also one of future topics to apply the proposed algorithm to practical applications.

This work was supported by Grant-in-Aid for Young Scientists (B) 20760278.

References

[1] A. Bemporad and C. Filippi, "Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming," J. Optim. Theory Appl., vol.117, no.1, pp.9–38, 2003.
[2] A. Ingimundarson, C. Ocampo-Martinez, and A. Bemporad, "Suboptimal model predictive control of hybrid systems based on mode-switching constraints," Proc. 43rd IEEE Conf. Decision and Control, pp.5264–5269, 2007.
[3] K. Hiraishi, "Solving optimization problems on hybrid systems by graph exploration," Proc. 8th Int'l Workshop on Discrete Event Systems, pp.282–287, 2006.
[4] K. Hiraishi, K. Kobayashi, and S. Choe, "Discrete abstraction of hybrid systems for finite time intervals," Trans. Soc. Instrum. Control Eng., vol.44, no.9, pp.751–759, 2008.
[5] T.A. Johansen and A. Grancharova, "Approximate explicit constrained linear model predictive control via orthogonal serach tree," IEEE Trans. Autom. Control, vol.48, no.5, pp.810–815, 2003.
[6] H.S. Ko, J.-S. Kim, T.-W. Yoon, M. Lim, D.R. Yang, and I.S. Jun, "Modeling and predictive control of a reheating furnace," Proc. American Control Conf., pp.2725–2729, 2000.
[7] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems, LNCS 538, Springer, 1991.
[8] G. Pannocchia, J.B. Rawlings, and S.J. Wright, "Fast, large-scale model predictive control by partial enumeration," Automatica, vol.43, no.5, pp.852–860, 2006.
[9] Y. Tazaki and J. Imura, "Model predictive control for continuous-time piecewise affine systems based on mode controllability," Proc. IEEE Int'l Conf. on Control Applications, 2006.
[10] D.-L. Yu, D.-W. Yu, and J.B. Gomm, "Neural model adaptation and predictive control of a chemical process rig," IEEE Trans. Control Syst. Technol., vol.14, no.5, pp.828–840, 2006.
[11] http://www.ilog.com/products/cplex/

Appendix: Mode Sequences Obtained in Numerical Example

As the result of the proposed offline computation algorithm in a numerical example, Table A· 1 shows enumerated mode sequences $\Theta_r^1(i)$.

Table A· 1 Enumerated mode sequences $\Theta_r^1(i)$.

	Time											
i	0	1	2	3	4	5	6	7	8	9	10	$\overline{J}_r^1$ [$\times 10^6$]
1	1	2	3	6	3	2	5	2	5	2	5	4.78
2	1	2	3	3	2	5	2	5	2	5	2	5.05
3	1	2	3	3	6	5	4	4	4	4	4	5.05
4	1	2	3	2	5	2	5	2	5	2	5	5.31
5	1	1	2	3	6	3	2	5	2	5	2	5.47
6	1	2	3	6	3	6	5	4	4	4	4	5.58
7	1	1	2	3	3	6	5	4	4	4	4	5.74
8	1	1	4	4	4	4	4	4	4	4	4	5.85
9	1	1	2	3	3	2	5	2	5	2	5	5.87
10	1	1	2	3	2	5	2	5	2	5	2	6.14
11	1	1	2	3	6	3	6	5	4	4	4	6.27
12	1	1	1	2	3	6	3	2	5	2	5	6.30
13	1	2	5	2	5	2	5	2	5	2	5	6.38
14	1	1	1	4	4	4	4	4	4	4	4	6.54
15	1	1	1	2	3	3	2	5	2	5	2	6.56
16	1	1	1	2	3	3	6	5	4	4	4	6.56
17	1	1	1	2	3	2	5	2	5	2	5	6.83
18	1	1	1	1	2	3	6	3	2	5	2	6.99
19	1	1	2	3	6	5	4	4	4	4	4	7.07
20	1	1	1	2	3	6	3	6	5	4	4	7.10
21	1	1	1	1	2	3	3	6	5	4	4	7.26
22	1	1	1	1	4	4	4	4	4	4	4	7.36
23	1	1	1	1	2	3	3	2	5	2	5	7.66
24	1	1	1	1	2	3	6	3	6	5	4	7.79
25	1	1	1	2	3	6	5	4	4	4	4	7.90
26	1	1	1	1	2	3	6	3	6	5	2	7.95
27	1	1	1	1	2	3	2	5	2	5	2	8.06
28	1	1	1	2	3	6	3	6	5	2	5	8.08
29	1	1	2	3	6	3	6	5	2	5	2	8.21
30	1	2	3	6	3	6	5	2	5	2	5	8.35
31	1	1	1	1	2	3	6	5	4	4	4	8.59
32	1	1	1	1	1	2	3	2	5	2	5	8.88
33	1	1	1	2	3	3	6	5	2	5	2	9.31
34	1	1	1	1	1	2	5	2	5	2	5	9.41
35	1	1	2	3	3	6	5	2	5	2	5	9.44
36	1	2	3	3	6	5	2	5	2	5	2	9.57
37	1	1	1	1	1	1	2	5	2	5	2	10.11
38	1	1	1	1	2	3	6	5	2	5	2	10.53
39	1	1	1	2	3	6	5	2	5	2	5	10.66
40	1	1	2	3	6	5	2	5	2	5	2	10.80
41	1	2	3	6	5	2	5	2	5	2	5	10.93
42	1	2	3	6	5	4	4	4	4	4	4	10.93
43	1	1	1	1	1	1	1	4	4	4	4	14.50
44	1	1	1	1	1	1	4	4	4	4	4	14.63
45	1	1	1	1	1	4	4	4	4	4	4	14.76
46	1	1	1	1	2	5	2	5	2	5	2	14.90
47	1	1	1	2	5	2	5	2	5	2	5	15.03
48	1	1	2	5	2	5	2	5	2	5	2	15.16
49	1	4	4	4	4	4	4	4	4	4	4	15.30
50	1	1	1	1	2	3	3	6	5	2	5	91.72

Koichi Kobayashi received the M.E. degree from Hosei University in 2000, and the D.E. degree from Tokyo Institute of Technology in 2007. From 2000 to 2004, he worked at Nippon Steel Corporation. He is currently an assistant professor at School of Information Science, Japan Advanced Institute of Science and Technology. His research interests include control of hybrid systems. He is a member of the SICE, ISCIE, and IEEE.

Kunihiko Hiraishi received from the Tokyo Institute of Technology the B.E. degree in 1983, the M.E. degree in 1985, and the D.E. degree in 1990. He is currently a professor at School of Information Science, Japan Advanced Institute of Science and Technology. His research interests include discrete event systems and formal verification. He is a member of the IEEE, SICE, and IPSJ.

Nguyen Van Tang received the M.S. degree from Vietnam National University in 2005. He is currently a Ph.D. candidate at School of Information Science, Japan Advanced Institute of Science and Technology. His research interests include model checking and software verification.