Automatica 50 (2014) 2108–2112

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Decentralized control of networked discrete event systems with communication delays[☆]

Shaolong Shu[a,1], Feng Lin[a,b]

[a] School of Electronics and Information Engineering, Tongji University, Shanghai, 201804, China

[b] Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA

ARTICLE INFO

Article history:
Received 29 September 2013
Accepted 12 April 2014
Available online 17 June 2014

Keywords:
Discrete event systems
Communication delays
Decentralized control
Supervisors
Networked systems
Automata

ABSTRACT

In many practical systems, supervisory control is not performed by one centralized supervisor, but by multiple local supervisors. When communication networks are used in such a system as the medium of information transmission, the communication channels between local supervisors and the system to be controlled will unavoidably result in communication delays. This paper investigates how to use these local supervisors to control the system in order to satisfy given specifications even under communication delays. The specifications are described by two languages: a minimal required language which specifies the minimal required performance that the supervised system must have and a maximal admissible language which specifies the maximal boundary that the supervised system must be in. The results show that if the control problem is solvable, then there exists the minimal control policy which can be calculated based on state estimates. Furthermore, we derive algorithms to check whether the control problem is solvable or not.

1. Introduction

As the medium of information transmission, communication networks have been widely used in control systems. Such communication networks have great technical advantages. On the other hand, using communication networks unavoidably introduces communication delays in communication channels between the supervisor and the plant. Since the conventional supervisory control theory (Cassandras & Lafortune, 1999; Lin & Wonham, 1988; Ramadge & Wonham, 1987; Shu, Lin, & Ying, 2007) for discrete event systems assumes that there are no communication delays between the supervisor and the plant, the supervised system designed using the conventional supervisory control theory will lead to performance degradations and even failures when used in networked systems. Recently we investigate the control problem of networked discrete event systems with communication delays (Shu & Lin, 2013). We assume there are communication delays in the observation channel via which the information of the system is sent to the supervisor and in the control channel via which the control commands issued by the supervisor is sent to control the system. Due to communication delays, the control may be different for the same event sequence. Hence two types of controlled languages are defined: a small language and a large language. Both of them are very useful. With these considerations, we successfully solve the centralized control problem of networked discrete event systems.

In the paper Shu and Lin (2013), we assume there is only one centralized supervisor to control the system. However, in practical engineering systems, the control objectives are often achieved using multiple local supervisors in a decentralized fashion. For decentralized control, every local supervisor has its own communication channels with communication delays. Different communication channels may have different communication delays. The control problem needs to consider these different communication delays in different communication channels. Hence the decentralized control problem of networked discrete event systems is more complex.

In the paper, we investigate the decentralized control problem of networked discrete event systems. Our goal is to find control policies for all local supervisors in order to satisfy given specifications. As discussed in our previous paper Shu and Lin (2013),

☆ This research is supported in part by the National Natural Science Foundation of China under Grants 60904019, 61143006 and 71071116. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Jan Komenda under the direction of Editor Ian R. Petersen.

E-mail addresses: shushaolong@tongji.edu.cn (S. Shu), flin@ece.eng.wayne.edu (F. Lin).

[1] Tel.: +86 2169589873.

http://dx.doi.org/10.1016/j.automatica.2014.05.035
0005-1098/

the specifications are described by two languages: a minimal required language and a maximal admissible language. The minimal required language specifies the minimal required performance that the supervised system must have. The maximal admissible language specifies the maximal boundary that the supervised system must be in. In order to deal with the two specifications, we define two languages for the supervised system: a small language and a large language. The decentralized control problem for networked discrete event systems is to find a decentralized control policy such that the small language is larger than the minimal required language and the large language is smaller than the maximal admissible language. Furthermore, because some events are uncontrollable and some events are unobservable, the control policy implemented by the local supervisors must be co-observation feasible and co-control feasible. We show that if the decentralized control problem is solvable, then there exists a minimal control policy for every local supervisor. Furthermore, we investigate how to check whether the control problem is solvable or not. The approach is to construct an automaton of which the language is equal to the large language of the supervised system.

Some papers have been published in the literature that study communication delays in discrete event systems. Debouk, Lafortune, and Teneketzis (2003), Qiu and Kumar (2008) and Takai and Kumar (2012) consider how to diagnose faults for discrete event systems when there are communication delays among local diagnosors. Liu and Lin (2009) and Tripakis (2004) consider the control problem of discrete event systems with communication delays among local supervisors. The work investigating the communication delays between the supervisor and the plant is done by Balemi (1994), Lin (2012), Park and Cho (2006), Park and Cho (2007b) and Shu and Lin (2013). The above papers assume that there is only one centralized supervisor. Park and Cho (2007a) discusses how to design a set of local supervisors when there are communication delays between these local supervisors and the supervised system. However, our control mechanism is rather different than that used in Park and Cho (2007a). In Park and Cho (2007a), it is assumed that no (controllable) event will be enabled until the delayed control decision (command) arrives. In some applications, this assumption is not acceptable because it will slow down the system significantly. It is also assumed in Park and Cho (2007a) that all controllable events are observable, which may or may not be reasonable, depending on the applications. We do not make these assumptions in this paper.

Due to the space limitation, all the proofs are omitted. They can be obtained from the authors.

2. Background

In this paper, a discrete event system to be controlled (plant) is modeled by an automaton as follows (Cassandras & Lafortune, 1999).

$$G = (Q, \Sigma, \delta, q_0),$$

where Q is the set of (finite) discrete states; Σ is the set of (finite) discrete events; and $\delta : Q \times \Sigma \to Q$ is the transition function. The transition function describes the dynamics of the discrete event system. The transition function is extended to $\delta : Q \times \Sigma^* \to Q$ in the usual way. We use $\delta(q, s)!$ to mean that $\delta(q, s)$ is defined. q_0 is the initial state. The language of automaton G is denoted as $L(G) = \{s \in \Sigma^* : \delta(q_0, s)!\}$. The set of all events that are defined in state q is denoted by $\Gamma(q)$. For a set Q', we use $|Q'|$ to denote the number of elements in Q'. For a string s, we use $|s|$ to denote the length of s. We assume some events are controllable. Hence $\Sigma = \Sigma_c \dot{\cup} \Sigma_{uc}$, where Σ_c is the controllable event set and Σ_{uc} is the uncontrollable event set. Similarly $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, where Σ_o is the observable event set and Σ_{uo} is the unobservable event set.

For string s, the observation of the supervisor is described by the natural projection $P : \Sigma^* \to \Sigma_o^*$ as

$$P(\varepsilon) = \varepsilon \qquad P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in \Sigma_o \\ P(s) & \text{if } \sigma \notin \Sigma_o \end{cases}$$

where ε denotes the empty string.

For string $s = \sigma_1\sigma_2 \cdots \sigma_n$, we denote the prefix of s with the last m events removed by $s_{-m} = \sigma_1\sigma_2 \cdots \sigma_{n-m}$; and the suffix of s with the last m events by $s_m = \sigma_{n-m+1}\sigma_{n-m+2} \cdots \sigma_n$. Hence $s = s_{-m}s_m$. If $n < m$, then $s_{-m} = \varepsilon$ and $s_m = s$.

Now let us briefly review the results on centralized control of networked discrete event systems discussed in Shu and Lin (2013). We allow communication delays in both observation channel and control channel between the supervisor and the plant. We call communication delays in the observation channel observation delays and communication delays in the control channel control delays. Observation delays and control delays are random and upper bounded. N_o is the upper bound of observation delays and N_c is the upper bound of control delays. We assume that in the observation channel, communication delays do not change the order of the events, that is, if event α occurs before event β in the system, then α is observed by the supervisor before β. We also assume that in the control channel, the initial control policy is not delayed. The assumption ensures the system have a control initially. The system will then use the latest control it receives. Thus, a networked discrete event system is denoted by

$$\hat{G} = (Q, \Sigma, \delta, q_o, N_o, N_c).$$

When event sequence s occurs in the plant, the supervisor may have different observations due to uncertainties in observation delays. The set of possible observations is denoted by $\Theta(s) = \{P(t) : (\exists m \le N_o)t = s_{-m}\}$. The control policy is based on the current observation. Hence, control policy π with communication delays is a mapping $\pi : \Sigma^* \times \Sigma_o^* \to 2^{\Sigma}$. $\pi(s, \theta(s))$ is the set of events enabled by the supervisor when string s occurs and the supervisor sees $\theta(s)$. Note that $\theta(s) \in \Theta(s)$.

Given two control policies π and π', we say π is smaller (that is, more restrictive) than π', denoted by $\pi \le \pi'$ if $(\forall(s, \theta(s)) \in \Sigma^* \times \Sigma_o^*)\pi(s, \theta(s)) \subseteq \pi'(s, \theta(s))$. We say π is strictly smaller than π', denoted by $\pi < \pi'$ if $\pi \le \pi' \wedge (\exists(s, \theta(s)) \in \Sigma^* \times \Sigma_o^*)\pi(s, \theta(s)) \subset \pi'(s, \theta(s))$.

Given two control policies π_1 and π_2. We define their conjunction, denoted by $\pi_1 \wedge \pi_2$ as follows. For all $(s, \theta(s)) \in \Sigma^* \times \Sigma_o^*$, $(\pi_1 \wedge \pi_2)(s, \theta(s)) = \pi_1(s, \theta(s)) \cap \pi_2(s, \theta(s))$.

To ensure that a control policy can be implemented, we require that it is control feasible and observation feasible. The control feasibility means $(\forall(s, \theta(s)) \in \Sigma^* \times \Sigma_o^*)\Sigma_{uc} \subseteq \pi(s, \theta(s))$. The observation feasibility means $(\forall(s, \theta(s)), (s', \theta(s')) \in \Sigma^* \times \Sigma_o^*)\theta(s) = \theta(s') \Rightarrow \pi(s, \theta(s)) = \pi(s', \theta(s'))$.

Since control action can be delayed for up to N_c steps, the control action in use now can be one of the control actions issued by the supervisor within the past N_c events. Because a supervisor may have different observations for the same string (event sequence) and may disable different events even for the same string, two languages for the supervised system $\pi/\hat{G}$ are defined. One is the small language. The other is the large language. Their definitions are as follows.

Definition 1. For a networked discrete event system $\hat{G} = (Q, \Sigma, \delta, q_0, N_o, N_c)$ controlled under control policy π, the small language $L_r(\pi/\hat{G})$ generated by the supervised system is defined recursively as follows.

$$\varepsilon \in L_r(\pi/\hat{G}),$$

$$s\sigma \in L_r(\pi/\hat{G}) \Leftrightarrow s \in L_r(\pi/\hat{G}) \wedge s\sigma \in L(\hat{G})$$
$$\wedge(\forall m \le N_c)(\forall\theta(s_{-m}) \in \Theta(s_{-m}))\sigma \in \pi(s_{-m}, \theta(s_{-m})).$$