PAPER *Special Section on Mathematical Systems Science and its Applications*

Predictive Pinning Control with Communication Delays for Consensus of Multi-Agent Systems

Koichi KOBAYASHI[†a)], ***Member***

SUMMARY In this paper, based on the policy of model predictive control, a new method of predictive pinning control is proposed for the consensus problem of multi-agent systems. Pinning control is a method that the external control input is added to some agents (pinning nodes), e.g., leaders. By the external control input, consensus to a certain target value (not the average of the initial states) and faster consensus are achieved. In the proposed method, the external control input is calculated by the controller node connected to only pinning nodes. Since the states of all agents are required in calculation of the external control input, communication delays must be considered. The proposed algorithm includes not only calculation of the external control input but also delay compensation. The effectiveness of the proposed method is presented by a numerical example.

key words: *consensus, model predictive control (MPC), multi-agent systems, pinning control*

1. Introduction

In the last decade, control of multi-agent systems has attracted much attention. Especially, the consensus problem has been widely studied so far (see, e.g., [5], [7], [9], [13]–[15], [18]). The consensus problem is to find a control input such that the agents reach a particular ordered state by using only information on neighborhood agents. There are many applications such as load balancing, unmanned aerial vehicles, and so on.

Controllers in the consensus problem are a kind of distributed controllers. In basic consensus problems, the states of all agents converge to the average of initial states, and the convergence speed depends on a given graph expressing neighborhood agents. To achieve consensus to the other target value and faster consensus, it is important to consider the external control inputs. From this viewpoint, pinning control has been proposed (see, e.g., [16], [17]). Pinning control is a method that the external control input is added to some agents (pinning nodes), e.g., leaders.

In many existing methods of pinning control, a simple controller using only information on neighborhood agents is utilized. On the other hand, it is also important to develop optimization-based methods such as model predictive control (MPC). MPC is a control method that the control input is generated by solving the finite-time optimal control problem at each discrete time (see, e.g., [4], [11]). In [18], predictive pinning control for the consensus problem has been proposed. In this method, a controller can be obtained based on the policy of MPC, and the convergence rate is analyzed theoretically. However, there are two weaknesses. First, input constraints cannot be imposed. Next, it is assumed that the pinning nodes are connected to all nodes. In other words, communication delays are not considered.

In this paper, we propose a new method of predictive pinning control with communication delays. First, we add a controller node to a network of multi-agent systems. The controller node is connected to only pinning nodes, and solves the finite-time optimal control problem. Communication delays are characterized by the number of edges in paths from each node to the controller node. Next, we propose an on-line algorithm consisting of both estimation of the initial state and calculation of the external control input for the pinning nodes. In calculation of the external control input, we impose input constraints. Then, the finite-time optimal control problem is reduced to a quadratic programming (QP) problem. Finally, using a numerical example, we show the effectiveness of the proposed method from the viewpoints of the control performance and the computation time.

This paper is organized as follows. In Sect. 2, the outline of consensus and pinning control is explained. In Sect. 3, a simple example and the proposed on-line algorithm are explained. In Sect. 4, a numerical example is presented. In Sect. 5, we conclude this paper.

Notation: Let $\mathcal{R}$ denote the set of real numbers. 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 of notation, we sometimes use the symbol 0 instead of $0_{m\times n}$, and the symbol I instead of I_n. Let $1_{m\times n}$ denote the $m \times n$ matrix whose elements are all one. For the vector $a = [a_1\ a_2\ \cdots\ a_n]^\top$, let $\mathrm{diag}(a)$ denote the diagonal matrix, i.e., $\mathrm{diag}(a) = \begin{bmatrix} a_1 & & 0 \\ & \ddots & \\ 0 & & a_n \end{bmatrix}$. For the matrices $A_1, A_2, \ldots, A_n$, let block-diag$(A_1, A_2, \ldots, A_n)$ denote the block diagonal matrix, i.e., block-diag$(A_1, A_2, \ldots, A_n) = \begin{bmatrix} A_1 & & 0 \\ & \ddots & \\ 0 & & A_n \end{bmatrix}$.

2. Preliminaries

In this section, first, the consensus problem of multi-agent systems is explained. Next, the outline of pinning control is

Manuscript received April 14, 2018.
Manuscript revised August 5, 2018.

[†]The author is with the Graduate School of Information Science and Technology, Hokkaido University, Sapporo-shi, 060-0814 Japan.
a) E-mail: k-kobaya@ssi.ist.hokudai.ac.jp

DOI: 10.1587/transfun.E102.A.359

explained.

2.1 Consensus Problem

Let $G = (\mathcal{V}, \mathcal{E})$ denote an undirected connected graph, where the $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of nodes (vertices) and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of edges. For simplicity of discussion, we consider undirected graphs, but we can easily extend the following discussion to that in the case of directed graphs. Each node corresponds to each agent, and each edge corresponds to a communication link between agents. If there is the edge from the node i to j (i.e., the edge from j to i), then the information about the state in the node i (j) can be transmitted to the node j (i). Let $A \in \{0, 1\}^{n \times n}$ denote the adjacency matrix of G. We assume that there is no self-loop, that is, (i, i)-th element of A is zero. Let $\mathcal{N}_i \subset \mathcal{V}$ denote the set of nodes that are adjacent to the node i. Then, the degree matrix D is defined by $D := \mathrm{diag}([|\mathcal{N}_1|\ |\mathcal{N}_2|\ \cdots\ |\mathcal{N}_n|]^\top)$. In addition, the graph Laplacian matrix L is defined by $L := D - A$.

Next, the dynamics of the agent $i \in \{1, 2, \ldots, n\}$ are defined by the following discrete-time single integrator:

$$x_i(k+1) = x_i(k) + u_i(k), \tag{1}$$

where $x_i \in \mathcal{R}$ and $u_i \in \mathcal{R}$ are the state and the control input of the agent i, respectively. For the system (1), the consensus problem is formulated as follows.

Problem 1: It is said that the agents have reached consensus if $\lim_{k \to \infty}(x_i(k) - x_j(k)) = 0$ holds for all $i, j \in \mathcal{V}$. Then, find a control input such that the agents have reached consensus, where $u_i(k)$ must be given by a function with respect to only $x_i(k)$ and $x_j(k)$, $j \in \mathcal{N}_i$.

The solution for this problem is given by the following lemma (see [13] for further details).

Lemma 1: Consider the following controller:

$$u_i(k) = -\varepsilon \sum_{j \in \mathcal{N}_i} (x_i(k) - x_j(k)), \tag{2}$$

where $\varepsilon \in (0, 1/d_{\max})$, $d_{\max} = \max(|\mathcal{N}_1|, |\mathcal{N}_2|, \ldots, |\mathcal{N}_n|)$. Then, the agents have reached consensus, that is, the following relation holds:

$$\lim_{k \to \infty} x_i(k) = \frac{1}{n} \sum_{i=1}^{n} x_i(0). \tag{3}$$

From this lemma, we see that all states converge to the average of the initial states. The closed-loop system consisting of (1) and (2) can be obtained by

$$x(k+1) = Px(k), \tag{4}$$

where $x = [x_1\ x_2\ \ldots\ x_n]^\top$ and $P = I_n - \varepsilon L$.

2.2 Pinning Control

In pinning control, we introduce pinning nodes. Pinning nodes may be regarded as "leaders". Pinning control is a method that only agents corresponding to pinning nodes are controlled by the external signals. Using pinning control, consensus to the different target value (not the average of the initial states) is achieved. We can also consider faster consensus.

Without loss of generality, the set of pinning nodes is given by $\mathcal{V}_p = \{1, 2, \ldots, m\} \subset \mathcal{V}$, $m \ll n$. In pinning nodes, the external control input $v_i(k) \in \mathcal{R}$ is added, that is,

$$u_i(k) = -\varepsilon \sum_{j \in \mathcal{N}_i} (x_i(k) - x_j(k)) + v_i(k),\ i \in \mathcal{V}_p, \tag{5}$$

$$u_i(k) = -\varepsilon \sum_{j \in \mathcal{N}_i} (x_i(k) - x_j(k)),\ i \in \mathcal{V} \setminus \mathcal{V}_p. \tag{6}$$

The system consisting of (1), (5), and (6) can be obtained by

$$x(k+1) = Px(k) + Bv(k), \tag{7}$$

where $v = [v_1\ v_2\ \cdots\ v_m]^\top$ and $B = [I_m\ 0_{m \times (n-m)}]^\top$. In this paper, we propose a method to find $v(k)$ based on MPC.

3. Predictive Pinning Control with Communication Delays

In this section, a new method of pinning control with communication delays is proposed based on MPC. First, the notion of a controller node is introduced. After that, a simple example of the proposed predictive control is explained. Next, the finite-time optimal control problem is formulated. Finally, an on-line algorithm is proposed.

3.1 Introduction of Controller Node and Simple Example of Predictive Pinning Control

For the graph G, a controller node is added. As a simple example, consider the undirected graph in Fig. 1. We assume that $m = 1$, that is, only the node 1 is the pinning node. The multi-agent system consists of the nodes 1,2,3,4. The node 5 is the controller node, which is newly added.

In the controller node, calculation to find $v(k)$ is performed. That is, the controller node is implemented by a computer that can solve an optimization problem to find $v(k)$. In other words, it may be implemented by a device that is different to agents. In the controller node, the initial states of all agents are collected, and are used in calculation of $v(k)$.

The following two assumptions are made for the controller node and communications as follows.

Assumption 1: The controller node has a self-loop, and is adjacent to only pinning nodes.

Assumption 2: In communications to collect the initial states, there exist a communication delay, that is, the message that is transmitted from some node at time k reaches to the other adjacent node at time $k + 1$.

We remark that the self-loop in the controller node is

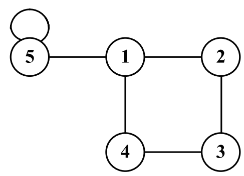

Fig. 1 Example of undirected graphs. The pinning node is 1, and the controller node is 5.

a dummy edge, and does not imply data transmission. In calculation of $u_i(k)$, we use $x_j(k)$, $j \in \mathcal{N}_i$ (see (5) and (6)). We suppose that in communications to collect the initial states, a different communication rule is applied. Hence, Assumption 2 is imposed. Based on the method in [18], in the controller node, we suppose that P and B in (7) are given in advance.

Next, we present a simple example of predictive pinning control. Consider the undirected graph in Fig. 1 again. Since there exists an input delay, (7) is modified to $x(k+1) = Px(k)+Bv(k-1)$. In MPC, the control input is generated by solving the finite-time optimal control problem at each time. We suppose that the finite-time optimal control problem is solved in the controller node. We also suppose that the initial states $x_i(0), i = 1,2,3,4$ are unknown in the controller node, and the information about $x_i(0)$ is sent from each node to the controller node through a given graph. After the controller node receives the information about $x_i(0)$, we can use $x_i(0)$ in the controller node. Hence, we must use the estimated initial state until the controller node receives all initial states. Let $\hat{x}_i(k)$ denote the estimated state at time k. Define $\hat{x}(k) := [\hat{x}_1(k)\ \hat{x}_2(k)\ \hat{x}_3(k)\ \hat{x}_4(k)]^\top$.

The proposed procedure of MPC is summarized. Assume that $v(-1)$ is preset. At time $k = 0$, the finite-time optimal control problem is solved using $\hat{x}(0)$. Let $v^*(0), v^*(1), \ldots, v^*(N-1)$ denote the obtained time sequence of $v(k)$, where N is the prediction horizon. Only $v^*(0)$ is sent to the pinning nodes.

At time $k = 1$, $v^*(0)$ is applied to the pinning nodes. The controller node receives the information about $x_1(0)$. In a similar way, the node 1 is received the information about $x_2(0)$ and $x_4(0)$. The nodes 2 and 4 are received the information about $x_3(0)$. Then, the estimated state $\hat{x}(1)$ is calculated by

$$\hat{x}(1) = P\begin{bmatrix} x_1(0) \\ \hat{x}_2(0) \\ \hat{x}_3(0) \\ \hat{x}_4(0) \end{bmatrix} + Bv(-1).$$

By regarding $\hat{x}(1)$ as the initial state, we can solve the finite-time optimal control problem. Let $v^{**}(1), v^{**}(2), \ldots, v^{**}(N)$ denote the obtained time sequence of $v(k)$. Only $v^{**}(1)$ is sent to the pinning nodes.

At time $k = 2$, $v^{**}(1)$ is applied to the pinning nodes. The controller node receives the information about $x_2(0)$ and $x_4(0)$. The node 1 is received the information about $x_3(0)$. Then, the estimated state $\hat{x}(2)$ is calculated by

$$\hat{x}(2) = P^2\begin{bmatrix} x_1(0) \\ x_2(0) \\ \hat{x}_3(0) \\ x_4(0) \end{bmatrix} + Bv^*(0) + PBv(-1).$$

We can solve the finite-time optimal control problem with $\hat{x}(2)$ as the initial state. Let $v^{***}(2), v^{***}(3), \ldots, v^{***}(N+1)$ denote the obtained time sequence of $v(k)$. Only $v^{***}(2)$ is sent to the pinning nodes.

Finally, at time $k = 3$, $v^{***}(2)$ is applied to the pinning nodes. The controller node receives the information about $x_3(0)$. Then, the estimated state $\hat{x}(3)$ is calculated by

$$\hat{x}(3) = P^3\begin{bmatrix} x_1(0) \\ x_2(0) \\ x_3(0) \\ x_4(0) \end{bmatrix} + Bv^{**}(1) + PBv^*(0) + P^2Bv(-1).$$

Note that the estimated initial state $\hat{x}_i(0)$ is not used. That is, $\hat{x}(3)$ is a correct value. After $k = 3$, the state $\hat{x}(k)$ can be calculated in a similar way.

3.2 Finite-Time Optimal Control Problem

We formulate the finite-time optimal control problem solved in MPC. Before that, we define $\tilde{x}(k) := \hat{x}(k) - x_d$, where $\hat{x}(k) \in \mathcal{R}^n$ is the estimated state, and $x_d \in \mathcal{R}^n$ is the target state given in advance. Noting that Assumption 2 is imposed, the problem is given as follows.

Problem 2:

$$\begin{aligned} \text{given}\quad & \hat{x}(t) = \hat{x}_t \text{ (current state)},\ \ v(t-1) = v_{t-1} \\ \text{find}\quad & v(t), v(t+1), \ldots, v(t+N-1) \\ \text{minimize}\quad & J = \sum_{k=t}^{t+N-1} \left\{ \tilde{x}^\top(k)Q\tilde{x}(k) + v^\top(k)Rv(k) \right\} \\ & \quad + \tilde{x}^\top(t+N)Q_f\tilde{x}(t+N) \qquad (8) \\ \text{subject to}\quad & \hat{x}(k+1) = P\hat{x}(k) + Bv(k-1), \qquad (9) \\ & v_{\min} \le v(k) \le v_{\max}. \qquad (10) \end{aligned}$$

In this problem, $Q \ge 0$, $R > 0$, and $Q_f \ge 0$ are given weighting matrices. We impose the input constraint ($v_{\min}, v_{\max} \in \mathcal{R}^m$ are given lower and upper bounds, respectively).

Consider rewriting Problem 2 into a QP problem. For simplicity of discussion, we explain the case of $x_d = 0$. It is easy to extend it to the case of $x_d \neq 0$. First, from (9), we can obtain

$$\hat{x}(t+k) = P^k\hat{x}_t + \sum_{i=1}^{k} P^{i-1}Bv(t+k-i-1).$$

From this expression, we can obtain

$$\bar{x} = \bar{P}\hat{x}_t + \bar{B}\bar{v}, \qquad (11)$$

where

$$\bar{x} = [\hat{x}^\top(t)\ \hat{x}^\top(t+1)\ \cdots\ \hat{x}^\top(t+N)]^\top,$$
$$\bar{v} = [v^\top(t-1)\ v^\top(t)\ \cdots\ v^\top(t+N-2)]^\top,$$
$$\bar{P} = \begin{bmatrix} I \\ P \\ P^2 \\ \vdots \\ P^N \end{bmatrix},\ \bar{B} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ PB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ P^{N-1}B & \cdots & PB & B \end{bmatrix}.$$

The cost function (8) can be rewritten as

$$J = \bar{x}^\top \bar{Q}\bar{x} + \bar{v}^\top \bar{R}\bar{v}, \tag{12}$$

where $\bar{Q} = $ block-diag$(Q, Q, \ldots, Q, Q_f)$ and $\bar{R} = $ block -diag$(R, R, \ldots, R)$. By substituting (11) into (12), we can obtain

$$J = \bar{v}^\top(\bar{R} + \bar{B}^\top\bar{Q}\bar{B})\bar{v} + 2\hat{x}_t\bar{P}^\top\bar{Q}\bar{B}\bar{v} + \hat{x}_t^\top\bar{P}^\top\bar{Q}\bar{P}\hat{x}_t. \tag{13}$$

Finally, the input constraint (10) can be rewritten as

$$\begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix}\bar{v} = v_{t-1}, \tag{14}$$
$$\bar{v}_{\min} \le \bar{v} \le \bar{v}_{\max}, \tag{15}$$

where $\bar{v}_{\min(\max)} = [v_{\min(\max)}^\top\ v_{\min(\max)}^\top\ \cdots\ v_{\min(\max)}^\top]^\top$. Thus, Problem 2 can be rewritten as the following QP problem:

$$\begin{aligned} &\text{find} && \bar{v} \\ &\text{minimize} && \text{the cost function (13)} \\ &\text{subject to} && \text{(14) and (15).} \end{aligned}$$

This problem can be solved by using a suitable solver such as IBM ILOG CPLEX Optimizer and Gurobi Optimizer. We remark that in this QP problem (i.e., Problem 2), the number of the decision variable $\bar{v}$ is mN. Hence, the computation time for solving Problem 2 does not depends on n (the number of nodes). However, since the coefficient matrix/vector in the cost function (13) are given using the current estimated state $\hat{x}_t$, for a large n, the processing time for calculating these matrix/vector may become large.

3.3 Proposed On-line Algorithm

An on-line algorithm based on MPC is proposed. Here, we assume that the number of the controller nodes is 1.

First, we add the controller node as the $(n+1)$-th node in the graph G. Then, the enlarged adjacency matrix $\bar{A}$ is given by

$$\bar{A} = \begin{bmatrix} A & \begin{bmatrix} 1_{m\times 1} \\ 0_{(n-m)\times 1} \end{bmatrix} \\ \begin{bmatrix} 1_{1\times m} & 0_{1\times(n-m)} \end{bmatrix} & 1 \end{bmatrix}.$$

Then, let $z(k) \in \mathcal{R}^n$ denote the vector obtained by excluding the $(n+1)$-th element from the $(n+1)$-th column of $\bar{A}^k$. If the i-th element of $z(k)$ is not zero, then the controller node can receive the information about $x_i(0)$ from the node i. In addition, let $\tilde{z}(k) \in \{0,1\}^n$ denote the vector obtained by replacing non-zero elements of $z(k)$ with '1'. Using $\tilde{z}(k)$, the initial estimated state $\hat{x}(0)$ in the controller node can be updated by

$$\hat{x}(0) = \mathrm{diag}(\tilde{z}(k))x(0) + \mathrm{diag}(1_{n\times 1} - \tilde{z}(k))\hat{x}_0, \tag{16}$$

where $\hat{x}_0$ is given in advance. If the controller node receives $x_i(0)$, then the i-th element of $\hat{x}_0$ is replaced with $x_i(0)$. Furthermore, the current estimated state $\hat{x}(t)$ can be estimated by using the control input sequence applied to the pinning nodes.

Next, based on the above preparation, we propose an on-line algorithm.

On-line Algorithm for Predictive Pinning Control:

Step 1: Set $t = 0$, $\hat{x}_0$, and $v(-1)$.

Step 2: Solve Problem 2 (i.e., the QP problem).

Step 3: Send only $v(t)$ obtained by Step 2 to the pinning nodes.

Step 4: Apply only $v(t-1)$ to the pinning nodes.

Step 5: The controller node receives the information about the initial state through pinning nodes.

Step 6: Update the initial estimated state by using (16).

Step 7: Calculate the current estimated state by using the control input applied to the pinning nodes.

Step 8: Update $t := t + 1$, and return to Step 2.

In this algorithm, we consider communication delays. That is, since the current state cannot be directly estimated, the initial state is firstly estimated. After that, the current state is estimated by using the control input sequence. Hence, simple delay compensation is embedded in the proposed algorithm.

Finally, we discuss stability of the closed-loop system. Stability (i.e., the convergence to the target state) is guaranteed by imposing the terminal equality constraint $\hat{x}(t+N) = 0$ in Problem 2 or terminal inequality constraints. See, e.g., [3], [10], [11] for further details. However, a longer prediction horizon is generally required, and the computation time for solving Problem 2 becomes longer.

Remark 1: In the proposed algorithm, the controller node must aggregate the data on $x_i(0)$ for each node. Since the graph G is fixed in this paper, we can utilize multi-hop communication. In multi-hop communication, transmissions between nodes are achieved through multi-hop paths (see, e.g., [6]). By giving a certain routing protocol in advance for each node, the controller node can aggregate $x_i(0)$. Multi-hop communication is utilized in also the consensus problem (see, e.g., [12]).

4. Numerical Examples

Consider a multi-agent system with fifty agents. The graph expressing communication links is given by Fig. 2. This graph was generated by using Barabási-Albert model [1], [2]. In this graph, the number of nodes is 51, and one controller

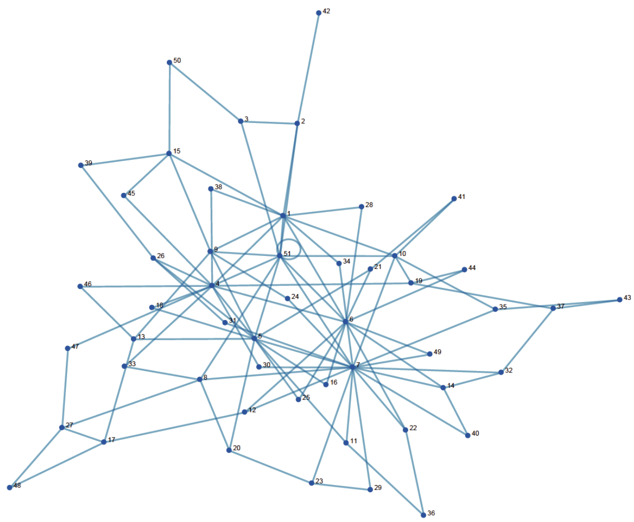

Fig. 2 Undirected graph with 50 nodes and the controller node.

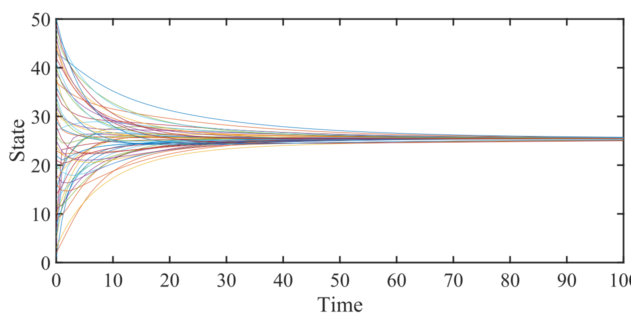

Fig. 3 Time response of the state with no pinning control.

node (the node 51) is included. The number of the pinning nodes is 10 (i.e., the nodes $1, 2, \ldots, 10$ are the pinning node). The parameter ε in the matrix P is given by $\varepsilon = 0.95/d_{\max} = 0.95/17 = 0.0559$. The weighting matrices, Q, R, and Q_f are given by $Q = 100I$, $R = 1$, and $Q_f = Q$, respectively. In addition, we set $v_{\min} = -5 \times 1_{10\times 1}$ and $v_{\max} = 5 \times 1_{10\times 1}$. The initial state and the initial estimated state are given by $x(0) = [1\ 2\ \cdots\ 50]^\top$ and $\hat{x}_0 = 25.5 \times 1_{50\times 1}$, respectively. The initial control input is given by $u(-1) = 0_{10\times 1}$.

First, consider the case of $N = 10$ and $x_d = 25.5 \times 1_{50\times 1}$ (the average of initial states is 25.5). Figure 3 shows time response of the state with no pinning control (i.e., $v(k) = 0$). From this figure, we see that the state converges to the average of initial states i.e., x_d. We remark that in this case, the target state x_d is not given, but the state converges to x_d. Figure 4 shows time response of the state with pinning control. Comparing Fig. 4 with Fig. 3, we see that faster consensus is achieved. Figure 5 shows the external control input. In this case, the control input is saturated.

Next, consider the case of $N = 10$ and $x_d = 0_{50\times 1}$. Figure 6 shows time response of the state with pinning control. From this figure, we see that the state converges to x_d. Figure 7 shows the external control input. In this case, the control input is saturated. Here, based on the prediction horizon N, we discuss the con-

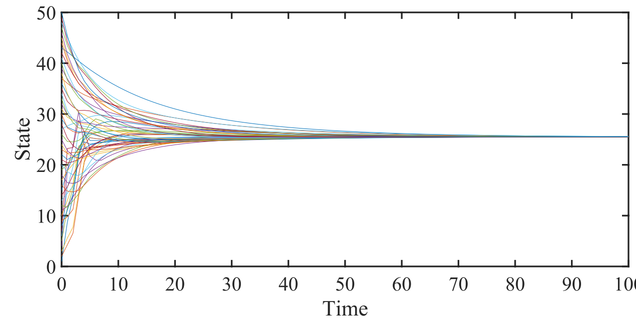

Fig. 4 Time response of the state with pinning control ($x_d = 25.5 \times 1_{50\times 1}$).

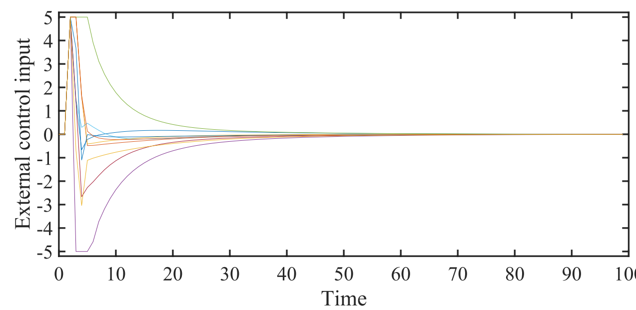

Fig. 5 External control input in pinning control ($x_d = 25.5 \times 1_{50\times 1}$).

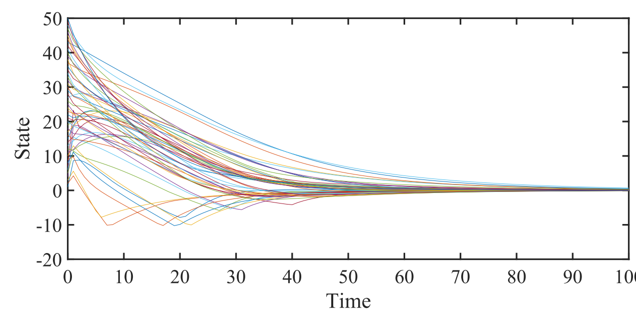

Fig. 6 Time response of the state with pinning control ($x_d = 0_{50\times 1}$).

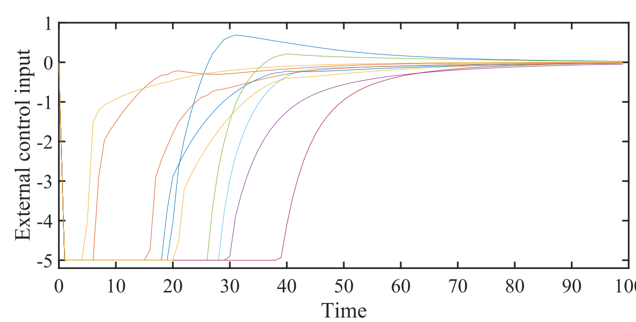

Fig. 7 External control input in pinning control ($x_d = 0_{50\times 1}$).

trol performance. As a performance index, we define $\tilde{J}(N) := \sum_{k=0}^{100} \tilde{x}^\top(k) Q \tilde{x}(k) + \sum_{k=0}^{99} v^\top(k) R v(k)$. By $\tilde{J}(N)$, we can evaluate the convergence speed and so on. For $N = 2, 5, 10, 20, 30$, we can obtain

$$\tilde{J}(2) = 5.2221 \times 10^7, \quad \tilde{J}(5) = 4.9966 \times 10^7,$$

$$\tilde{J}(10) = 4.9511 \times 10^7, \quad \tilde{J}(20) = 4.9429 \times 10^7,$$
$$\tilde{J}(30) = 4.9424 \times 10^7.$$

From these values, we see that a longer N achieves a better performance. However, for a longer N, the computation time for solving Problem 2 may become long. We must consider the trade-off between the control performance and the computation time.

Finally, we discuss the computation time for solving Problem 2. In the case of $N = 10$ and $x_d = 25.5 \times 1_{50\times1}$, the worst computation time was 0.0166 sec, and the mean computation time was 0.0118 sec, where the QP problems were solved by IBM ILOG CPLEX 12.7.1 on the computer with CPU: Intel Core i7-6700K 4.00GHz processor and Memory: 16GB. In the case of $N = 10$ and $x_d = 0_{50\times1}$, the worst computation time was 0.0155 sec, and the mean computation time was 0.0108 sec. In the case of $N = 30$ and $x_d = 0_{50\times1}$, the worst computation time was 0.3037 sec, and the mean computation time was 0.1046 sec. Thus, in this example, Problem 2 can be solved fast.

5. Conclusion

In this paper, we studied predictive pinning control with communication delays. Based on the policy of MPC, the on-line algorithm for estimating the current state and finding the external control input was proposed. The effectiveness of the proposed method is presented by a numerical example. The main result in this paper provides us one of the fundamentals in predictive pinning control.

There are several open problems. First, in some cases, it is appropriate that only the states of the pinning nodes are aggregated. Then, it is important to develop a method of observer-based pinning control. Next, in this paper, we assumed that the number of the controller nodes is 1. It is important to develop a distributed on-line algorithm for multiple controller nodes. Third, in networked control, event-triggered and self-triggered control methods has been developed (see, e.g., [8]). It is also important to develop a new method combined these methods and predictive pinning control. Finally, it is also significant to consider uncertainties such as disturbances and switching networks.

The author would like to thank Mr. Shin Kanazawa, Hokkaido University for fruitful discussions.

This work was partly supported by the Telecommunications Advancement Foundation and JSPS KAKENHI Grant Number 17K06486.

References

[1] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," Rev. Mod. Phys., vol.74, no.1, pp.47–97, 2002.
[2] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," Science, vol.286, no.5439, pp.509–512, 1999.
[3] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," Automatica, vol.35, no.3, pp.407–427, 1999.
[4] E.F. Camacho and C.B. Alba, Model Predictive Control, Second Edition, Springer, 2007.
[5] G. Ferrari-Trecate, L. Galbusera, M.P.E. Marciandi, and R. Scattolini, "Model predictive control schemes for consensus in multi-agent systems with single- and double-integrator dynamics," IEEE Trans. Autom. Control, vol.54, no.11, pp.2560–2572, 2009.
[6] K. Govindan, D. Chander, B.G. Jagyasi, S.N. Merchant, and U.B. Desai, Multihop Mobile Wireless Networks, River Publishers, 2010.
[7] K. Hamada, N. Hayashi, and S. Takai, "Event-triggered and self-triggered control for discrete-time average consensus problems," SICE Journal of Control, Measurement, and System Integration, vol.7, no.5, pp.297–303, 2014.
[8] W.P.M.H. Heemels, K.H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," Proc. 51st IEEE Conf. on Decision and Control, pp.3270–3285, 2012.
[9] S. Iwase, N. Hayashi, and S. Takai, "A gradient-based approach for discrete-time average consensus with self-triggered control," SICE Journal of Control, Measurement, and System Integration, vol.9, no.3, pp.122–127, 2016.
[10] M. Lazar, W.P.M.H. Heemels, S. Weiland, and A. Bemporad, "Stabilizing model predictive control of hybrid systems," IEEE Trans. Autom. Control, vol.51, no.11, pp.1813–1818, 2006.
[11] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert, "Constrained model predictive control: Stability and optimality," Automatica, vol.36, no.6, pp.789–814, 2000.
[12] S. Miyake, N. Hayashi, and S. Takai, "Discrete-time average consensus with multi-hop communication," SICE Journal of Control, Measurement, and System Integration, vol.9, no.5, pp.187–191, 2016.
[13] R. Olfati-Saber and R.M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," IEEE Trans. Autom. Control, vol.49, no.9, pp.1520–1533, 2004.
[14] W. Ren and R.W. Beard, "Consensus seeking in multiagents systems under dynamically changing interaction topologies," IEEE Trans. Autom. Control, vol.50, no.5, pp.655–661, 2005.
[15] W. Ren, R.W. Beard, and E.M. Arkins, "Information consensus in multivehicle cooperative control," IEEE Control Syst. Mag., vol.71, no.2, pp.71–82, 2007.
[16] H. Su and X. Wang, Pinning Control of Complex Networked Systems: Synchronization, Consensus and Flocking of Networked Systems via Pinning, Springer, 2013.
[17] X. Wang and H. Su, "Pinning control of complex networked systems: A decade after and beyond," Annual Reviews in Control, vol.38, no.1, pp.103–111, 2014.
[18] H.-T. Zhang, M.Z.Q. Chen, and G.-B. Stan, "Fast consensus via predictive pinning control," IEEE Trans. Circuits Syst. I, Reg. Papers, vol.58, no.9, pp.2247–2258, 2011.

Koichi Kobayashi received the B.E. degree in 1998 and the M.E. degree in 2000 from Hosei University, and the D.E. degree in 2007 from Tokyo Institute of Technology. From 2000 to 2004, he worked at Nippon Steel Corporation. From 2007 to 2015, he was an Assistant Professor at Japan Advanced Institute of Science and Technology. Since 2015, he has been an Associate Professor at the Graduate School of Information Science and Technology, Hokkaido University. His research interests include analysis and control of discrete event and hybrid systems. He is a member of IEEE, IEEJ, IEICE, ISCIE, and SICE.