The 23rd International Technical Conference on Circuits/Systems,
Computers and Communications (ITC-CSCC 2008)

On-line and Off-line Based Approximation Algorithm for Model Predictive Control of Hybrid Systems

Koichi Kobayashi[1], Nguyen Van Tang[1], and Kunihiko Hiraishi[1]
[1] School of Information Science, Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
E-mail : {k-kobaya,tang_nguyen,hira}@jaist.ac.jp

Abstract: In this paper, a new approximate algorithm for model predictive control of hybrid systems is proposed. The proposed algorithm consists of the off-line computation and the on-line computation. In the off-line computation, lower and upper bounds of the optimal value of a given cost function for each mode sequence are calculated. In the on-line computation, after the mode sequence is decided by using off-line computation results, the finite-time optimal control problem, i.e., the quadratic programming problem is solved. So the reduction of the computation time in the on-line computation is achieved. In this paper, the effectiveness of the proposed algorithm is shown by a numerical example.

1. Introduction

Hybrid systems consist of continuous dynamical systems such as differential/difference equations and discrete dynamical systems such as if-then rules and finite automata. In the last few years, there have been a lot of studies contributed to analysis and control of hybrid systems in both the computer science community and the systems and control community. As is well known, analysis/control problems of hybrid systems are in general reduced to a combinatorial optimization problem. So it is difficult to apply the framework of hybrid systems to practical plants.

In particular, model predictive control (MPC) [2], which must solve the finite-time optimal control problem repeatedly at each time step, causes the above difficulty critically. Then it is important from the practical viewpoint to discuss the fast and approximation algorithm with guaranteed accuracy. For the standard linear constrained systems, the off-line computing (precomputing) method of the MPC problem has been proposed (e.g. [4] and so on). This method can be extended to the hybrid systems case. However, in the case that the reference signal is time-varying, the off-line computing method cannot be applied. On the other hand, for the on-line computation of the finite-time optimal control problem of hybrid systems, approximation algorithm has been proposed in [1]. This method is simple and effective, but is only the on-line computation.

In this paper, using on-line and off-line computations, a new approximation algorithm to solve the MPC problem of hybrid systems is proposed. In the proposed method, as the off-line computation, candidates of mode (discrete state) sequences such that the MPC problem is feasible are computed. In the on-line computation, first, from candidates of mode sequences, the suboptimal mode sequence is selected by simple calculations. Next, using the obtained mode sequences, the MPC problem, i.e., the quadratic programming problem, is solved. So in the proposed method, the on-line computation is simple, and can be used to real-time control.

Notation: Let $\mathcal{R}$ and $\mathcal{Z}_+$ express the set of real numbers and the set of semi-positive integers, respectively. Let I_n express the $n \times n$ identity matrix. For a given matrix M, Let M^T express the transpose matrix of M.

2. Problem Formulation

Consider the following discrete-time piecewise affine 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 $x(k) \in \mathcal{X} \subset \mathcal{R}^n$ and $u(k) \in \mathcal{U} \subset \mathcal{R}^m$ are the state and the input, respectively. $\mathcal{X}$ and $\mathcal{U}$ are given closed and bounded convex sets. $I(k) \in \mathcal{M} := \{1,\ 2,\ \cdots,\ M\}$ is the mode of system (discrete state), and suppose that mode transition constraints are given by a finite automaton (directed graph). Further, for guaranteeing the well-posedness of the DT-PWA system (1), assume that $\mathcal{S}_I$ is the bounded convex polyhedron satisfying $\bigcup_{I\in\mathcal{M}} \mathcal{S}_I = \mathcal{X}$ 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 control problem is considered.

Problem 2.1. Suppose that the DT-PWA system (1), the current time $t \in \mathcal{Z}_+$, the current state $x(t) = x_t \in \mathcal{R}^n$ and the current mode $I(t) = I_t \in \mathcal{M}$ are given. Then for the DT-PWA system (1), find $u(k)$, $k = t,\ t+1,\ \cdots,\ t+N-1$ minimizing the cost function

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

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

$$x_d(k) = x_d^r,\ k = t,\ t+1,\ \cdots,\ t+L-1, \tag{3}$$

$$x_d(k) = x_d^{r+1},\ k = t+L,\ t+L+1,\ \cdots,\ t+N. \tag{4}$$

In this paper, we discuss the case that there is the switching from x_d^r to x_d^{r+1}. In practical plants, e.g., the temperature control and the vehicle control, there exists the switching of the reference vector frequently. For simplicity of discussion, the following assumption is made:

Assumption A2.1. The switching number of the reference vector is less than or equal to 1.

In the case that there does not exist the switching of the ref-

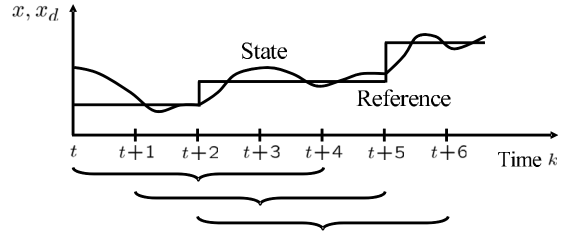

Figure 1. Illustration of MPC with the time-varying reference vector in the case of $N = 4$

erence vector, suppose that $x_d^r = x_d^{r+1}$ holds, and the switching time L is a suitable positive integer in the interval $[1, N]$.

In the MPC problem, Problem 2.1 must be solved repeatedly at each time step $t,\ t+1,\ t+2,\ \cdots$. Then suppose that the reference vector is given by

$$\begin{aligned}
&x_d(k) = x_d^1,\ k = t,\ t+1,\ \cdots,\ t+L_1-1,\\
&x_d(k) = x_d^2,\ k = t+L_1,\ t+L_1+1,\ \cdots,\\
&\qquad\qquad t+L_1+L_2-1,\\
&\quad\vdots\\
&x_d(k) = x_d^\gamma,\ k = t+\sum_{i=0}^{\gamma-1} L_i,\ t+\sum_{i=0}^{\gamma-1} L_i+1,\ \cdots,\\
&\qquad\qquad t+\sum_{i=0}^{\gamma} L_i - 1
\end{aligned} \tag{5}$$

where $L_i \in \mathcal{Z}_+$ satisfying Assumption A2.1. Because in the reference vector of Problem 2.1,

$$\begin{aligned}
\{x_d^r,\ x_d^{r+1}\} \in \Big\{&\{x_d^1,\ x_d^1\},\ \{x_d^2,\ x_d^2\},\ \cdots,\ \{x_d^\gamma, x_d^\gamma\},\\
&\{x_d^1,\ x_d^2\},\ \{x_d^2,\ x_d^3\},\ \cdots,\ \{x_d^{\gamma-1},\ x_d^\gamma\}\Big\}
\end{aligned} \tag{6}$$

holds, note here that the non-switching case must be included in Problem 2.1.

Finally, the illustration of MPC with the time-varying reference vector is shown in Fig. 1.

3. The Proposed Algorithm

Problem 2.1 can be written as the mixed integer quadratic programming (MIQP) problem. So for solving this problem within the practical computational time, Problem 2.1 must be a small scale problem, i.e., the number of modes in (1) must be small. In this paper, an approximation algorithm for Problem 2.1 is proposed. The proposed algorithm consists of the off-line computation algorithm and the on-line computation algorithm. In the proposed off-line algorithm, the list of mode sequences, which are feasible from each initial mode, is generated. Futhermore, the switching of the reference vector is considered. In the proposed on-line algorithm, based on the information obtained via the off-line algorithm, the mode sequence is decided, and Problem 2.1 is solved approximately. Therefore, the computation cost of the on-line algorithm is small, and is useful for practical plants.

3.1 Off-line Algorithm

First, for each reference vector of (5), consider the following cost function

$$\begin{aligned}
J_j(x_t, u) = &\sum_{i=t}^{t+N-1} \left\{\bar{x}_j^T(i)Q\bar{x}_j(i) + u^T(i)Ru(i)\right\}\\
&+\bar{x}_j^T(N)Q_f\bar{x}_j(N),\ j = 1,\ 2,\ \cdots,\ \gamma
\end{aligned} \tag{7}$$

where $\bar{x}_j(i) := x(i) - x_d^j$. Note that there exists no switching action of the reference vector in (7). Then we compute lower and upper bounds of (7) under (1). By using the KCLP-HS tool [3], which is based on the constraint logic programming, lower/upper bounds and feasible mode sequences can be computed easily.

Next, consider the difference between (2) and (7). For fixed $j = r$ and $x_d^r \neq x_d^{r+1}$, the difference between J and J_r is obtained as

$$\begin{aligned}
J - J_r = &\sum_{i=t+L}^{t+N-1} \left\{-2(x_d^{r+1} - x_d^r)^T Qx(i)\right\}\\
&-2(x_d^{r+1} - x_d^r)^T Q_f x(t+N) + \eta
\end{aligned} \tag{8}$$

where η is some constant. Then lower and upper bounds of $J - J_r$ can be computed as follows:

$$\underline{\alpha}_I^l \le -2(x_d^{l+1} - x_d^l)^T Qx \le \overline{\alpha}_I^l,\ \forall x \in \mathcal{S}_I, \tag{9}$$

$$\underline{\beta}_I^l \le -2(x_d^{l+1} - x_d^l)^T Q_f x \le \overline{\beta}_f^l,\ \forall x \in \mathcal{S}_I, \tag{10}$$

$$l = 1,\ 2,\ \cdots,\ \gamma-1,\ I = 1,\ 2,\ \cdots,\ M$$

and

$$\alpha_I^l = \frac{\underline{\alpha}_I^l + \overline{\alpha}_I^l}{2},\ \ \beta_I^l = \frac{\underline{\alpha}_I^l + \overline{\beta}_I^l}{2}.$$

Similarly, for each element of the finite set of (6), lower and upper bounds of $J - J_r$ are computed.

From the above discussion, we obtain

(i) Feasible mode sequences and corresponding lower/upper bounds of (7),

(ii) Lower and upper bounds (9), (10) of all $J - J_r$ of (8).

Remark 1: Obtained mode sequences may not be feasible in the entire region of $\mathcal{S}_{I_t}$. In this case, it is necessary to compute convex polyhedra such that each mode sequence is feasible. Then the procedure of the on-line algorithm in the next subsection can be used similarly. Also, by using lower/upper bounds of (7) and $J - J_r$, the accuracy of the optimal value of (2) can be guaranteed.

3.2 On-line Algorithm

First, for the current state $x(t) = x_t$ and the current mode $I(t) = I_t$, the suboptimal mode sequence is decided. In the case of $x_d^r = x_d^{r+1}$, we select the mode sequence corresponding to the lower bound of (7) among all mode sequences starting from I_t. In the case of $x_d^r \neq x_d^{r+1}$, the lower bound of $J - J_r$ is also used. The details are omitted.

Next, using the obtained mode sequence, Problem 2.1 is solved. Then Problem 2.1 can be written as the quadratic programming (QP) problem. The QP problem can be solved easily by using a suitable solver, e.g., ILOG CPLEX [5].

Finally, we show the proposed on-line algorithm for model predictive control.

Procedure of the on-line computation algorithm:
Step 1: Suppose that x_0 and I_0 are given. Set $t = 0$, $x(0) = x_0$ and $I(0) = I_0$.
Step 2: Check whether these exists the switching action of the reference vector from (5). If there exists the switching action, then go to Step 3, else go to Step 5.
Step 3: Decide the switching time L and x_d^r, x_d^{r+1}.
Step 4: Decide the mode sequence by using the list of mode sequnces and lower and upper bounds (9), (10). Then based on the characteristics of a given plant, the mode sequence must be determined.
Step 5: Decide the reference vector $x_d^r = x_d^{r+1}$ from (5), and select the mode sequence, which corresponds to minimum lower bound in the list of mode sequences.
Step 6: Solve Problem 2.1, i.e., the quadratic programming problem using the derived mode sequence, the current state x_t and the current mode I_t, and find the optimal control input sequence $u(k)$, $k = t,\ t+1,\ \cdots,\ t+N-1$.
Step 7: Apply $u(t)$ to the plant.
Step 8: Set $t = t+1$, and go to Step 2.

The accuracy of the approximate solution obtained by the above procedure is guaranteed in the sense that lower and upper bounds are given by using (9), (10) and so on.

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

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

The partition of the state space and the mode transition constraints are given by Fig. 2 and Fig. 3, respectively. Also, the input constraints are given by $-1 \le u(k) \le +1$. For this DT-PWA system, let us consider the finite-time optimal control problem (Problem 2.1). The current state, the reference vectors, the switching time, and the prediction horizon are given by

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

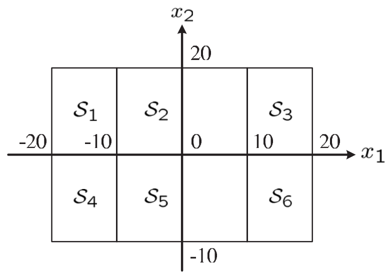

Figure 2. Partition of the state space

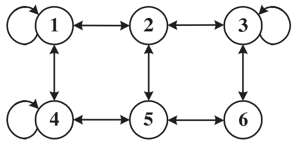

Figure 3. Mode transition constraints expressed via the directed graph

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

Now we consider to use the proposed algorithm. First, by the off-line algorithm, we obtain the list of mode sequences in Table 1. Futhermore, In Table 1, we show only mode sequences from the initial mode 1. (9) is given by

$$\begin{aligned}
-1.2\times10^6 \le -2(x_d^{r+1} - x_d^r)^T Qx \le -0.2\times10^6,& \\ \forall x \in \mathcal{S}_1,& \\
-0.6\times10^6 \le -2(x_d^{r+1} - x_d^r)^T Qx \le +1.0\times10^6& \\ \forall x \in \mathcal{S}_2,& \\
+0.6\times10^6 \le -2(x_d^{r+1} - x_d^r)^T Qx \le +1.6\times10^6& \\ \forall x \in \mathcal{S}_3,& \\
-1.4\times10^6 \le -2(x_d^{r+1} - x_d^r)^T Qx \le -0.6\times10^6& \\ \forall x \in \mathcal{S}_4,& \\
-0.8\times10^6 \le -2(x_d^{r+1} - x_d^r)^T Qx \le +0.6\times10^6& \\ \forall x \in \mathcal{S}_5,& \\
+0.4\times10^6 \le -2(x_d^{r+1} - x_d^r)^T Qx \le +1.2\times10^6& \\ \forall x \in \mathcal{S}_6&
\end{aligned}$$

From $Q = Q_f = 1000I_2$, (9) and (10) are equivalent. From these data, we obtain

$$\begin{cases} \alpha_1^d = -0.7\times10^6, \\ \alpha_2^d = +0.2\times10^6, \\ \alpha_3^d = +1.1\times10^6, \\ \alpha_4^d = -1.0\times10^6, \\ \alpha_5^d = -0.1\times10^6, \\ \alpha_6^d = +0.8\times10^6. \end{cases} \tag{11}$$

In this example, $\min_i |\alpha_i^d| = -0.1 \times 10^6$ $(i = 5)$.

In the on-line algorithm, the mode sequence is selected among mode sequences in Table 1 by using (11). In this

Table 1. Enumerated mode sequences from the initial mode 1, where the unit of lower/upper bounds is $[\times 10^6]$.

	Time											Bounds	
i	0	1	2	3	4	5	6	7	8	9	10	lower	upper
1	1	2	3	3	3	6	5	2	5	2	5	0.9	8.4
2	1	2	3	3	6	5	2	5	2	5	2	0.9	9.6
3	1	2	3	6	3	2	5	2	5	2	5	0.9	4.8
4	1	2	3	3	2	5	2	5	2	5	2	0.9	5.1
5	1	2	3	3	2	5	2	5	2	5	2	1.0	5.3
6	1	2	3	6	5	2	5	2	5	2	5	1.0	10.9
7	1	2	5	2	5	2	5	2	5	2	5	1.0	6.4
8	1	1	2	3	6	3	6	5	2	5	2	1.5	8.2
9	1	1	2	3	6	3	2	5	2	5	2	1.5	5.5
10	1	1	2	3	3	6	5	2	5	2	5	1.5	9.4
11	1	1	2	3	3	2	5	2	5	2	5	1.5	5.9
12	1	1	2	3	2	5	2	5	2	5	2	1.5	6.1
13	1	1	2	3	6	5	2	5	2	5	2	1.5	10.8
14	1	1	2	5	2	5	2	5	2	5	2	1.6	15.2
15	1	1	1	2	3	6	3	6	5	2	5	2.1	8.1
16	1	1	1	2	3	3	6	5	2	5	2	2.1	9.3
17	1	1	1	2	3	6	3	2	5	2	5	2.1	6.3
18	1	1	1	2	3	3	2	5	2	5	2	2.1	6.6
19	1	1	1	2	3	2	5	2	5	2	5	2.1	6.8
20	1	1	1	2	3	6	5	2	5	2	5	2.1	10.7
21	1	1	1	2	5	2	5	2	5	2	5	2.2	15.0
22	1	1	1	1	2	3	6	3	2	5	2	2.7	7.0
23	1	1	1	1	2	3	6	3	6	5	2	2.7	8.0
24	1	1	1	1	2	3	3	2	5	2	5	2.7	7.7
25	1	1	1	1	2	3	3	6	5	2	5	2.7	91.7
26	1	1	1	1	2	3	2	5	2	5	2	2.7	8.1
27	1	1	1	1	2	3	6	5	2	5	2	2.7	10.5
28	1	1	1	1	2	5	2	5	2	5	2	2.8	14.9
29	1	1	1	1	2	3	6	3	6	5	4	3.3	7.8
30	1	1	1	2	3	6	3	6	5	4	4	3.3	7.1
31	1	1	1	1	1	2	3	2	5	2	5	3.3	8.9
32	1	1	2	3	6	3	6	5	4	4	4	3.3	6.3
33	1	2	3	3	3	6	5	4	4	4	4	3.4	5.6
34	1	1	1	1	1	2	5	2	5	2	5	3.4	9.4
35	1	1	1	1	2	3	3	6	5	4	4	3.9	7.3
36	1	1	1	2	3	3	6	5	4	4	4	3.9	6.6
37	1	1	1	1	1	1	2	5	2	5	2	3.9	10.1
38	1	1	2	3	3	6	5	4	4	4	4	4.0	5.7
39	1	2	3	3	6	5	4	4	4	4	4	4.0	5.1
40	1	1	1	1	2	3	6	5	4	4	4	4.6	8.6
41	1	1	1	2	3	6	5	4	4	4	4	4.6	7.9
42	1	1	2	3	6	5	4	4	4	4	4	4.6	7.1
43	1	2	3	6	5	4	4	4	4	4	4	4.6	10.1
44	1	1	4	4	4	4	4	4	4	4	4	5.9	7.1
45	1	1	1	1	1	1	1	1	1	4	4	6.5	7.1
46	1	1	1	1	1	1	1	4	4	4	4	7.0	14.5
47	1	1	1	1	1	1	4	4	4	4	4	7.0	14.6
48	1	1	1	1	1	4	4	4	4	4	4	7.0	14.8
49	1	1	1	1	4	4	4	4	4	4	4	7.1	7.4
50	1	4	4	4	4	4	4	4	4	4	4	7.1	15.3
	1	1	1	1	2	3	6	3	6	5	4	Approximate	
	1	1	1	2	3	3	6	5	4	4	4	Optimal	

exmaple, we obtain the mode sequence "1, 1, 1, 1, 2, 3, 6, 3, 6, 5, 4". The detail is omitted due to the limited space.

Then the state trajectories are shown in Fig. 4. In Fig. 4, for simplicity, the finite-time optimal control problem (Problem 2.1) is solved only one time. From Fig. 4, we see that the state trajectory obtained via the proposed approximation algorithm is similar to the optimal state trajectory.

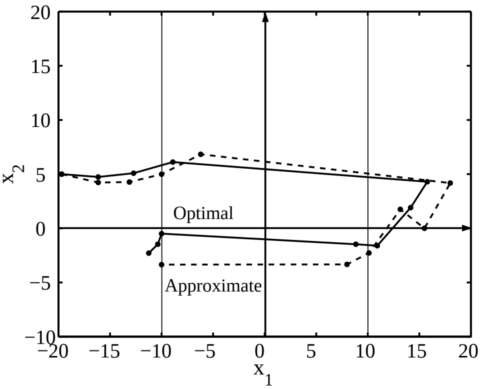

Figure 4. State trajectories

5. Conclusion

In this paper, a new approximation algorithm for model predictive control of hybrid systems has been proposed. The proposed algorithm consists of the off-line and the on-line computation. In particular, because the computational time of the on-line algorithm is similar to that of the quadratic programming problem, the proposed algorithm is useful from the practical viewpoint.

One of the future topics is to estimate more exactly the accuracy of the approximate solution obtained via the proposed algorithm.

References

[1] A. Ingimundarson, C. Ocampo-Martinez, and A. Bemporad, "Suboptimal Model Predictive Control of Hybrid Systems Based on Mode-Switching Constraints," *Proc. of the 43rd IEEE Conf. Decision and Control*, pp. 5264–5269, 2007.

[2] E. F. Camacho and C. Bordons, *Model Predictive Control*, Springer, 2004.

[3] K. Hiraishi, "Solving Optimization Problems on Hybrid Systems by Graph Exploration," *Proc. 8th Intl. Workshop on Discrete Event Systems*, pp. 282–287, 2006.

[4] T. A. Johansen and A. Grancharova, "Approximate Explicit Constrained Linear Model Predictive Control via Orthogonal Serach Tree," *IEEE Trans. on Automatic Control*, Vol. 48, No. 5, pp. 810–815, 2003.

[5] http://www.ilog.com/products/cplex/