

Research Article

Computational Techniques for Model Predictive Control of Large-Scale Systems with Continuous-Valued and Discrete-Valued Inputs

Koichi Kobayashi and Kunihiko Hiraishi

School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan

Correspondence should be addressed to Koichi Kobayashi; k-kobaya@jaist.ac.jp

Received 9 November 2012; Accepted 5 February 2013

Academic Editor: Piyapong Niamsup

Copyright © 2013 K. Kobayashi and K. Hiraishi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose computational techniques for model predictive control of large-scale systems with both continuous-valued control inputs and discrete-valued control inputs, which are a class of hybrid systems. In the proposed method, we introduce the notion of virtual control inputs, which are obtained by relaxing discrete-valued control inputs to continuous variables. In online computation, first, we find continuous-valued control inputs and virtual control inputs minimizing a cost function. Next, using the obtained virtual control inputs, only discrete-valued control inputs at the current time are computed in each subsystem. In addition, we also discuss the effect of quantization errors. Finally, the effectiveness of the proposed method is shown by a numerical example. The proposed method enables us to reduce and decentralize the computation load.

1. Introduction

Control of large-scale systems is one of the fundamental problems in control theory and has been extensively studied so far (see, e.g., [1–7]). From the viewpoint of total optimization, it is necessary to regard a set of individual systems as one system. In recent years, it is important to consider energy management systems such as smart grid [8] and HEMS (Home Energy Management System) [9–11] as an application. As the other example, we give air-conditioning systems [12, 13] and so on. Thus, it is also important at the current stage to consider control of large-scale systems.

In large-scale systems, many kinds of actuators are included. For example, in air-conditioning systems, the output of air conditioners can be regarded as a continuous-valued control input, and ON/OFF switches of ceiling fans can be regarded as a discrete-valued control input. Thus, it is appropriate to consider large-scale systems with both continuous-valued control inputs and discrete-valued control inputs. However, to our knowledge, such a large-scale system has not been directly considered so far. Furthermore, such a large-scale system is a class of hybrid systems, and the finite-time optimal control problem is in general reduced to

a mixed integer quadratic programming (MIQP) problem. For large-scale systems, the computation time for solving the MIQP problem is too long, and it is difficult to realize the model predictive control (MPC) method in which the MIQP problem is solved at each time.

We, thus, propose computational techniques for model predictive control of large-scale systems with both continuous-valued control inputs and discrete-valued control inputs. In the proposed method, we introduce the notion of virtual control inputs, which are obtained by relaxing discrete-valued control inputs to continuous variables. This term has been used in, for example, [12], but we define a different notion. By using the virtual control input, the MIQP problem is approximately rewritten as a quadratic programming (QP) problem, which can be relatively solved faster than the MIQP problem.

In online computation, first, continuous-valued control inputs and virtual control inputs minimizing a cost function are found in a centralized controller. Next, using the obtained virtual control inputs, only discrete-valued control inputs at the current time are computed in each decentralized controller. In other words, a sequence of discrete-valued control inputs is not computed because in MPC only the control

input at the current time is applied to the plant. In a centralized controller, the QP problem is solved at each time. In each decentralized controller, the problem of finding the discrete-valued control input is solved at each time. This problem is reduced to an integer programming problem, or can be solved by the look-up table method in which a look-up table is generated off-line. Thus, the computation load is reduced and decentralized by the proposed method.

In addition, we also discuss the effect of quantization errors based on our previous result [14]. The effectiveness of the proposed method is shown by a numerical example. The proposed method provides us an easy-to-use method in control of large-scale systems.

This paper is organized as follows. In Section 2, large-scale systems studied here are explained, and an example of air-conditioning systems is shown. In Section 3, the optimal control problem is formulated. In Section 4, after the outline of the proposed method is explained, the details are explained. In Section 5, we discuss quantization errors. In Section 6, we show a numerical example. In Section 7, we conclude this paper.

Notation. Let \mathcal{R} denote the set of real numbers. Let \mathcal{Z} denote the set of integers. Let I_n and $0_{m \times n}$ denote the $n \times n$ identity matrix and the $m \times n$ zero matrix, respectively. For simplicity, we sometimes use the symbol 0 instead of $0_{m \times n}$, and the symbol I instead of I_n . For a matrix M , let the matrix $|M|$ denote a matrix such that each element is given by an absolute value of each element of M .

2. Large-Scale Systems

Consider the discrete-time large-scale system consisting of s subsystems given by

$$\Sigma_i : x_i(k+1) = \sum_{j=1}^s A_{ij} x_j(k) + B_i^c u_i^c(k) + B_i^d u_i^d(k), \quad (1)$$

where $i = 1, 2, \dots, s$, $k = 0, 1, 2, \dots$ is the discrete-time, and $x_i(k) \in \mathcal{R}^{n_i}$ is the state in the subsystem i . The control inputs $u_i^c(k) \in \mathcal{U}_i^c \subseteq \mathcal{R}^{m_i^c}$ and $u_i^d(k) \in \mathcal{U}_i^d$ are the continuous-valued control input and the discrete-valued control input in the subsystem i , respectively. The set \mathcal{U}_i^c is given as a closed convex set. The discrete-valued control input $u_i^d(k)$ is given by

$$u_i^d(k) = \begin{bmatrix} u_{i,1}^d(k) \\ u_{i,2}^d(k) \\ \vdots \\ u_{i,m_i^d}^d(k) \end{bmatrix}, \quad (2)$$

$$u_{i,l}^d(k) \in \{u_{i,l,1}^d, u_{i,l,2}^d, \dots, u_{i,l,p_{i,l}}^d\} =: \mathcal{U}_{i,l}^d \subseteq \mathcal{R}^1,$$

where $\mathcal{U}_{i,l}^d$ is the finite set expressing the candidates of the l th element of the discrete-valued control input in the subsystem i , and $p_{i,l}$ is the number of elements of $\mathcal{U}_{i,l}^d$. In system

(1), the pair $(A_{ii}, [B_i^c \ B_i^d])$ expresses the dynamics of the subsystem i , and the matrices A_{ij} , $i \neq j$ express the effect of other subsystems j , $i \neq j$.

We show an example of air-conditioning systems.

Example 1. Suppose that the dynamics of temperature in room i are given by the following first-order plus time delay model:

$$P_i(s) = \frac{K_i}{1 + T_i s} e^{-Ls}, \quad (3)$$

where K_i , T_i , and L are the static process gain, the time constant, and the time delay, respectively. For simplicity of discussion, the value of L does not depend on room i . From (3), we can obtain the following state equation:

$$\begin{aligned} \dot{q}_i(t) &= -\frac{1}{T_i} q_i(t) + \frac{K_i}{T_i} u_i(t-L), \\ y_i(t) &= q_i(t), \end{aligned} \quad (4)$$

where q_i is the state that implies the temperature in one room, and u_i is the continuous-valued control input that implies, for example, the output of a heat pump. Hereafter, the output equation $y_i(t) = q_i(t)$ is omitted.

Next, consider transforming (4) into a discrete-time system. For simplicity of discussion, suppose that the sampling period is given as L . Then, we can obtain the following discrete-time linear system:

$$\begin{aligned} q_i(k+1) &= e^{-L/T_i} q_i(k) + \int_0^L e^{-\tau/T_i} \frac{K_i}{T_i} d\tau u_i(k-1) \\ &= e^{-L/T_i} q_i(k) + K_i (1 - e^{-L/T_i}) u_i(k-1). \end{aligned} \quad (5)$$

By defining $v_i(k) := u_i(k-1)$, we can obtain

$$\begin{aligned} \begin{bmatrix} q_i(k+1) \\ v_i(k+1) \end{bmatrix} &= \begin{bmatrix} e^{-L/T_i} & K_i(1 - e^{-L/T_i}) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} q_i(k) \\ v_i(k) \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_i(k). \end{aligned} \quad (6)$$

Furthermore, consider adding discrete-valued control inputs. We consider three kinds of discrete-valued inputs as follows: (i) $u_i^w \in \{0, 1\}$: window (close/open), (ii) $u_i^h \in \{0, 1, 2\}$: local heater (off/low/high), and (iii) $u_i^v \in \{0, 1, 2\}$: ventilation fan (off/low/high). Then, we can obtain the following temperature model in room i :

$$\begin{aligned} \underbrace{\begin{bmatrix} q_i(k+1) \\ v_i(k+1) \end{bmatrix}}_{x_i(k+1)} &= \underbrace{\begin{bmatrix} e^{-L/T_i} & K_i(1 - e^{-L/T_i}) \\ 0 & 0 \end{bmatrix}}_{A_i} \underbrace{\begin{bmatrix} q_i(k) \\ v_i(k) \end{bmatrix}}_{x_i(k)} \\ &\quad + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{B_i^c} \underbrace{u_i^c(k)}_{u_i^c(k)} + \underbrace{\begin{bmatrix} b_i^w & b_i^h & b_i^v \\ 0 & 0 & 0 \end{bmatrix}}_{B_i^d} \underbrace{\begin{bmatrix} u_i^w(k) \\ u_i^h(k) \\ u_i^v(k) \end{bmatrix}}_{u_i^d(k)}, \end{aligned} \quad (7)$$

Step 3. Apply only $u(t)$ to the plant.

Step 4. Set $t := t + 1$, measure $x(t)$, and return to Step 2.

In the above procedure, Problem 1, that is, the MIQP problem must be solved at each time. However, in large-scale systems, it is hard to solve Problem 1 within the practical computation time. Thus, it is necessary to consider a new method for solving Problem 1 under the situation that Problem 1 is used in MPC.

4. Proposed Method

In this section, first, the outline of the proposed method is explained. Next, the notion of virtual control inputs is proposed. Finally, using virtual control inputs, a solution method for Problem 1 is proposed.

4.1. Outline. For large-scale systems, it is in general difficult to solve Problem 1 within the practical computation time. Here, we focus on the fact that in MPC, only $u(t)$ is applied to the plant. From this fact, if the state at $k = t+1, t+2, \dots, t+N$ can be appropriately evaluated, then computation of the discrete-valued control input at $k = t+1, t+2, \dots, t+N$ is not required. In this paper, we consider relaxing the discrete-valued control input at $k = t, t+1, t+2, \dots, t+N$ to a continuous variable, which is called here a *virtual control input* (see also Figure 3). In the proposed procedure of MPC, first, we find a continuous-valued control input and a virtual control input minimizing the cost function. Next, we derive only the discrete-valued control input at $k = t$ from the virtual control input at $k = t$.

Furthermore, on implementation of the proposed procedure of MPC, we consider both centralized and decentralized controllers (see Figure 4). In the centralized controller, the continuous-valued control input and the virtual control input are computed. In each decentralized controller, the discrete-valued control input at $k = t$ is computed using the virtual control input at $k = t$. Thus, the computation load is decentralized.

Hereafter in this section, first, the notion of the virtual control input will be formally defined. Next, an approximate solution method of Problem 1 will be proposed. Finally, the proposed procedure of MPC will be shown.

4.2. Virtual Control Input. The matrix B_i^d in subsystem (1) is rewritten as

$$B_i^d = S_i \begin{bmatrix} I_{r_i} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} T_i^1 \\ T_i^2 \end{bmatrix}, \quad (13)$$

where $r_i := \text{rank } B_i^d \leq m_i^d$. Then, instead of subsystem (1), we consider the following subsystem:

$$\Sigma_i^l : x_i(k+1) = \sum_{j=1}^s A_{ij} x_j(k) + B_i^c u_i^c(k) + B_i^v u_i^v(k), \quad (14)$$

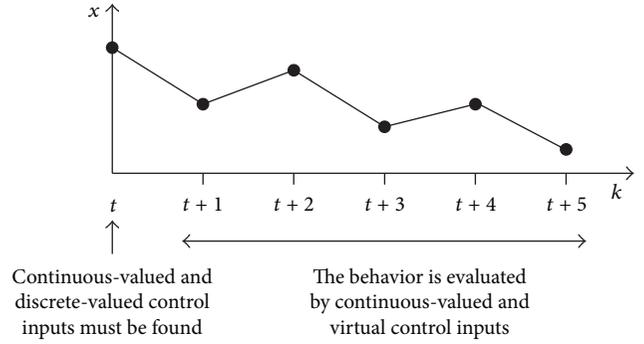


FIGURE 3: Illustration of state trajectory.

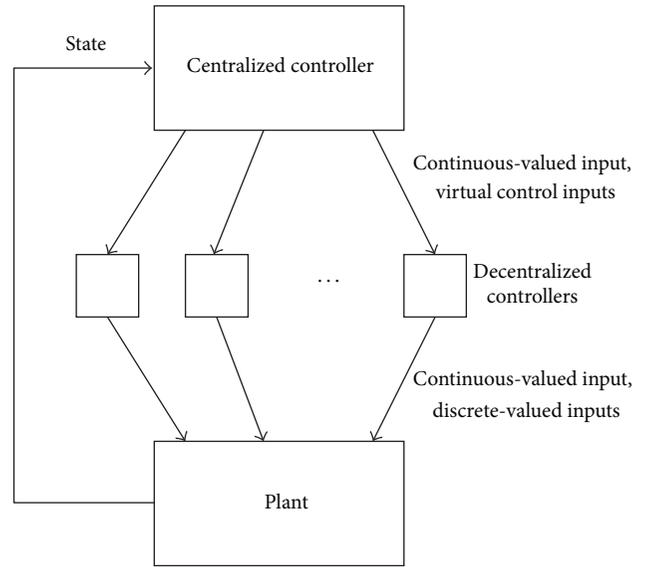


FIGURE 4: Control system considered in this paper.

where

$$B_i^v = S_i \begin{bmatrix} I_{r_i} \\ 0 \end{bmatrix}. \quad (15)$$

The vector $u_i^v(k) \in \mathcal{U}_i^v \subseteq \mathcal{R}^{r_i}$ is the r_i -dimensional continuous-valued control input. The set \mathcal{U}_i^v is given as $\mathcal{U}_i^v = [u_i^v, \bar{u}_i^v]$. The vector $u_i^v(\bar{u}_i^v)$ is given as the value of the vector $T_i^1 u_i^d(k)$ such that a sum of each element of $T_i^1 u_i^d(k)$ is minimized (maximized) and can be derived from the finite set $\mathcal{U}_{i,l}^d$.

Hereafter, $u_i^v(k)$ is called a *virtual control input*. We remark that the discrete-valued control input and the virtual control input are not one-to-one correspondence.

We show one example.

Example 2. Suppose that $B_i^d u_i^d(k)$ in the subsystem (1) is given as

$$B_i^d u_i^d(k) = \begin{bmatrix} -1 & 1 & 2 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} u_{i,1}^d(k) \\ u_{i,2}^d(k) \\ u_{i,3}^d(k) \end{bmatrix}, \quad (16)$$

where

$$\begin{aligned} u_{i,1}^d(k) \in \{0, 1, 2\} &=: \mathcal{U}_{i,1}^d, \\ u_{i,2}^d(k) \in \{0, 2, 4\} &=: \mathcal{U}_{i,2}^d, \\ u_{i,3}^d(k) \in \{0, \pm 3, \pm 5, \pm 7\} &=: \mathcal{U}_{i,3}^d. \end{aligned} \quad (17)$$

Noting here that $\text{rank } B_i^d = 2$ holds, $B_i^d u_i^d(k)$ in (16) can be rewritten as

$$B_i^d u_i^d(k) = \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} u_{i,1}^d(k) - u_{i,2}^d(k) \\ u_{i,3}^d(k) \end{bmatrix}. \quad (18)$$

Thus, we can obtain

$$B_i^v u_i^v(k) = \begin{bmatrix} -1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} u_{i,1}^v(k) \\ u_{i,2}^v(k) \end{bmatrix}. \quad (19)$$

In addition, the constraint for virtual control inputs $u_{i,1}^v(k)$ and $u_{i,2}^v(k)$ is given as

$$\underbrace{\begin{bmatrix} -4 \\ -7 \end{bmatrix}}_{\underline{u}_i^v} \leq \begin{bmatrix} u_{i,1}^v(k) \\ u_{i,2}^v(k) \end{bmatrix} \leq \underbrace{\begin{bmatrix} 2 \\ 7 \end{bmatrix}}_{\bar{u}_i^v}, \quad (20)$$

which can be computed from the finite sets $\mathcal{U}_{i,1}^d$, $\mathcal{U}_{i,2}^d$, and $\mathcal{U}_{i,3}^d$.

Remark 3. Many actuators are included in each subsystem. In the example of air-conditioning systems, we can consider ceiling fans, local heaters, and so on. In many cases, these correspond to discrete-valued control inputs, and the number of these may be greater than the dimension of the state. Thus on the derivation of virtual control inputs, we reduce m_i^d to r_i . Another method is proposed in Section 5.

4.3. Solution Method for Optimal Control Problem. First, define

$$\begin{aligned} u_i^l(k) &:= \left[(u_i^c(k))^T \quad (u_i^v(k))^T \right]^T, \\ u^l(k) &:= \left[(u_1^l(k))^T \quad (u_2^l(k))^T \quad \cdots \quad (u_s^l(k))^T \right]^T. \end{aligned} \quad (21)$$

Then, consider the following finite-time optimal control problem, instead of Problem 1.

Problem 2. Suppose that for the large-scale system consisting of subsystems Σ_i^l , $i = 1, 2, \dots, s$, the current state $x(t) = x_t$ is given, where t is the current time. Then, find both continuous-valued and discrete-valued control inputs $u^l(k)$, $k = t, t+1, \dots, t+N-1$ minimizing the following cost function:

$$\begin{aligned} J^l &= \sum_{k=t}^{t+N-1} \left\{ (x(k) - x_d)^T Q (x(k) - x_d) + u^l(k)^T R^l u^l(k) \right\} \\ &+ (x(t+N) - x_d)^T Q_f (x(t+N) - x_d), \end{aligned} \quad (22)$$

where R^l is the weighting matrix obtained from the weighting matrix R in Problem 1 and the matrix T_i^1 in (13).

Since the decision variable in Problem 2 is only a continuous-valued variable, Problem 2 can be rewritten as a quadratic programming (QP) problem, which can be solved by a suitable solver such as IBM ILOG CPLEX Optimizer [16]. The dimension of the decision variables is given by $\sum_{i=1}^s (m_i^c + r_i)N$. The procedure for rewriting Problem 2 as a QP problem is the same as that in the case of Problem 1. See the appendix for further details. In addition, from the definition of the virtual control input, we see that the minimum value of the cost function in Problem 2 gives the lower bound of the minimum value of the cost function in Problem 1.

By solving Problem 2, we can obtain the virtual control input. Next, we consider finding a discrete-valued control input from the obtained virtual control input. By $u_i^{v*}(k)$, denote the optimal value of the virtual control input obtained by solving Problem 2. In addition, a binary variable $\delta_{i,l,q}$ is assigned to each candidate of each element in the discrete-valued control input. Then, focusing on only $u_i^{v*}(t)$, consider the following problem for each subsystem Σ_i .

Problem 3. Find binary variables $\delta_{i,l,q}$, $l = 1, 2, \dots, m_i^d$, $q = 1, 2, \dots, p_{i,m_i^d}$ minimizing the following cost function:

$$J_d = \left\| T_i^1 \begin{bmatrix} \sum_{q=1}^{p_{i,1}} u_{i,1,q}^d \delta_{i,1,q} \\ \sum_{q=1}^{p_{i,2}} u_{i,2,q}^d \delta_{i,2,q} \\ \vdots \\ \sum_{q=1}^{p_{i,m_i^d}} u_{i,m_i^d,q}^d \delta_{i,m_i^d,q} \end{bmatrix} - u_i^{v*}(t) \right\|_1, \quad (23)$$

under the equality constraint $\sum_{q=1}^{p_{i,l}} \delta_{i,l,q} = 1$, where $\|\cdot\|_1$ is the 1-norm of a vector.

By a simple calculation, Problem 3 can be rewritten as an integer linear programming (ILP) problem, which can be solved by a suitable solver such as IBM ILOG CPLEX Optimizer. We remark that Problem 3 is independent of each subsystem. That is, computing environment can be independently prepared for each subsystem. Furthermore, we remark that only the case of $k = t$ is considered in Problem 3. This is because in MPC only the control input at $k = t$ is applied to the plant. Thus, it is not necessary to find the discrete-valued control input at $k = t+1, t+2, \dots, t+N-1$. Therefore, Problem 3 can be solved fast. In addition, by deriving a look-up table off-line, solving the ILP problem online is not necessary. See the numerical example in Section 6 for further details.

The proposed solution method for Problem 1 can be regarded as the method that the MIQP problem corresponding to Problem 1 is divided into one QP problem (Problem 2) and s ILP problems (Problem 3 for each subsystem). Problem 2 can be in general solved faster than Problem 1, and in

Problem 3, we find only the discrete-valued control input at $k = t$. From these observations, we see that the computation time for solving one QP problem and s ILP problems is smaller than that for solving one MIQP problem.

In addition, the fact that Problem 1 is directly solved (in other words, Problem 1 is rewritten as an MIQP problem) corresponds to centralized control of large-scale systems. In the proposed method, solving Problem 2 (in other words, finding both continuous-valued control inputs and virtual control inputs) corresponds to centralized control. On the other hand, solving Problem 3 (in other words, finding discrete-valued control inputs based on optimal virtual control inputs) corresponds to decentralized control. Thus, the proposed method can be regarded as the method combining centralized control with decentralized control, and the computation load is decentralized and reduced.

4.4. Model Predictive Control Law. Finally, we summarize the procedure of MPC combining centralized control with decentralized control.

The Proposed Procedure of MPC

Step 1. Set $t = 0$ and give the current state $x(t) = x_t$.

Step 2. In the centralized controller, find both a continuous-valued control input and a virtual control input by solving Problem 2.

Step 3. Send the optimal values of both a continuous-valued control input and a virtual control input from the centralized controller to each decentralized controller.

Step 4. In each decentralized controller, solve Problem 3 and find a discrete-valued control input at t .

Step 5. Apply only the control input at t to the plant.

Step 6. Set $t := t + 1$, measure $x(t)$, and return to Step 2.

5. Discussion on Quantization Errors

In the previous section, we do not consider quantization errors and focus on the lower bound of the cost function. In this section, we discuss a method for considering quantization errors.

First, the virtual control input is introduced. In this section, $B_i^d u_i^d(k)$ in (1) is transformed into the virtual control input. By $u_i^q(k) \in \mathcal{R}^n$, denote the virtual control input. By \underline{u}_i^q , denote the value of the vector $B_i^d u_i^d(k)$ such that a sum of each element of $B_i^d u_i^d(k)$ is minimized. By \bar{u}_i^q , denote the value of the vector $B_i^d u_i^d(k)$ such that a sum of each element of $B_i^d u_i^d(k)$ is maximized. Then, instead of (1), we consider the following system:

$$\Sigma_i^l : x_i(k+1) = \sum_{j=1}^s A_{ij} x_j(k) + B_i^c u_i^c(k) + u_i^q(k) + a_i^q, \quad (24)$$

where $u_i^q(k) \in [\underline{u}_i^q, \bar{u}_i^q] \subseteq \mathcal{R}^n$. The vector $a_i^q \in [\underline{a}_i^q, \bar{a}_i^q]$ is the uncertain affine term including quantization errors, and $\underline{a}_i^q, \bar{a}_i^q$ are obtained from quantization errors in each element of $B_i^d u_i^d(k)$.

We show one example.

Example 4. Suppose that $B_i^d u_i^d(k)$ in subsystem (1) is given as

$$B_i^d u_i^d(k) = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{i,1}^d(k) \\ u_{i,2}^d(k) \\ u_{i,3}^d(k) \end{bmatrix}, \quad (25)$$

where $u_{i,1}^d(k) \in \{0, \pm 1, \dots, \pm 6\}$, $u_{i,2}^d(k) \in \{0, \pm 6\}$, and $u_{i,3}^d(k) \in \{0, \pm 1, \dots, \pm 10\}$. Then, we can obtain the following virtual control inputs:

$$u_i^q(k) = \begin{bmatrix} u_{i,1}^d(k) + u_{i,2}^d(k) \\ u_{i,3}^d(k) \end{bmatrix}. \quad (26)$$

The vectors $\underline{u}_i^q, \bar{u}_i^q$ can be obtained as

$$\begin{bmatrix} -12 \\ -10 \end{bmatrix} \leq u_i^q(k) \leq \begin{bmatrix} 12 \\ 10 \end{bmatrix}. \quad (27)$$

In addition, the following relations hold:

$$\begin{aligned} u_{i,1}^d(k) + u_{i,2}^d(k) &\in \{-12, -11, \dots, 12\} \subset \mathcal{L}, \\ u_{i,3}^d(k) &\in \{-10, -9, \dots, 10\} \subset \mathcal{L}. \end{aligned} \quad (28)$$

From these relations, $\underline{a}_i^q, \bar{a}_i^q$ can be obtained as

$$\underline{a}_i^q = \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix}, \quad \bar{a}_i^q = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}. \quad (29)$$

Thus, we can obtain virtual control inputs and the uncertain affine term.

Next, consider transforming the system (24) including uncertainty into a linear system including no uncertainty. For simplicity of discussion, we assume that, for all $k \geq 0$, two relations $x_i(k) \geq 0$ and $u_i^q(k) \geq 0$ hold. This assumption can be satisfied by adding the affine term. Then, by using the result in [14], an overapproximation of the state can be expressed as

$$\begin{aligned} x_i(k+1) &\in [\underline{x}_i(k+1), \bar{x}_i(k+1)], \\ \underline{x}_i(k+1) &= \sum_{j=1}^s A_{ij} x_j^c(k) - |A_{ij}| x_j^r(k) \\ &\quad + B_i^c u_i^c(k) + u_i^q(k) + \underline{a}_i^q + b, \end{aligned} \quad (30)$$

$$\begin{aligned} \bar{x}_i(k+1) &= \sum_{j=1}^s A_{ij} x_j^c(k) + |A_{ij}| x_j^r(k) \\ &\quad + B_i^c u_i^c(k) + u_i^q(k) + \bar{a}_i^q + b, \end{aligned} \quad (31)$$

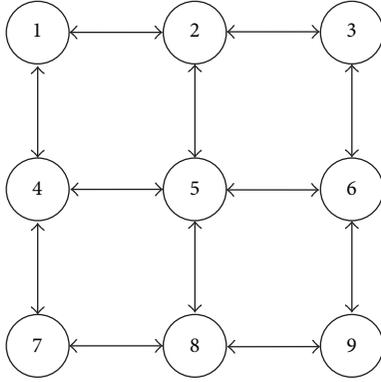


FIGURE 5: Directed graph expressing couplings.

where b is the affine term for satisfying the above assumption, and

$$\begin{bmatrix} x_i^c(k) \\ x_i^r(k) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} I_n & I_n \\ -I_n & I_n \end{bmatrix} \begin{bmatrix} \underline{x}_i(k) \\ \bar{x}_i(k) \end{bmatrix}. \quad (32)$$

Equations (30) and (31) are a discrete-time linear system, and by using (31), the worst case of the cost function (12) in Problem 1 can be minimized approximately. This problem is reduced to a quadratic programming (QP) problem, not a min-max optimization problem. On the other hand, by using both (30) and (31), the middle point of the cost function (12) can be evaluated. Also in this case, the optimal control problem is reduced to a QP problem.

Remark 5. In this section, the dimension of the virtual control input is n_i . On the other hand, in Section 4.2, the dimension of the virtual control input is $r_i = \text{rank } B_i^d \leq n_i$. Also in the discussion on quantization errors, the dimension of the virtual control input can be reduced to r_i . However, \underline{u}_i^q , \bar{u}_i^q and \underline{a}_i^q , \bar{a}_i^q must be determined depending on the choice of T_i^1 , T_i^2 in (13). In order to avoid this complexity, we have considered in this section the virtual control input with the dimension of n_i .

6. Numerical Example

As a numerical example, consider the large-scale system consisting of 9 subsystems given by (1). Couplings of subsystems are given by the directed graph in Figure 5. The matrices A_{ii} , $i = 1, 2, \dots, 9$, are given by

$$A_{ii} = \begin{bmatrix} a_i & 0.6 \\ 0 & 1 \end{bmatrix}, \quad (33)$$

where $a_1 = 1.3$, $a_2 = 1.2$, $a_3 = 1.1$, $a_4 = 1.0$, $a_5 = 0.9$, $a_6 = 0.8$, $a_7 = 0.7$, $a_8 = 0.6$, and $a_9 = 0.5$. The matrices A_{ij} , $i \neq j$ are given as follows. $A_{12}, A_{21}, A_{23}, A_{32}, A_{14}, A_{41}, A_{25}, A_{52}, A_{36}, A_{63}, A_{45}, A_{54}, A_{56}, A_{65}, A_{47}, A_{74}, A_{58}, A_{85}, A_{69}, A_{96}, A_{78}, A_{87}, A_{89}, A_{98}$ are given as $0.4I_2$, respectively. Other coupling terms A_{ij} , $i \neq j$ are given as a zero matrix. B_i^c is given by

$$B_i^c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (34)$$

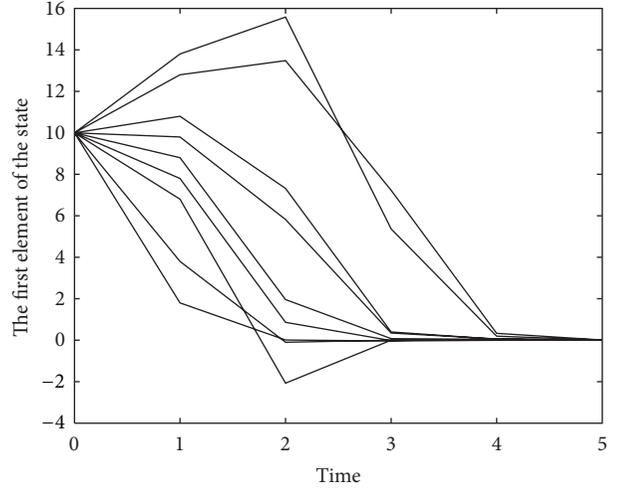


FIGURE 6: Trajectories of the first element of the state.

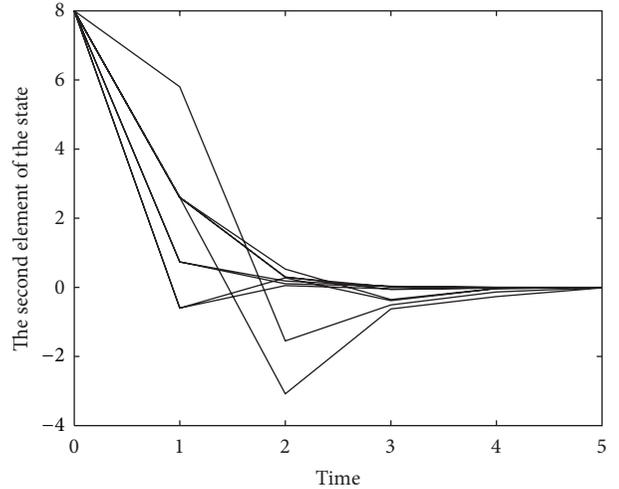


FIGURE 7: Trajectories of the second element of the state.

In addition, we impose the following constraint:

$$-10 \leq u_i^c(k) \leq +10. \quad (35)$$

The matrix B_i^d and the discrete-valued control input $u_i^d(k)$ are given by those in Example 2.

For this system, we consider solving Problem 2, instead of Problem 1. The prediction horizon, the weighting matrices, the desired state, the initial time, and the initial state are given as $N = 5$, $Q = 10I$, $R' = I$, $x_d = 0$, $t = 0$, and $x_i(0) = [10 \ 8]^T$, respectively.

We show the computation result. Figures 6 and 7 show state trajectories of 9 subsystems. Note here that in these figures, the virtual control input is used. From this fact, we see that trajectories in these figures imply those corresponding to the lower bound of the optimal value of a given cost function.

Next, we discuss how to derive the discrete-valued control input from the virtual control input. In this example, the optimal virtual control input at $k = 0$ can be obtained by

$$u_i^{v*}(0) = \begin{bmatrix} 2 \\ -7 \end{bmatrix}, \quad i = 1, 2, \dots, 9. \quad (36)$$

Then focusing on (18), we can obtain the discrete-valued control input as follows:

$$u_i^d(0) = \begin{bmatrix} u_{i,1}^d(0) \\ u_{i,2}^d(0) \\ u_{i,3}^d(0) \end{bmatrix} \in \left\{ \begin{bmatrix} 2 \\ 0 \\ -7 \end{bmatrix}, \begin{bmatrix} 4 \\ 2 \\ -7 \end{bmatrix} \right\}. \quad (37)$$

In this case, $u_{i,3}^d(0)$ can be uniquely determined. $u_{i,1}^d(0)$ and $u_{i,2}^d(0)$ are not uniquely determined. If it is desirable to set a small value, then we can obtain $u_{i,1}^d(0) = 2$ and $u_{i,2}^d(0) = 0$. In practice, the look-up table can be generated off-line. Then, the input of this table is the rounded virtual control input, and the output is the discrete-valued control input. In online computation, the discrete-valued control input can be determined from the look-up table and the rounded virtual control input. In this method, it is not necessary to solve Problem 3.

Finally, we explain the computation time for solving Problem 2. In this example, the computation time of the QP problem was 0.05 [sec], where we used IBM ILOG CPLEX Optimizer 11.0 as an MIQP solver on the computer with the Intel Core 2 Duo 3.0 GHz processor and the 4 GB memory. Thus, Problem 2 can be solved fast.

7. Conclusion

In this paper, we have proposed computational techniques for model predictive control of large-scale systems with both continuous-valued and discrete-valued control inputs. By introducing the notion of virtual control inputs, the computation load is reduced and decentralized. The effectiveness of the proposed method has been shown by a numerical example. The proposed method is useful for solving the control problem of large-scale system with both continuous-valued and discrete-valued control inputs.

One of the future works is to apply our approach to several practical systems such as air-conditioning systems. On the other hand, a large-scale system studied in this paper is a class of hybrid systems. It is also important to extend the proposed method to general hybrid systems such as piecewise affine systems and MLD systems.

Appendix

Reduction of Problem 1 to an MIQP Problem

First, by using a binary variable, the element $u_{i,l}^d(k)$ of $u_i^d(k)$ in subsystem (1) can be expressed as

$$\begin{aligned} u_{i,l}^d(k) &= [u_{i,l,1}^d \ \cdots \ u_{i,l,p_{i,l}}^d] \begin{bmatrix} \delta_{i,l,1}(k) \\ \vdots \\ \delta_{i,l,p_{i,l}}(k) \end{bmatrix} \\ &= C_{i,l}^d \delta_{i,l}(k), \end{aligned} \quad (A.1)$$

where we imposed the following equality constraint:

$$\delta_{i,l,1}(k) + \cdots + \delta_{i,l,p_{i,l}}(k) = 1. \quad (A.2)$$

By using $C_{i,l}^d$ and $\delta_{i,l}(k)$, $u_i^d(k)$ can be expressed as

$$u_i^d(k) = \begin{bmatrix} C_{i,1}^d & 0 \\ \vdots & \vdots \\ 0 & C_{i,m_i}^d \end{bmatrix} \begin{bmatrix} \delta_{i,1}(k) \\ \vdots \\ \delta_{i,m_i}(k) \end{bmatrix} = C_i^d \delta_i(k). \quad (A.3)$$

Next, by combing all subsystems, we can obtain

$$x(k+1) = Ax(k) + Bu(k), \quad (A.4)$$

where

$$\begin{aligned} A &= \begin{bmatrix} A_{11} & \cdots & A_{1s} \\ \vdots & \ddots & \vdots \\ A_{s1} & \cdots & A_{ss} \end{bmatrix}, \\ B &= \begin{bmatrix} [B_1^c \ B_1^d C_1^d] & & 0 \\ & \ddots & \\ 0 & & [B_s^c \ B_s^d C_s^d] \end{bmatrix}. \end{aligned} \quad (A.5)$$

See Section 3 for the definitions of $x(k)$ and $u(k)$. In addition, the constraint for the continuous-valued control input and the equality constraint of (A.2) can be expressed as

$$Cx(k) + Du(k) \leq E. \quad (A.6)$$

Since $u(k)$ consist of both continuous variables and binary variables, the pair of (A.4) and (A.6) is a class of mixed logical dynamical systems [15]. From (A.4) and (A.6), we have

$$\bar{x} = \bar{A}x_0 + \bar{B}\bar{u}, \quad (A.7)$$

$$\bar{C}\bar{x} + \bar{D}\bar{u} \leq \bar{E}, \quad (A.8)$$

where

$$\begin{aligned}\bar{x} &:= [x^T(t) \ x^T(t+1) \ \cdots \ x^T(t+N)]^T, \\ \bar{u} &:= [u^T(t) \ u^T(t+1) \ \cdots \ u^T(t+N-1)]^T, \\ \bar{A} &= \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A^{N-1}B & \cdots & AB & B \end{bmatrix}, \\ \bar{C} &= \begin{bmatrix} C & 0 & 0 \\ & \ddots & \vdots \\ 0 & C & 0 \end{bmatrix}, \quad \bar{D} = \begin{bmatrix} D & 0 \\ & \ddots \\ 0 & D \end{bmatrix}, \\ \bar{E} &= \begin{bmatrix} E \\ \vdots \\ E \end{bmatrix}.\end{aligned}\tag{A.9}$$

By substituting (A.7) into (A.8), we have

$$(\bar{D} - \bar{C}\bar{B})\bar{u} \leq \bar{E} - \bar{C}\bar{A}x_0.\tag{A.10}$$

Finally, consider the cost function (12). For simplicity of discussion, we consider the case of $x_d = 0$. Then the cost function (12) can be rewritten as

$$J = \bar{x}^T \bar{Q} \bar{x} + \bar{u}^T \bar{R} \bar{u},\tag{A.11}$$

where

$$\bar{Q} = \begin{bmatrix} Q & & 0 \\ & \ddots & \\ 0 & & Q & Q_f \end{bmatrix}, \quad \bar{R} = \begin{bmatrix} R & 0 \\ & \ddots \\ 0 & R \end{bmatrix}.\tag{A.12}$$

Furthermore, by substituting (A.7) into this cost function, we can obtain

$$J = \bar{u}^T (\bar{R} + \bar{B}^T \bar{Q} \bar{B}) \bar{u} + 2x_0^T \bar{A}^T \bar{Q} \bar{B} \bar{u} + x_0^T \bar{A}^T \bar{Q} \bar{A} x_0.\tag{A.13}$$

From the above, Problem 1 can be rewritten as the following MIQP problem: find a control input sequence \bar{u} minimizing the cost function (A.13) subject to the constraint (A.10). The obtained MIQP problem can be solved by a suitable solver such as IBM ILOG CPLEX Optimizer. However, for large-scale systems, it is difficult to solve the MIQP problem with practical computation time.

Acknowledgment

This work was partially supported by Grant-in-Aid for Young Scientists (B) 23760387.

References

- [1] A. Alessio, D. Barcelli, and A. Bemporad, "Decentralized model predictive control of dynamically coupled linear systems," *Journal of Process Control*, vol. 21, no. 5, pp. 705–714, 2011.
- [2] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.
- [3] M. Ikeda, D. D. Šiljak, and K. Yasuda, "Optimality of decentralized control for large-scale systems," *Automatica*, vol. 19, no. 3, pp. 309–316, 1983.
- [4] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, no. 12, pp. 2105–2115, 2006.
- [5] M. Rotkowitz and S. Lall, "A characterization of convex problems in decentralized control," *IEEE Transactions on Automatic Control*, vol. 51, no. 2, pp. 1984–1996, 2006.
- [6] N. R. Sandell, Jr., P. Varaiya, M. Athans, and M. G. Safonov, "Survey of decentralized control methods for large scale systems," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 108–128, 1978.
- [7] H. Tamura and T. Yoshikawa, *Large-Scale Systems Control and Decision Making*, Marcel Dekker, 1990.
- [8] About IEEE Smart Grid, <http://smartgrid.ieee.org/ieee-smart-grid>.
- [9] J. Han, C.-S. Choi, and I. Lee, "More efficient home energy management system based on ZigBee communication and infrared remote controls," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 85–89, 2011.
- [10] M. Inoue, T. Higuma, Y. Ito, N. Kushiro, and H. Kubota, "Network architecture for home energy management system," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 3, pp. 606–613, 2003.
- [11] N. Kushiro, S. Suzuki, M. Nakata, H. Takahara, and M. Inoue, "Integrated residential gateway controller for home energy management system," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 3, pp. 629–636, 2003.
- [12] M. Rodriguez, C. De Prada, F. Capraro, S. Cristea, and R. De Keyser, "Hybrid predictive control of a solar air conditioning plant," in *Proceedings of the 17th IFAC World Congress*, pp. 952–957, 2008.
- [13] C. Sonntag, H. Ding, and S. Engell, "Supervisory control of a solar air conditioning plant with hybrid dynamics," *European Journal of Control*, vol. 6, pp. 451–463, 2008.
- [14] K. Kobayashi and K. Hiraishi, "Analysis and control of hybrid systems with parameter uncertainty based on interval methods," in *Proceedings of the American Control Conference (ACC '09)*, pp. 3632–3637, 2009.
- [15] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.
- [16] IBM ILOG CPLEX Optimizer, <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>.