PAPER Special Section on Formal Approach Symbolic Design of Networked Control Systems with State Prediction

Masashi MIZOGUCHI[†], Member and Toshimitsu USHIO^{†a)}, Fellow

SUMMARY In this paper, we consider a networked control system where bounded network delays and packet dropouts exist in the network. The physical plant is abstracted by a transition system whose states are quantized states of the plant measured by a sensor, and a control specification for the abstracted plant is given by a transition system when no network disturbance occurs. Then, we design a prediction-based controller that determines a control input by predicting a set of all feasible abstracted states at time when the actuator receives the delayed input. It is proved that the prediction-based controller suppresses effects of network delays and packet dropouts and that the controlled plant still achieves the specification in spite of the existence of network delays and packet dropouts.

key words: networked control systems, symbolic control, approximate similarity relations, state prediction

1. Introduction

Symbolic controller design methods have been studied in the last decade [1]. For the design of a symbolic controller, we abstract a physical plant with a finite-state transition system whose behavior is an approximation of that of the physical plant. In the abstraction, the notion of (bi)simulation plays a key role. The bisimulation was introduced to show the equivalence of behaviors of two concurrent systems [2]. It is applied to abstract a system with the (infinite) state set to a model with the finite one. In general, however, it is very restrictive to abstract a continuous state set to a finite one with the (bi)simulation. Then, approximate (bi)simulation was proposed so that we can determine if two given systems exhibit similar behaviors within a specified precision [3], [4]. The stability of the controlled system with the approximate simulation is investigated with a Lyapunov-like function [5]. The approximate (bi)simulation-based abstraction has been studied for nonlinear systems [6], [7], switched linear systems [8], and time-delay systems [9].

In the control systems, there exist disturbances that affect behaviors of the controlled plant. Such disturbed behaviors can be modeled by nondeterministic transitions in the plant model. The approximate alternating (bi)simulation, which is an approximated version of the alternating (bi)simulation [10], is used to design a symbolic model for an incrementally globally asymptotically stable nonlinear control system with disturbances [11]. The ex-

[†]The authors are with the Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560–8531 Japan.

DOI: 10.1587/transinf.2016FOP0001

istence of a symbolic model for a time-varying time-delay system has been discussed in [12]. Parameterized approximate contractive alternating simulation was introduced as a key notion for the design of a robust symbolic state feedback controller under disturbances such as packet dropouts, where a desired behavior is specified for the abstracted model of the physical plant [13]. It is shown that the controlled plant by the robust controller is input-to-state dynamically stable. In addition, a symbolic output feedback controller is designed with a symbolic observer that estimates a current abstracted state [14].

In the last few years, a symbolic control approach was extended to a networked control system (NCS). A symbolic model of a nonlinear physical plant is derived when the plant is connected to the network with time-varying bounded network delays and packet dropouts, and a symbolic state feedback controller was constructed with a Lyapunov function [15]. The method to construct an abstracted model of the NCS from that of the physical plant was proposed [16]. These studies consider the robust controller design, that is, a network delay is dealt with a perturbation and the symbolic controllers are robust against it. In digital control, utilization of state prediction is often used as another approach to design a digital controller when a computational delay exists [17]. To the best of our knowledge, however, the design of a prediction-based symbolic controller for the networked control has not been considered.

In this paper, we assume that abstracted desired behaviors of a physical plant are given by a transition system, and that there exists a symbolic controller for a control specification such that there exists an approximate contractive alternating simulation relation (acASR) from the symbolic controller to the plant if there is no delay nor packet dropout between the controller and the plant. We consider NCSs with bounded network delays and packet dropouts. We propose a model of the unreliable communication network by introducing a transition system whose states are queues of inputs. Then, we design a prediction-based symbolic controller that predicts a set of feasible abstracted states of the plant at time when the actuator receives the delayed input. The controller determines a control input in such a way that every predicted state satisfies the control specification. Then, it is shown that there exists an acASR from the prediction-based controller to the networked plant, which means that the controlled plant exhibits a desired behavior and that the specification is still achieved in spite of the network disturbances.

The rest of the paper is organized as follows. In Sect. 2,

Manuscript received August 8, 2016.

Manuscript revised December 21, 2016.

Manuscript publicized March 7, 2017.

a) E-mail: ushio@sys.es.osaka-u.ac.jp

we review the existence of the symbolic controller in the case where there is no network delay nor packet dropout. In Sect. 3, we introduce a model of an unreliable network, and construct a transition system that predicts the state of the plant. The digital controller is designed in Sect. 4, and an illustrative example is shown in Sect. 5.

2. Preliminaries

2.1 Notations

Let \mathbb{Z} , \mathbb{R} , $\mathbb{Z}_{\geq 0}$, and $\mathbb{R}_{\geq 0}$ be the sets of integers, real numbers, non-negative integers, and non-negative real numbers, respectively. The ∞ -norm of $x \in \mathbb{R}^n$ is denoted by |x|. For a given set $A \subseteq \mathbb{R}^n$, we denote a uniform grid in A by $[A]_{\eta} := \{x \in A \mid \exists k \in \mathbb{Z}^n : x = 2k\eta\}$ where $\eta \in \mathbb{R}_{\geq 0} \setminus \{0\}$ is an abstraction parameter of the grid.

2.2 Transition Systems and Simulation Relations

A transition system is described by a tuple (X, X_0, U, r) where X is a set of states, $X_0 \subseteq X$ is a set of initial states, U is a set of inputs, and $r: X \times U \rightarrow 2^X$ is a transition map. For any $x \in X$, let U(x) be a set of inputs defined by $U(x) := \{u \in U \mid r(x, u) \neq \emptyset\}$. U^* denotes the set of all finite input sequences over U, and we always have $\varepsilon \in U^*$ where ε is the empty sequence. For any $t \in U^*$, last(t) is the last input of t, where $last(\varepsilon) = \varepsilon$. We extend the transition map $r: X \times U \to 2^X$ to $r: X \times U^* \to 2^X$ in the natural way. Moreover, a transition system with an output is denoted by (X, X_0, U, r, Y, H) , where (X, X_0, U, r) is a transition system, Y is a set of outputs, and $H : X \to Y$ is an output map. Let $S_1 = (X_1, X_{10}, U_1, r_1)$ and $S_2 = (X_2, X_{20}, U_2, r_2)$ be two transition systems. For a relation $R \subseteq X_1 \times X_2 \times U_1 \times U_2$ over the state sets X_1, X_2 and the input sets U_1, U_2 , denoted by $R_X \subseteq X_1 \times X_2$ is a projection of R to the state sets X_1, X_2 defined as follows:

$$R_X = \{ (x_1, x_2) \in X_1 \times X_2 \mid \\ \exists u_1 \in U_1, \exists u_2 \in U_2 : (x_1, x_2, u_1, u_2) \in R \}.$$

Then, we review key notions for transition systems [13].

Definition 1: Let $S_1 = (X_1, X_{10}, U_1, r_1)$ and $S_2 = (X_2, X_{20}, U_2, r_2)$ be two systems, let $\kappa, \lambda \in \mathbb{R}_{\geq 0}, \beta \in [0, 1]$ be some parameters, and consider a map $d : U_1 \times U_2 \to \mathbb{R}_{\geq 0}$. We call a parameterized (by $\epsilon \in [\kappa, \infty[)$ relation $R(\epsilon) \subseteq X_1 \times X_2 \times U_1 \times U_2$ a κ -approximate (β, λ) -contractive alternating simulation relation $((\kappa, \beta, \lambda)$ -acASR) from S_1 to S_2 with d if $R(\epsilon) \subseteq R(\epsilon')$ holds for all $\epsilon \leq \epsilon'$ and the following two conditions hold for all $\epsilon \in [\kappa, \infty[$:

- (1) For any $x_{10} \in X_{10}$, there exists $x_{20} \in X_{20}$ such that $(x_{10}, x_{20}) \in R_X(\kappa)$.
- (2) For any $(x_1, x_2) \in R_X(\epsilon)$ and any $u_1 \in U_1(x_1)$, there exists $u_2 \in U_2(x_2)$ satisfying the following conditions:

a.
$$(x_1, x_2, u_1, u_2) \in R(\epsilon)$$
; and

b.
$$\forall x'_2 \in r_2(x_2, u_2), \exists x'_1 \in r_1(x_1, u_1) : (x'_1, x'_2) \in R_X(\kappa + \beta \epsilon + \lambda \mathbf{d}(u_1, u_2)).$$

We call $R(\epsilon)$ an alternating simulation relation (ASR) from S_1 to S_2 if $R(\epsilon)$ is a (0, 0, 0)-acASR from S_1 to S_2 .

Definition 2: Let $S_1 = (X_1, X_{10}, U_1, r_1)$ and $S_2 = (X_2, X_{20}, U_2, r_2)$ be two systems, and let $R \subseteq X_1 \times X_2 \times U_1 \times U_2$ be a relation. We define the composition of S_1 and S_2 with respect to R, denoted by $S := S_1 \times_R S_2 = (X, X_0, U, r)$ where

- $X = X_1 \times X_2;$
- $X_0 = (X_{10} \times X_{20}) \cap R_X;$
- $U = U_1 \times U_2$; and
- $r: X \times U \rightarrow 2^X$ is defined as follows: $(x'_1, x'_2) \in r((x_1, x_2), (u_1, u_2))$ if and only if

$$\begin{aligned} x_1' &\in r_1(x_1, u_1) \ \land \ x_2' &\in r_2(x_2, u_2) \\ &\land \ (x_1, x_2, u_1, u_2) \in R \ \land \ (x_1', x_2') \in R_X. \end{aligned}$$

If $R(\epsilon)$ is a (κ, β, λ) -acASR from S_1 to S_2 with d, we replace the above definitions of X_0 and r with the following conditions:

- $X_0 = (X_{10} \times X_{20}) \cap R_X(\kappa)$; and
- $r: X \times U \rightarrow 2^X$ is defined as follows: $(x'_1, x'_2) \in r((x_1, x_2), (u_1, u_2))$ if and only if

$$\begin{aligned} x'_1 &\in r_1(x_1, u_1) \land x'_2 \in r_2(x_2, u_2) \\ \land & (x_1, x_2, u_1, u_2) \in R(e(x_1, x_2)) \\ \land & (x'_1, x'_2) \in R_X(\kappa + \beta e(x_1, x_2) + \lambda \mathsf{d}(u_1, u_2)) \end{aligned}$$

where $e(x_1, x_2) := \inf \{ e \in \mathbb{R}_{\geq 0} \mid (x_1, x_2) \in R_X(e) \}.$

2.3 Symbolic Synthesis of a Controller

We review a symbolic design of a controller proposed in [13]. A physical plant to be controlled, denoted by S = (X, X_0, U, r) , is a discrete-time system, and its state set X is a Euclidean space. Thus, S is an infinite transition system. In order to design a digital controller, a finite abstracted model of the plant, denoted by $\hat{S} = (\hat{X}, \hat{X}_0, \hat{U}, \hat{r})$, is introduced. Intuitively, \hat{X} is the set of the plant states measured by a sensor, and \hat{S} is implemented in the cyber space. The abstraction and refinement relationship between \hat{S} and S is described by the existence of an acASR $R(\epsilon) \subseteq \hat{X} \times X \times \hat{U} \times U$ from \hat{S} to S. Consider the case where the desired behavior of \hat{S} is already given by a finite transition system $\hat{S}_C = (\hat{X}_C, \hat{X}_{C0}, \hat{U}_C, \hat{r}_C)$. Intuitively, \hat{S}_C is a feedback controller for \hat{S} , and the feedback relationship between \hat{S}_C and \hat{S} is described by the existence of an ASR $\hat{R}_C \subseteq \hat{X}_C \times \hat{X} \times \hat{U}_C \times \hat{U}$ from \hat{S}_C to \hat{S} . Then, the following theorem shows the existence of a symbolic feedback controller for the physical plant *S* [13].

Theorem 1: Let $\hat{S}_C = (\hat{X}_C, \hat{X}_{C0}, \hat{U}_C, \hat{r}_C), \hat{S} = (\hat{X}, \hat{X}_0, \hat{U}, \hat{r}),$ and $S = (X, X_0, U, r)$ be systems, let $\kappa, \lambda \in \mathbb{R}_{\geq 0}, \beta \in [0, 1]$ be some parameters, and consider a map $d : \hat{U} \times U \to \mathbb{R}_{\geq 0}.$ Assume that there exists an ASR $\hat{R}_C \subseteq \hat{X}_C \times \hat{X} \times \hat{U}_C \times \hat{U}$ from \hat{S}_C to \hat{S} and a (κ, β, λ) -acASR $R(\epsilon) \subseteq \hat{X} \times X \times \hat{U} \times U$ from \hat{S} to S with d. Then, the following relation $R_C(\epsilon) \subseteq$ $(\hat{X}_C \times \hat{X}) \times X \times (\hat{U}_C \times \hat{U}) \times U$ is a (κ, β, λ) -acASR from $S_C := \hat{S}_C \times_{\hat{R}_C} \hat{S}$ to S with $\mathsf{d}_C((\hat{u}_C, \hat{u}), u) = \mathsf{d}(\hat{u}, u)$:

$$R_C(\epsilon) = \{ ((\hat{x}_C, \hat{x}), x, (\hat{u}_C, \hat{u}), u) \mid \\ (\hat{x}, x, \hat{u}, u) \in R(\epsilon) \land (\hat{x}_C, \hat{x}) \in \hat{R}_{CX} \}.$$

$$(1)$$

Theorem 1 shows that the composed system S_C = $\hat{S}_C \times_{\hat{R}_C} \hat{S}$ is a feedback controller of \hat{S} . When the current state of the physical plant S is x, the state \hat{x} of \hat{S} is determined by the relation $R(\epsilon)$. Then, the state \hat{x}_C of \hat{S}_C is given by the relation \hat{R}_{C} . The controller determines the control input \hat{u}_C such that $\hat{r}_C(\hat{x}_C, \hat{u}_C) \neq \emptyset$. The inputs \hat{u} of \hat{S} and u of S are also given by relations \hat{R}_C and $R(\epsilon)$, respectively.

However, the network delay is not considered in Theorem 1. It is common that the state (\hat{x}_C, \hat{x}) (resp. the input *u*) will be determined several time steps after the state x (resp. the input (\hat{u}_C, \hat{u}) is determined. Moreover, packet dropouts may occur in the networks, which is not considered, neither.

3. **Modeling Network Delays and Packet Dropouts**

In this paper, we extend Theorem 1 to design a controller for the networked plant shown in Fig. 1. In the following, we assume that all conditions in Theorem 1 are satisfied and that the networks satisfy the following conditions:

- The networks are cyber components, thus they transmit symbols;
- The networks are unreliable, and packet dropouts sometimes occur;
- Every data is time-stamped, which means that the data includes the time when it was sent to the network;
- Delay time is variable due to the amount of the other data on the network:
- The networks follow the First-In-First-Out basis;
- The worst delay time from the physical plant S to the controller S_C is given by $L^{sc} \in \mathbb{Z}_{\geq 0}$; and
- The worst delay time from the controller S_C to the physical plant S is given by $L^{ca} \in \mathbb{Z}_{>0}$.

Then, we model the networks by queues and introduce a special symbol \perp that means

- 1. the element of the queue is empty; or
- 2. a packet dropout occurs.

The sensor measures a state of the physical plant and



Fig. 1 Structure of the networked control system.

transforms it to its abstracted state, which is modeled by a map sen : $X \to \hat{X}$. The abstracted state is transmitted to the symbolic controller via the network. The symbolic controller determines the abstracted input and transmits it to the actuator via the network. The actuator updates the input of the physical plant based on the received data from the network. Based on these facts, for a design of a networked controller, we introduce a symbol $u^{\perp} \in U$ that denotes an input satisfying the following two conditions:

- For any $x \in X$, $u^{\perp} \in U(x)$; and
- The actuator updates the control input to u^{\perp} whenever it does not receive the data from the network due to the data dropout and/or the increase of the network delay.

Then, the actuator is modeled by a map act : $\hat{U} \cup \{\bot\} \rightarrow U$, where $act(\perp) = u^{\perp}$. In addition, we introduce a symbol $\hat{u}^{\perp} \in \hat{U}$ that denotes an input satisfying the following two conditions:

- For any $\hat{x} \in \hat{X}$, $\hat{u}^{\perp} \in \hat{U}(\hat{x})$; and
- $\operatorname{act}(\hat{u}^{\perp}) = u^{\perp}$.

The actuator determines $act(\perp) = u^{\perp}$ as an updated input when no input signal arrives. Then, we have two cases: delays and packet dropouts. Even though there are delays, the control input will be injected to the plant in the future. However, if the packet dropout occurs, the input is updated to u^{\perp} automatically. In this paper, we design a controller that computes reachable states of the plant from the measured state with the input history. Thus, \hat{u}^{\perp} is introduced for the controller to consider the case of the packet dropout.

In the following, we also assume that the (κ, β, λ) acASR $R(\epsilon)$ from \hat{S} to S satisfies the following conditions:

- For any $(\hat{x}, x, \hat{u}, u) \in R(\epsilon)$, $\hat{x} = \operatorname{sen}(x)$ and $u = \operatorname{act}(\hat{u})$;
- For any $\epsilon \in [\kappa, \infty[, (\hat{x}, x) \in R_X(\epsilon) \land \hat{u} \in \hat{U}(\hat{x}) \Rightarrow$ $(\hat{x}, x, \hat{u}, \operatorname{act}(\hat{u})) \in R(\epsilon)$; and
- There exists $d \in \mathbb{R}_{>0}$ such that

$$\forall \epsilon \in [\kappa, \infty[, \forall (\hat{x}, x, \hat{u}, u) \in \hat{X} \times X \times \hat{U} \times U : \\ (\hat{x}, x, \hat{u}, u) \in R(\epsilon) \Rightarrow \mathsf{d}(\hat{u}, u) \le d.$$

Then, we model the network from the plant to the controller. The model depends on a network protocol, and in this paper, for simplicity, we consider a protocol such that the packet dropouts do not occur consecutively.

Definition 3: Let $L^{sc} \in \mathbb{Z}_{\geq 0}$ be the worst delay time. Then, an unreliable communication network from the physical plant S to the symbolic digital controller S_C , denoted by S_{O} , is defined by the following transition system with an output:

$$S_Q = (X_Q, X_{Q0}, U_Q, r_Q, Y_Q, H_Q),$$
(2)

where

- $\begin{array}{l} \bullet \ X_{Q} \ = \ \{(\hat{x}^{0}, \hat{x}^{1}, \hat{x}^{2}, \ldots, \hat{x}^{L^{sc}}, l) \ | \ \forall i \ \in \ \{0, 1, 2, \ldots, L^{sc}\} \ : \\ \hat{x}^{i} \ \in \ \hat{X} \cup \{ \bot \}, l \ \in \ \{1, 2, \ldots, L^{sc} + 1\} \}; \\ \bullet \ X_{Q0} \ = \ \{(\hat{x}^{0}, \bot, \ldots, \bot, L^{sc} + 1) \ \in \ X_{Q} \ | \ \hat{x}^{0} \ \in \ \hat{X}_{0} \cup \{ \bot \} \}; \end{array}$

• $U_Q = X;$

•
$$r_Q: X_Q \times U_Q \to 2^{X_Q}$$
 is defined as follows:

$$\begin{split} r_{Q}((\hat{x}^{0}, \hat{x}^{1}, \hat{x}^{2}, \dots, \hat{x}^{L^{sc}}, l), x') \\ &= \begin{cases} \{(\hat{x}^{*}, \hat{x}^{0}, \hat{x}^{1}, \dots, \hat{x}^{L^{sc}-l}, \bot, \dots, \bot, l') \mid \\ \hat{x}^{*} \in \{\text{sen}(x'), \bot\}, l' \in \{1, 2, \dots, L^{sc} + 1\} \} \\ \{(\text{sen}(x'), \hat{x}^{0}, \hat{x}^{1}, \dots, \hat{x}^{L^{sc}-l}, \bot, \dots, \bot, l') \mid \\ l' \in \{1, 2, \dots, L^{sc} + 1\} \} \\ \text{if } \hat{x}^{0} = \bot; \end{cases} \end{split}$$

- Y_Q is a sequence of the symbolic states whose length is at most $L^{sc} + 1$; and
- $H_Q: X_Q \to Y_Q$ is defined as follows:

$$H_{Q}((\hat{x}^{0}, \hat{x}^{1}, \dots, \hat{x}^{L^{sc}}, l)) \\ = \hat{x}^{L^{sc}-l+1} \hat{x}^{L^{sc}-l+2} \dots \hat{x}^{L^{sc}}$$

The network is modeled by an FIFO queue as shown in Fig. 2. When the time is elapsed, data is enqueued at the top, and the other elements are shifted to their next ones. The last element of $x_Q \in X_Q$, denoted by $l \in$ $\{1, 2, ..., L^{sc} + 1\}$, indicates the number of data dequeued at the time. Then, the output is the oldest l data sequence $\hat{x}^{L^{sc}-l+1}\hat{x}^{L^{sc}-l+2}\dots\hat{x}^{L^{sc}}$ stored in the queue. Since the controller designates an initial state \hat{x}_0 , l is fixed to $L^{sc} + 1$ at the initial state. Packet dropouts are modeled by the nondeterministic transitions. If it occurs, the enqueued data is set to \bot . The transition map r_Q is defined considering two cases: $\hat{x}^0 \neq \bot$ and $\hat{x}^0 = \bot$ because the dropouts do not occur consecutively \dagger .

On the other hand, the control inputs are sent from S_C to S via the same network. However, in this case, we consider the worst delay time L^{ca} only. When the delay time of a control input is less than L^{ca} , the data is stored in an input queue at the actuator, and injected to the plant accurately L^{ca} steps after the controller transmits the abstracted input. This is because the controller determines a control input by considering the worst case, which means that the input is de-



Fig. 2 S_Q modeling the unreliable network from S to S_C .

[†]It is noted that, if the worst delay time is known and the packet dropout depending on the network protocol is modeled by a nondeterministic transition system, then the proposed controller can be constructed. termined for states after L^{ca} time steps are elapsed. An unreliable network with the fixed delay time and packet dropouts is modeled by the following definition that is a special case of Definition 3 (*l* is fixed to 1, thus omitted).

Definition 4: Let $L^{ca} \in \mathbb{Z}_{\geq 0}$ be the worst delay time. Then, an unreliable communication network from S_C to S, denoted by $S_{L^{ca}}$, is defined by the following transition system with an output:

$$S_{L^{ca}} = (X_{L^{ca}}, X_{L^{ca}}, U_{L^{ca}}, r_{L^{ca}}, Y_{L^{ca}}, H_{L^{ca}}),$$
(3)

where

•
$$X_{L^{ca}} = \{(\hat{u}^0, \hat{u}^1, \hat{u}^2, \dots, \hat{u}^{L^{ca}}) \mid \forall i \in \{0, 1, 2, \dots, L^{ca}\}$$

 $\hat{u}^i \in \hat{U} \cup \{\bot\}\};$

•
$$X_{L^{ca}0} = \{(\bot, \ldots, \bot) \in X_{L^{ca}}\}$$

- $U_{L^{ca}} = \hat{U};$
- $r_L: X_{L^{ca}} \times U_{L^{ca}} \to 2^{X_{L^{ca}}}$ is defined as follows:

$$\begin{split} r_{L^{ca}}((\hat{u}^{0}, \hat{u}^{1}, \hat{u}^{2}, \dots, \hat{u}^{L^{ca}}), \hat{u}') \\ &= \begin{cases} \{(\hat{u}^{*}, \hat{u}^{0}, \hat{u}^{1}, \dots, \hat{u}^{L^{ca}-1}) \mid \hat{u}^{*} \in \{\hat{u}', \bot\}\} \\ & \text{if } \hat{u}^{0} \neq \bot, \\ \{(\hat{u}', \hat{u}^{0}, \hat{u}^{1}, \dots, \hat{u}^{L^{ca}-1})\} & \text{if } \hat{u}^{0} = \bot; \end{cases} \end{split}$$

- $Y_{L^{ca}} = \hat{U} \cup \{\bot\};$ and
- $H_{L^{ca}}: X_{L^{ca}} \to Y_{L^{ca}}$ is defined as follows:

$$H_{L^{ca}}((\hat{u}^{0}, \hat{u}^{1}, \dots, \hat{u}^{L^{ca}})) = \operatorname{act}(\hat{u}^{L^{ca}})$$

Then, the following composed system S_D models the networked plant:

$$S_D = (X_D, X_{D0}, U_D, r_D) := S_{L^{ca}} \times_{R^u} S \times_{R^x} S_Q, \qquad (4)$$

where the relations $R^u \subseteq X_{L^{ca}} \times X \times U_{L^{ca}} \times U$ and $R^x \subseteq X \times X_Q \times U \times U_Q$ are given as follows:

$$R^{u} = \{(x_{L^{ca}}, x, u_{L^{ca}}, u) \in X_{L^{ca}} \times X \times U_{L^{ca}} \times U \mid u = H_{L^{ca}}(x_{L^{ca}})\}, \quad (5)$$
$$R^{x} = \{(x, x_{Q}, u, u_{Q}) \in X \times X_{Q} \times U \times U_{Q} \mid u_{Q} = \operatorname{sen}(x)\}. \quad (6)$$

For notational convenience, we introduce the transition maps $\hat{r}^{\perp} : \hat{X} \times \hat{U}^* \to 2^{\hat{X}}$ and $r^{\perp} : X \times U^* \to 2^X$ defined as follows: For any $\hat{x} \in \hat{X}$, $x \in X$, $\hat{t} \in \hat{U}^*$, $t \in U^*$, $\hat{v} \in \hat{U}$, and $v \in U$,

$$\hat{r}^{\perp}(\hat{x},\hat{t}\hat{v}) = \begin{cases} \hat{r}(\hat{x},\hat{t}\hat{v}) \cup \hat{r}(\hat{x},\hat{t}\hat{u}^{\perp}) & \text{if } \text{last}(\hat{t}) \neq \hat{u}^{\perp}, \\ \hat{r}(\hat{x},\hat{t}\hat{v}) & \text{if } \text{last}(\hat{t}) = \hat{u}^{\perp}; \end{cases}$$
(7)

$$r^{\perp}(x,tv) = \begin{cases} r(x,tv) \cup r(x,tu^{\perp}) & \text{if } \text{last}(t) \neq u^{\perp}, \\ r(x,tv) & \text{if } \text{last}(t) = u^{\perp}. \end{cases}$$
(8)

Now, we introduce a symbolic transition system \hat{S}_D that predicts the plant state at time when the actuator receives the delayed input from the network.

Definition 5: Let $L = L^{ca} + L^{sc} \in \mathbb{Z}_{\geq 0}$ be the worst delay time. Then, a symbolic plant model with networks induced

by \hat{S} , denoted by \hat{S}_D , is defined by the following transition system:

$$\hat{S}_D = (\hat{X}_D, \hat{X}_{D0}, \hat{U}_D, \hat{r}_D),$$
(9)

where

- $\hat{X}_D = \{(\hat{x}, \hat{u}^0, \hat{u}^1, \dots, \hat{u}^L, n) \mid \hat{x} \in \hat{X}, \forall i \in \{0, 1, \dots, L\} : \hat{u}^i \in \hat{U}, n \in \{0, 1, \dots, L^{sc} + 1\}\};$
- $\hat{X}_{D0} = \{ (\hat{x}_0, \hat{u}^{\perp}, \dots, \hat{u}^{\perp}, 0) \in \hat{X}_D \mid \hat{x}_0 \in \hat{X}_0 \};$
- $\hat{U}_D = \hat{U}$; and
- $\hat{r}_D: \hat{X}_D \times \hat{U}_D \to 2^{\hat{X}_D}$ is defined as follows:

$$\begin{split} \hat{r}_{D}((\hat{x}, \hat{u}^{0}, \hat{u}^{1}, \dots, \hat{u}^{L}, n), \hat{u}_{D}) \\ &= \begin{cases} \{(\hat{x}, \hat{u}^{1}, \hat{u}^{2}, \dots, \hat{u}^{L}, \hat{u}_{D}, n+1)\} \\ \cup \{(\hat{x}^{n}, \hat{u}^{1}, \hat{u}^{2}, \dots, \hat{u}^{L}, \hat{u}_{D}, n') \mid \\ n' \in \{0, 1, \dots, n\}, \\ \hat{x}^{n} \in \hat{r}^{\perp}(\hat{x}, \hat{u}^{L^{sc} - n + 1} \hat{u}^{L^{sc} - n + 2} \dots \hat{u}^{L^{sc} - n' + 1})\} \\ &\text{if } \forall \hat{x}^{*} \in \hat{r}^{\perp}(\hat{x}, \hat{u}^{L^{sc} - n + 1} \hat{u}^{L^{sc} - n + 2} \dots \hat{u}^{L}) \\ &\hat{r}^{\perp}(\hat{x}^{*}, \hat{u}_{D}) \neq \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

The state \hat{x}_D is composed of the latest received symbolic data \hat{x} , the L + 1-length FIFO queue $\hat{u}^0, \hat{u}^1, \dots, \hat{u}^L$ that holds the latest symbolic control inputs transmitting in the network, and the number *n* that means how many time steps the controller should predict to determine a control input. The following lemma shows how \hat{S}_D predicts the plant state.

Lemma 1: The following parameterized relation $R_D(\epsilon) \subseteq \hat{X}_D \times X_D \times \hat{U}_D \times U_D$ is a (κ, β, λ) -acASR from \hat{S}_D to S_D with $\mathsf{d}_D(\hat{u}_D, u_D) = d$ for any $\hat{u}_D \in \hat{U}_D$ and $u_D \in U_D$:

$$R_{D}(\epsilon) = \{ ((\hat{x}, \hat{u}^{0}, \dots, \hat{u}^{L}, n), (x_{L^{ca}}, x, x_{Q}), \hat{u}_{D}, (\hat{u}_{L^{ca}}, H_{L^{ca}}(x_{L^{ca}}), \operatorname{sen}(x))) \mid \hat{u}_{D} = \hat{u}_{L^{ca}} \land$$

$$\exists \hat{x}^{*} \in \hat{r}^{\perp}(\hat{x}, \hat{u}^{L^{sc}-n+1} \hat{u}^{L^{sc}-n+2} \dots \hat{u}^{L^{sc}}) :$$

$$(\hat{x}^{*}, x, \hat{u}^{L^{sc}+1}, H_{L^{ca}}(x_{L^{ca}})) \in R(\epsilon) \}.$$
(10)

A proof of Lemma 1 is shown in Appendix A. Intuitively, in (10), \hat{x}^* is the actual abstracted state of *x*, that is, the current state of the plant *S*. \hat{S}_D calculates all reachable states from the latest received state \hat{x} with the input history $\hat{u}^0 \dots \hat{u}^L$. It is noticed that the controller does not always receive a measured state due to network delays and packet dropouts. Nevertheless, the definition of \hat{r}_D ensures that the transitions with $\hat{u}^{L^{x+1}}$ are defined for all predicted states, which means that the controlled plant satisfies the specification.

4. Controller Synthesis

In this section, we design a symbolic feedback controller induced by \hat{S}_C . In order to connect \hat{S}_C to the networks, we introduce a symbol $\hat{u}_C^{\perp} \in \hat{U}_C$ that denotes an input satisfying the following condition:

• For any $\hat{x}_C \in \hat{X}_C$, $\hat{u}_C^{\perp} \in \hat{U}_C(\hat{x}_C)$.

We assume that the ASR \hat{R}_C from \hat{S}_C to \hat{S} satisfies the

following conditions:

• For any $\hat{u}_C \in \hat{U}_C$, there exists $\hat{u} \in \hat{U}$ such that

$$\forall (\hat{x}_C, \hat{x}) \in \hat{R}_{CX} :$$
$$\hat{u}_C \in \hat{U}_C(\hat{x}_C) \Rightarrow (\hat{x}_C, \hat{x}, \hat{u}_C, \hat{u}) \in \hat{R}_C; \text{ and }$$

• For any $(\hat{x}_C, \hat{x}) \in \hat{R}_{CX}, (\hat{x}_C, \hat{x}, \hat{u}_C^{\perp}, \hat{u}^{\perp}) \in \hat{R}_C.$

For notational convenience, we introduce the following transition map $\hat{r}_C^{\perp} : \hat{X}_C \times \hat{U}_C^* \to 2^{\hat{X}_C}$ defined as follows: For any $\hat{X}_C \in \hat{X}_C, \hat{t}_C \in \hat{U}_C^*$ and $\hat{v}_C \in \hat{U}_C$,

$$\hat{r}_{C}^{\perp}(\hat{x}_{C}, \hat{t}_{C}\hat{v}_{C}) = \begin{cases} \hat{r}_{C}(\hat{x}_{C}, \hat{t}_{C}\hat{v}_{C}) \cup \hat{r}_{C}(\hat{x}_{C}, \hat{t}_{C}\hat{u}_{C}^{\perp}) \\ & \text{if } \operatorname{last}(\hat{t}_{C}) \neq \hat{u}_{C}^{\perp}, \\ \hat{r}_{C}(\hat{x}_{C}, \hat{t}_{C}\hat{v}_{C}) \\ & \text{if } \operatorname{last}(\hat{t}_{C}) = \hat{u}_{C}^{\perp}. \end{cases}$$
(11)

Then, we introduce a transition system whose state is a set of all candidates of the plant state predicted by \hat{S}_D .

Definition 6: We define a system $\hat{\mathbf{S}}_{\mathbf{C}} = (\hat{\mathbf{X}}_{\mathbf{C}}, \hat{\mathbf{X}}_{\mathbf{C}0}, \hat{\mathbf{U}}_{\mathbf{C}}, \hat{\mathbf{r}}_{\mathbf{C}})$ induced by \hat{S}_{C} where

•
$$\hat{\mathbf{X}}_{\mathbf{C}} = 2^{X_C} \setminus \{\emptyset\};$$

• $\hat{\mathbf{X}}_{\mathbf{C}0} = \{\hat{r}_C^{\perp}(\hat{x}_{C0}, \widetilde{\hat{u}_C^{\perp} \dots \hat{u}_C^{\perp}}) \mid \hat{x}_{C0} \in \hat{X}_{C0}\} \subseteq \hat{\mathbf{X}}_{\mathbf{C}};$

- $\hat{\mathbf{U}}_{\mathbf{C}} = \hat{U}_{C}$; and
- $\hat{\mathbf{r}}_{C}: \hat{\mathbf{X}}_{C} \times \hat{\mathbf{U}}_{C} \to 2^{\hat{\mathbf{X}}_{C}}$ is defined as follows:

$$\mathbf{\hat{r}}_{\mathbf{C}}(\mathbf{\hat{x}}_{\mathbf{C}}, \hat{u}_{C}) = \begin{cases} 2^{\bigcup_{\hat{x}_{C} \in \mathbf{\hat{x}}_{C}} \hat{r}_{C}(\hat{x}_{C}, \hat{u}_{C})} \setminus \{\emptyset\} \\ \text{if } \forall \hat{x}_{C} \in \mathbf{\hat{x}}_{\mathbf{C}} : \hat{r}_{C}(\hat{x}_{C}, \hat{u}_{C}) \neq \emptyset, \\ \emptyset & \text{otherwise.} \end{cases}$$

It is noticed that each state of $\hat{\mathbf{S}}_{\mathbf{C}}$ is a set of states of \hat{S}_{C} , which means that the cardinality of $\hat{\mathbf{S}}_{\mathbf{C}}$ increases exponentially with respect to that of \hat{S}_{C} . This is because there are several possible current states if the physical plant *S* is non-deterministic. Note that if *S* is deterministic, we can replace $\hat{\mathbf{S}}_{\mathbf{C}}$ with \hat{S}_{C} . Then, we have the following lemma.

Lemma 2: The following relation $\hat{\mathbf{R}}_{\mathbf{C}} \subseteq \hat{\mathbf{X}}_{\mathbf{C}} \times \hat{U}_{\mathbf{C}} \times \hat{U}_{D}$ is an ASR from $\hat{\mathbf{S}}_{\mathbf{C}}$ to \hat{S}_{D} :

$$\hat{\mathbf{R}}_{\mathbf{C}} = \{ (\hat{\mathbf{x}}_{\mathbf{C}}, (\hat{x}, \hat{u}^{0}, \hat{u}^{1}, \dots, \hat{u}^{L}, n), \hat{u}_{C}, \hat{u}) \mid \\
\forall \hat{x}^{*} \in \hat{r}^{\perp} (\hat{x}, \hat{u}^{L^{sc} - n + 1} \hat{u}^{L^{sc} - n + 2} \dots \hat{u}^{L}), \\
\exists \hat{x}_{C} \in \hat{\mathbf{x}}_{\mathbf{C}} : (\hat{x}_{C}, \hat{x}^{*}, \hat{u}_{C}, \hat{u}) \in \hat{R}_{C} \}.$$
(12)

A proof of Lemma 2 is shown in Appendix B. Lemma 2 implies that $\hat{\mathbf{S}}_{\mathbf{C}}$ is a feedback controller of \hat{S}_D . Intuitively, in (12), $\hat{\mathbf{S}}_{\mathbf{C}}$ determines a control input \hat{u}_C that realizes a desired behavior for each predicted state. Thus, the control specification is achieved in spite of the existence of network delays and packet dropouts.

We have the following main theorem that ensures that the networked physical plant S_D controlled by the feedback controller $\mathbf{S}_{\mathbf{C}} := \hat{\mathbf{S}}_{\mathbf{C}} \times_{\hat{\mathbf{R}}_C} \hat{S}_D$ exhibits a desired behavior. **Theorem 2:** The following relation $\mathbf{R}_{\mathbf{C}}(\epsilon) \subseteq (\hat{\mathbf{X}}_{\mathbf{C}} \times \hat{X}_D) \times X_D \times (\hat{\mathbf{U}}_{\mathbf{C}} \times \hat{U}_D) \times U_D$ is a (κ, β, λ) -acASR from $\mathbf{S}_{\mathbf{C}} = \hat{\mathbf{S}}_{\mathbf{C}} \times_{\hat{\mathbf{R}}_{\mathbf{C}}}$ \hat{S}_D to S_D with $\mathbf{d}_{\mathbf{C}}((\hat{u}_C, \hat{u}_D), u_D) = \mathbf{d}_D(\hat{u}_D, u_D)$:

$$\mathbf{R}_{\mathbf{C}}(\epsilon) = \{ ((\hat{\mathbf{x}}_{\mathbf{C}}, \hat{x}_D), x_D, (\hat{u}_C, \hat{u}_D), u_D) \mid \\ (\hat{x}_D, x_D, \hat{u}_D, u_D) \in R_D(\epsilon) \land (\hat{\mathbf{x}}_C, \hat{x}_D) \in \hat{\mathbf{R}}_{\mathbf{CX}} \}.$$
(13)

Proof: Lemma 1 shows that $R_D(\epsilon)$ is the (κ, β, λ) -acASR from \hat{S}_D to S with d_D . Lemma 2 shows that $\hat{\mathbf{R}}_C$ is the ASR from $\hat{\mathbf{S}}_C$ to \hat{S}_D , which implies with Theorem 1 that $\mathbf{R}_C(\epsilon)$ defined by (13) is a (κ, β, λ) -acASR from \mathbf{S}_C to S_D with $d_C((\hat{u}_C, \hat{u}_D), u_D) = d_D(\hat{u}_D, u_D)$.

The existence of the (κ, β, λ) -acASR from the proposed controller to the delayed plant implies that, by the predictionbased approach, the error between the state of the controller and that of the plant does not increase though network disturbances occur. In addition, the effects of network disturbances are suppressed, and the specification is still satisfied because the input is determined in such a way that $R(\epsilon)$ holds for all predicted states.

5. Illustrative Example

Let us consider a physical plant given by

$$\begin{bmatrix} \xi_1[k+1] \\ \xi_2[k+1] \end{bmatrix} = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} \xi_1[k] \\ \xi_2[k] \end{bmatrix} + \begin{bmatrix} -0.1487 \\ 0.06 \end{bmatrix} u[k].$$
(14)

We model the physical plant by a transition system $S = (X, X_0, U, r)$, where $X = \mathbb{R}^2$, $X_0 =] -0.05$, $0.05[^2$, $U = \{0, 8\}$, and *r* is computed using (14). We construct its abstracted model $\hat{S} = (\hat{X}, \hat{X}_0, \hat{U}, \hat{r})$, where $\hat{X} = [[-2, 2]^2]_{0.05}$, $\hat{X}_0 = \{[0 \ 0 \]^T\}$, and $\hat{U} = U$. The transition map \hat{r} is given by the following dynamics:

$$\begin{bmatrix} \hat{\xi}_{1}[k+1] \\ \hat{\xi}_{2}[k+1] \end{bmatrix}$$

= rd $\left(\begin{bmatrix} 0.25 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} \hat{\xi}_{1}[k] \\ \hat{\xi}_{2}[k] \end{bmatrix} + \begin{bmatrix} -0.1487 \\ 0.06 \end{bmatrix} \hat{u}[k] \right),$ (15)

where $rd([\xi_1 \xi_2]^T)$ rounds $[\xi_1 \xi_2]^T$ off to the nearest value in \hat{X} . Then, it is easily proved that the following relation $R(\epsilon) \subseteq \hat{X} \times X \times \hat{U} \times U$ is a (0.05, 0.5, 0)-acASR from \hat{S} to S:

$$R(\epsilon) = \{ (\hat{x}, x, \hat{u}, u) \mid |\hat{x} - x| \le \epsilon \land \hat{u} = u \}.$$

$$(16)$$

We design a transition system that describes desired behaviors of \hat{S} . The specification is that ξ_1 goes back and forth between -1.5 and 0. We design $\hat{S}_C = (\hat{X}_C, \hat{X}_{C0}, \hat{U}_C, \hat{r}_C)$, where $\hat{X}_C = \hat{X} \times \{0, 1\}, \hat{X}_{C0} = \{[0 \ 0 \ 0]^T\}, \text{ and } \hat{U}_C = \hat{U} = U$. The transition map $\hat{r}_C : \hat{X}_C \times \hat{U}_C \to 2^{\hat{X}_C}$ is defined as follows: For any $[\hat{\xi}_{C1} \hat{\xi}_{C2} \hat{\xi}_{C3}]^T \in \hat{X}_C = \hat{X} \times \{0, 1\}$ and $\hat{u}_C \in \hat{U}_C$,





$$\begin{split} \hat{r}_{C}([\hat{\xi}_{C1}\,\hat{\xi}_{C2}\,\hat{\xi}_{C3}\,]^{T},\hat{u}_{C}) \\ &= \begin{cases} \left\{ [\hat{\xi}_{C1}'\,\hat{\xi}_{C2}'\,1\,]^{T} \right| \\ [\hat{\xi}_{C1}'\,\hat{\xi}_{C2}'\,]^{T} \in \hat{r}([\hat{\xi}_{C1}\,\hat{\xi}_{C2}\,]^{T},\hat{u}_{C}) \right\} \\ &\text{if } |\hat{\xi}_{1}+1.5| > e \ \land \ |\hat{\xi}_{1}'+1.5| \le e, \\ \left\{ [0\,0\,0\,]^{T} \right\} \\ &\text{if } [\hat{\xi}_{C1}\,\hat{\xi}_{C2}\,]^{T} = [0\,0\,]^{T} \ \land \ \hat{u}_{C} = 0, \\ \left\{ [\hat{\xi}_{C1}'\,\hat{\xi}_{C2}'\,\hat{\xi}_{C3}\,]^{T} \right| \\ [\hat{\xi}_{C1}'\,\hat{\xi}_{C2}'\,]^{T} \in \hat{r}([\hat{\xi}_{C1}\,\hat{\xi}_{C2}\,]^{T},\hat{u}_{C}) \right\} \\ &\text{otherwise,} \end{cases}$$

where $e \in \mathbb{R}_{\geq 0}$ denotes an allowed error from -1.5. The third element $\hat{\xi}_3 \in \{0, 1\}$ indicates the control mode. If $\hat{\xi}_3 = 0$, the controller chooses $\hat{u}_C = 8$ because ξ_1 does not reach -1.5. On the other hand, if $\hat{\xi}_3 = 1$, the controller can choose both $\hat{u}_C = 0$ and $\hat{u}_C = 8$ because ξ_1 has already reached -1.5. When $[\hat{\xi}_{C1} \hat{\xi}_{C2}]^T = [0 \ 0]^T \land \hat{u}_C = 0$, \hat{r}_C transits to $[0 \ 0 \ 0]^T$ to reset the mode. Then, it is easily proved that the following relation $\hat{R}_C \subseteq \hat{X}_C \times \hat{X} \times \hat{U}_C \times \hat{U}$ is an ASR from \hat{S}_C to \hat{S} :

$$\hat{R}_{C} = \{ ([\hat{\xi}_{C1} \ \hat{\xi}_{C2} \ \hat{\xi}_{C3} \]^{T}, [\hat{\xi}_{1} \ \hat{\xi}_{2} \]^{T}, \hat{u}_{C}, \hat{u}) \mid \\ \hat{\xi}_{C1} = \hat{\xi}_{1} \ \land \ \hat{\xi}_{C2} = \hat{\xi}_{2} \ \land \ \hat{u}_{C} = \hat{u} \}.$$

We consider the case where the control signal \hat{u} and measured state $\hat{x} = \operatorname{sen}(x)$ are sent via unreliable networks where $\hat{u}_C^{\perp} = \hat{u}^{\perp} = u^{\perp} = 0$. Then, it is shown that the network satisfies the conditions in Sects. 3 and 4. We construct the proposed controller $\hat{\mathbf{S}}_{\mathbf{C}}$ as in Sect. 4 for the networked plant. Intuitively, in this case, $\hat{\mathbf{S}}_{\mathbf{C}}$ determines a control input as follows: When all modes $\hat{\xi}_{C3}$ of the candidates predicted by \hat{S}_D are 1, $\hat{\mathbf{S}}_{\mathbf{C}}$ chooses $\hat{u}_C = 0$; Otherwise, $\hat{\mathbf{S}}_{\mathbf{C}}$ chooses $\hat{u}_C = 8$. Let e = 0.1, $L^{sc} = 2$, and $L^{ca} = 3$. By computer simulation, we obtain the time response of $\xi_1(t)$ shown in Fig. 3. It is shown that the controlled plant satisfies the specification in spite of network disturbances.

6. Conclusion

We considered a symbolic networked controller based on the state prediction. A transition system that describes desired behaviors of the plant is given when there is no network disturbance. The system is modified to predict states of the plant by considering the measured state, the input history, and the network protocol. We showed that the there exists the acASR from the proposed controller to the networked plant, which means that the prediction-based approach suppresses the effects of network disturbance and the specification is achieved. It is future work to analyze the stability of the controller quantitatively by introducing cost functions.

Acknowledgments

This work was supported in part by JSPS KAKENHI Grant Number JP15K14007.

References

- [1] P. Tabuada, Verification and Control of Hybrid Systems—A Symbolic Approach, Springer Verlag, 2009.
- [2] R. Milner, Communication and Concurrency, Prentice-Hall, 1989.
- [3] A. Girard and G.J. Pappas, "Approximation metrics for discrete and continuous systems," IEEE Trans. Automatic Control, vol.52, no.5, pp.782–798, 2007.
- [4] A. Girard and G.J. Pappas, "Approximation bisimulation: a bridge between computer science and control theory," European J. of Control, vol.5, no.6, pp.568–578, 2011.
- [5] P. Tabuada, "An approximate simulation approach to symbolic control," IEEE Trans. Automatic Control, vol.53, no.6, pp.1406–1418, 2008.
- [6] G. Pola, A. Girard, and P. Tabuada, "Approximately bisimilar symbolic models for nonlinear control systems," Automatica, vol.44, no.10, pp.2508–2516, 2008.
- [7] A. Girard, G. Pola, and P. Tabuada, "Approximately bisimilar symbolic models for incrementally stable switched systems," IEEE Trans. Automatic Control, vol.55, no.1, pp.116–126, 2010.
- [8] E.A. Gol, X. Ding, M. Lazar, and C. Belta, "Finite bisimulation for switched linear systems," IEEE Trans. Automatic Control, vol.59, no.12, pp.3122–3134, 2014.
- [9] G. Pola, P. Pepe, M.D. Di Benedetto, and P. Tabuada, "Symbolic models for nonlinear time-delay systems using approximate bisimulation," Systems and Control Letters, vol.59, no.6, pp.365–373, 2010.
- [10] R. Alur, T.A. Henzinger, O. Kaupfman, and M.Y. Vardi, "Alternating refinement relations," Lect. Notes Comput. Sci., vol.1466, pp.163– 178, Springer, 1998.
- [11] G. Pola and P. Tabuada, "Symbolic models for nonlinear control systems: alternating approximate bisimulations," SIAM J. Control Optim., vol.48, no.2, pp.719–733, 2009.
- [12] G. Pola, P. Pepe, and M.D.D. Benedetto, "Symbolic models for time-varying time-delay systems via alternating approximate bisimulation," Int. J. Robust Nonlinear Control, vol.25, no.14, pp.2328– 2347, 2015.
- [13] M. Rungger and P. Tabuada, "A notion of robustness for cyberphysical systems," IEEE Trans. Automatic Control, vol.61, no.8, pp.2108–2123, 2016.
- [14] M. Mizoguchi and T. Ushio, "Observer-based similarity output feedback control of cyber-physical systems," Proc. 5th IFAC Conf. on Analysis and Design of Hybrid Systems, pp.248–253, 2015.
- [15] A. Borri, G. Pola, and M.D.D. Benedetto, "A symbolic approach to the design of nonlinear networked control systems," Proc. 15th Hybrid Systems: Computation and Control, pp.255–264, 2012.
- [16] M. Zamani, M. Mazo, and A. Abate, "Finite abstractions of networked control systems," Proc. 53rd IEEE Conference on Decision and Control, pp.95–100, 2014.
- [17] T. Mita, "Optimal digital feedback control systems counting computation time of control laws," IEEE Trans. Automatic Control, vol.30, no.6, pp.542–548, 1985.

Appendix A: Proof of Lemma 1

We will show that $R_D(\epsilon)$ satisfies the conditions of a (κ, β, λ) -acASR from \hat{S}_D to S_D with d_D .

Consider any $\hat{x}_{D0} = (\hat{x}_0, \hat{u}^{\perp}, \dots, \hat{u}^{\perp}, 0) \in \hat{X}_{D0}$. Let $x_0 \in X_0$ be an initial state such that $(\hat{x}_0, x_0) \in R_X(\kappa)$. Note that by the (κ, β, λ) -acASR $R(\epsilon)$, x_0 always exists, and we have $\operatorname{sen}(x_0) = \hat{x}_0$. Let $x_{L^{ca}0} = (\perp, \dots, \perp) \in X_{L^{ca}0}$ and $x_{Q0} = (\operatorname{sen}(x_0), \perp, \dots, \perp, l_0) = (\hat{x}_0, \perp, \dots, \perp, L^{sc} + 1) \in X_{Q0}$. Then, $(x_{L^{ca}0}, x_0, x_{Q0}) \in X_{D0}$ holds, and by the definition of $R_D(\epsilon)$, we have

 $(\hat{x}_{D0}, (x_{L^{ca}0}, x_0, x_{Q0})) \in R_{DX}(\kappa).$

First, consider any $(\hat{x}_D, (x_{L^{ca}}, x, x_Q)) \in R_{DX}(\epsilon)$. Let $\hat{x}_D = (\hat{x}, \hat{u}^0, \hat{u}^1, \dots, \hat{u}^L, n)$ and $x_Q = (\operatorname{sen}(x), \hat{x}^1, \hat{x}^2, \dots, \hat{x}^{L^{sc}}, l)$. By the definition of $R_D(\epsilon)$, we have

$$\begin{aligned} \exists \hat{x}^* \in \hat{r}^{\perp}(\hat{x}, \hat{u}^{L^{sc}-n+1}\hat{u}^{L^{sc}-n+2}\dots\hat{u}^{L^{sc}}) :\\ (\hat{x}^*, x, \hat{u}^{L^{sc}+1}, H_{L^{ca}}(x_{L^{ca}})) \in R(\epsilon) \end{aligned}$$

Choose any $\hat{u}_D \in \hat{U}_D(\hat{x}_D)$. Then, $(\hat{u}_D, H_{L^{ca}}(x_{L^{ca}}), \operatorname{sen}(x)) \in \hat{U}_D((x_{L^{ca}}, x, x_O))$ holds, and we have

$$(\hat{x}_D, (x_{L^{ca}}, x, x_Q), \hat{u}_D, (\hat{u}_D, H_{L^{ca}}(x_{L^{ca}}), \operatorname{sen}(x))) \in R_D(\epsilon).$$

Note that by the definitions of $S_{L^{ca}}$ and \hat{S}_{D} , we have

 $x_{L^{ca}} \in \{\hat{u}_D, \bot\} \times \{\hat{u}^L, \bot\} \times \{\hat{u}^{L-1}, \bot\}$ $\times \dots \times \{\hat{u}^{L^{sc}+1}, \bot\}.$

Then, after L^{ca} time steps are elapsed, any plant state $x^* \in X$ satisfies

$$x^* \in r^{\perp}(x, \operatorname{act}(\hat{u}^{L^{sc}+1})\operatorname{act}(\hat{u}^{L^{sc}+2}) \dots \operatorname{act}(\hat{u}^{L})).$$

By the acASR $R(\epsilon)$ and the definition of \hat{r}_D , we have the following condition:

$$\begin{aligned} \forall x^* \in r^{\perp}(x, \operatorname{act}(\hat{u}^{L^{sc}+1})\operatorname{act}(\hat{u}^{L^{sc}+2}) \dots \operatorname{act}(\hat{u}^{L})), \\ \exists \hat{x}^* \in \hat{r}^{\perp}(\hat{x}, \hat{u}^{L^{sc}+1} \hat{u}^{L^{sc}+2} \dots \hat{u}^{L}) : \\ (\hat{x}^*, x^*, \hat{u}_D, \operatorname{act}(\hat{u}_D)) \in R(\epsilon_{L^{ca}}), \end{aligned}$$

where $\epsilon_0 = \epsilon$ and $\epsilon_{i+1} = \kappa + \beta \epsilon_i + \lambda d$ for $i \in \{0, \dots, L^{ca} - 1\}$. Next, consider any $(\hat{x}'_{L^{ca}}, x', x'_O) \in X_D$ such that

$$(\hat{x}'_{L^{ca}}, x', x'_Q)$$

 $\in r_D((\hat{x}_{L^{ca}}, x, x_Q), (\hat{u}_D, H_{L^{ca}}(x_{L^{ca}}), \operatorname{sen}(x)).$

By the definitions of r and r_0 , we have

$$x' \in r(x, H_{L^{ca}}(x_{L^{ca}}));$$
 and
 $\exists l' \in \{1, 2, \dots, L^{sc} + 1\} : x'_Q \in \{(\hat{x}^*, \operatorname{sen}(x), \hat{x}^1, \dots, \hat{x}^{L^{sc}-1}, \bot, \dots, \bot, l') \mid \hat{x}^* \in \{\operatorname{sen}(x'), \bot\}\}.$

Then, \hat{S}_D receives an output sequence: $H_Q(x'_Q) = \hat{x}^{L^{sc}-l'+1} \hat{x}^{L^{sc}-l'+2} \dots \hat{x}^{L^{sc}}$. Now, we consider two cases.

1. If $H_Q(x'_Q) = \bot$, then the state of \hat{S}_D transits to the following state:

$$\hat{x}'_D = (\hat{x}, \hat{u}^1, \hat{u}^2, \dots, \hat{u}^L, \hat{u}_D, n+1) \in \hat{r}_D(\hat{x}_D, \hat{u}_D).$$

Then, we have

$$\begin{aligned} \exists \hat{x}^{**} \in \hat{r}^{\perp}(\hat{x}, \hat{u}^{L^{sc}-n+1} \hat{u}^{L^{sc}-n+2} \dots \hat{u}^{L^{sc}+1}) : \\ (\hat{x}^{**}, x') \in R_X(\kappa + \beta \epsilon + \lambda d), \end{aligned}$$

which implies $(\hat{x}'_D, x') \in R_{DX}(\kappa + \beta \epsilon + \lambda d)$.

2. If $H_Q(x'_Q) \neq \bot$, then \hat{S}_D receives several states. Recall that each data is time-stamped. Let $\hat{x}^n \in \hat{X}$ is the latest data in $H_Q(x'_Q)$. Note that the data is received only when the delay time does not increase. Thus, let \hat{x}^n be the abstracted plant state $k \in \{0, 1, ..., n\}$ time steps before. It is also noticed that we have $\hat{x}^n \in \hat{r}^{\bot}(\hat{x}, \hat{u}^{L^{sc}-n+1}\hat{u}^{L^{sc}-n+2} \dots \hat{u}^{L^{sc}-k+1})$. Then, the state of \hat{S}_D transits to the following state:

$$\hat{x}'_D = (\hat{x}^n, \hat{u}^1, \hat{u}^2, \dots, \hat{u}^L, \hat{u}_D, k) \in \hat{r}_D(\hat{x}_D, \hat{u}_D).$$

Then, we have

$$\exists \hat{x}^{**} \in \hat{r}^{\perp}(\hat{x}^n, \hat{u}^{L^{sc}-k+2}\hat{u}^{L^{sc}-k+3}\dots\hat{u}^{L^{sc}+1}):$$

$$(\hat{x}^{**}, x') \in R_X(\kappa + \beta\epsilon + \lambda d),$$

which implies $(\hat{x}'_D, x') \in R_{DX}(\kappa + \beta \epsilon + \lambda d)$.

Therefore, in both cases, it is shown that $R_D(\epsilon)$ satisfies the conditions of the (κ, β, λ) -acASR from \hat{S}_D to S_D with d_D .

Appendix B: Proof of Lemma 2

Consider any $\hat{\mathbf{x}}_{C0} \in \hat{\mathbf{X}}_{C0}$. The definition of $\hat{\mathbf{X}}_{C0}$ implies that there exists $\hat{x}_{C0} \in \hat{X}_{C0}$ such that $\hat{\mathbf{x}}_{C0} = \hat{r}_C^{\perp}(\hat{x}_{C0}, \hat{u}_C^{\perp} \dots \hat{u}_C^{\perp})$. By the ASR \hat{R}_C , there always exists $\hat{x}_0 \in \hat{X}_0$ such that $(\hat{x}_{C0}, \hat{x}_0) \in \hat{R}_{CX}$. Then, the definition of \hat{S}_D implies that $\hat{x}_{D0} = (\hat{x}_0, \hat{u}^{\perp}, \dots, \hat{u}^{\perp}, 0) \in \hat{X}_{D0}$. From the assumptions of \hat{R}_C , we have

$$\forall \hat{x}^* \in \hat{r}^{\perp}(\hat{x}_0, \hat{u}^{\perp}, \dots, \hat{u}^{\perp}), \exists \hat{x}_C^* \in \hat{r}_C^{\perp}(\hat{x}_{C0}, \hat{u}_C^{\perp}, \dots, \hat{u}_C^{\perp}) :$$
$$(\hat{x}_C^*, \hat{x}^*) \in \hat{R}_{CX},$$

which implies $(\hat{\mathbf{x}}_{\mathbf{C}0}, \hat{x}_{D0}) \in \hat{\mathbf{R}}_{\mathbf{C}X}$.

First, consider any $(\hat{\mathbf{x}}_{\mathbf{C}}, (\hat{x}, \hat{u}^0, \hat{u}^1, \dots, \hat{u}^L, n)) \in \hat{\mathbf{R}}_{\mathbf{C}X}$. By the definition of $\hat{\mathbf{R}}_{\mathbf{C}}$, we have

$$\forall \hat{x}^* \in \hat{r}^{\perp}(\hat{x}, \hat{u}^{L^{sc}-n+1}\hat{u}^{L^{sc}-n+2}\dots\hat{u}^L), \exists \hat{x}_C \in \hat{\mathbf{x}}_C :$$
$$(\hat{x}_C, \hat{x}^*) \in \hat{R}_{CX}.$$

Choose any $\hat{u}_C \in \hat{\mathbf{U}}_{\mathbf{C}}(\hat{\mathbf{x}}_{\mathbf{C}})$. By the definition of $\hat{\mathbf{r}}_{\mathbf{C}}$ and the assumptions of \hat{R}_C , there exists $\hat{u}_D \in \hat{U}_D(\hat{x}_D)$ such that

$$\forall \hat{x}^* \in \hat{r}^{\perp}(\hat{x}, \hat{u}^{L^{sc}-n+1}\hat{u}^{L^{sc}-n+2}\dots\hat{u}^L), \exists \hat{x}_C \in \hat{\mathbf{x}}_C : \\ (\hat{x}_C, \hat{x}^*, \hat{u}_C, \hat{u}_D) \in \hat{R}_C,$$

which implies $(\hat{\mathbf{x}}_{\mathbf{C}}, (\hat{x}, \hat{u}^0, \hat{u}^1, \dots, \hat{u}^L, n), \hat{u}_C, \hat{u}_D) \in \hat{\mathbf{R}}_{\mathbf{C}}.$

Next, consider any $\hat{x}'_D \in \hat{r}_D((\hat{x}, \hat{u}^0, \hat{u}^1, \dots, \hat{u}^L, n), \hat{u}_D)$. We consider two cases.

1. Consider the case where the state of \hat{S}_D transits to the following state:

$$\hat{x}'_D = (\hat{x}, \hat{u}^1, \hat{u}^2, \dots, \hat{u}^L, \hat{u}_D, n+1).$$

Then, by the ASR \hat{R}_C , there exists $\hat{\mathbf{x}}'_C \neq \emptyset \subseteq \bigcup_{\hat{x}_C \in \hat{\mathbf{x}}_C} \hat{r}_C(\hat{x}_C, \hat{u}_C)$ such that

$$\begin{aligned} \forall \hat{x}^{**} \in \hat{r}^{\perp}(\hat{x}, \hat{u}^{L^{sc}-n+1} \hat{u}^{L^{sc}-n+2} \dots \hat{u}^{L} \hat{u}_{D}), \\ \exists \hat{x}_{C}' \in \hat{\mathbf{x}}_{C}' : (\hat{x}_{C}', \hat{x}') \in \hat{R}_{CX}. \end{aligned}$$

Note that $\hat{\mathbf{x}}'_{\mathbf{C}} \in \hat{\mathbf{r}}_{\mathbf{C}}(\hat{\mathbf{x}}_{\mathbf{C}}, \hat{u}_{\mathbf{C}})$. By the definition of $\hat{\mathbf{R}}_{\mathbf{C}}$, we have $(\hat{\mathbf{x}}'_{\mathbf{C}}, \hat{x}'_{\mathbf{D}}) \in \hat{\mathbf{R}}_{\mathbf{C}X}$.

2. Consider the case where there exist $n' \in \{0, 1, ..., n\}$ and $\hat{x}^n \in \hat{r}^{\perp}(\hat{x}, \hat{\mu}^{L^{sc}-n+1}\hat{\mu}^{L^{sc}-n+2}\dots\hat{\mu}^{L^{sc}-n'+1})$ such that $\hat{x}'_D = (\hat{x}^n, \hat{\mu}^1, \hat{\mu}^2, \dots, \hat{\mu}^L, \hat{\mu}_D, n')$. Then, we have the following condition:

$$\hat{r}^{\perp}(\hat{x}^n, \hat{u}^{L^{sc}-n'+2}\hat{u}^{L^{sc}-n'+3}\dots\hat{u}^L\hat{u}_D)$$
$$\subseteq \hat{r}^{\perp}(\hat{x}, \hat{u}^{L^{sc}-n+1}\hat{u}^{L^{sc}-n+2}\dots\hat{u}^L\hat{u}_D).$$

Thus, by the ASR \hat{R}_C , there exists $\hat{\mathbf{x}}'_C \subseteq \bigcup_{\hat{x}_C \in \hat{\mathbf{x}}_C} \hat{r}_C(\hat{x}_C, \hat{u}_C)$ such that $\hat{\mathbf{x}}'_C \neq \emptyset$ and

$$\begin{aligned} \forall \hat{x}^{**} \in \hat{r}^{\perp}(\hat{x}^n, \hat{u}^{L^{sc}-n'+2}\hat{u}^{L^{sc}-n'+3} \dots \hat{u}^L \hat{u}_D), \\ \exists \hat{x}'_C \in \hat{\mathbf{x}}'_C : (\hat{x}'_C, \hat{x}') \in \hat{R}_{CX}, \end{aligned}$$

which implies $(\hat{\mathbf{x}}'_{\mathbf{C}}, \hat{x}'_{D}) \in \hat{\mathbf{R}}_{\mathbf{C}X}$.

Therefore, in both cases, it is shown that $\hat{\mathbf{R}}_{\mathbf{C}}$ satisfies the conditions of the ASR from $\hat{\mathbf{S}}_{\mathbf{C}}$ to \hat{S}_{D} .



Masashi Mizoguchi received B.S. and M.S. degrees in 2015 and 2017 from Osaka University. His research interest includes the symbolic synthesis based on the abstraction.



Toshimitsu Ushio received B.S., M.S. and Ph.D degrees in 1980, 1982 and 1985, from Kobe University. He was a Research Assistant at the UC Berkeley in 1985. From 1986 to 1990, he was a Research Associate in Kobe University. He joined Osaka University in 1994, and is currently a Professor. His research interests include control of discrete event and hybrid systems and analysis of nonlinear systems. He is a member of SICE, ISCIE, and IEEE.