Robustness of Synchronous Communication Protocols with Bounded Delay for Decentralized Discrete-Event Control

W.H. Sadid * **S.L. Ricker** ** **S. Hashtrudi-Zad** *

* *Dept. of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada (e-mail:m_sadid, shz@ece.concordia.ca)*
** *Dept. of Mathematics & Computer Science, Mount Allison University, Sackville, NB, Canada (e-mail: lricker@mta.ca)*

Abstract: Recent work on modeling communication delay between communicating decentralized discrete-event controllers considers only the case when all observations are communicated. When this condition is relaxed, it may yet be possible to formulate communicating decentralized controllers that can solve the control problem. Instead of synthesizing reduced communication protocols under bounded delay, synchronous communication protocols (where not all observations are communicated) are examined for their robustness under conditions when only the upper bound for the delay is known.

Keywords: discrete-event systems, control, communication.

1. INTRODUCTION

Various strategies exist for incorporating synchronous communication into decentralized discrete-event control (see van Schuppen (2004) for a summary). Further, minimal communication protocols have also been defined for this class of problems. In Rudie et al. (2003) optimality is defined with respect to set theoretic criteria (removing an element from the set of communications violates a solution property), whereas in Ricker (2008) and Sadid et al. (2010) a quantitative measure is imposed. In contrast, when communication occurs under delay (either bounded or unbounded) recent studies of communicating decentralized controllers (Tripakis (2004); Hiraishi (2009)), diagnosers (Qiu and Kumar (2008)) and prognosicators (Takai and Kumar (2009)) consider only communication protocols where every observation is communicated. When constructing any type of synchronous communication protocol, it is impractical to require that each decentralized controller communicates to other controllers after the occurrence of every observation. Such a scenario arises only if such a protocol is the sole outcome of a communication protocol construction procedure.

Communicating information about every observable event occurrence, even in the presence of a finite (bounded) delay, may not be necessary to construct correct control and communication policies for decentralized discrete-event control. We are interested in recycling strategies for synthesizing synchronous communication protocols by introducing a non-zero but finite delay. We assume that there is a bounded communication (FIFO) buffer. As detailed in Tripakis (2004), when the delay is unbounded and when all observations are communicated, the problem of decentralized control is undecidable, and this result holds even when only a subset of observations is communicated.

To reason about delay in communication, we take inspiration from the supervisory control framework for timed discrete-event systems (TDES) of Brandin and Wonham (1994). A TDES features an additional event, the tick of a global clock τ, to mark the passage of one unit of time. In addition to the observational issues that arise when communication is delayed, we are interested in the relative timing effect of the speed of the plant versus the communication delay. We begin with a communication protocol that is designed to solve the decentralized control and communication problem with zero delay. Then we want to verify if the protocol is robust enough to withstand timing delays associated with a more realistic communication network, as well as taking into account timing characteristics of the plant. For purposes of verifying the robustness of the synchronous communication protocol under delay, we introduce τ events in the plant to represent a clock cycle.

The paper is organized as follows. The next section contains terminology and notation of discrete-event systems and synchronous communication protocols. Section 3 introduces the idea of robustness of synchronous communication protocols under bounded delay. A structure with which to verify the robustness of synchronous communication protocol under bounded delay is defined in Section 4. Finally, we conclude with a brief discussion in Section 5.

2. BACKGROUND

We employ the supervisory control framework of Ramadge and Wonham (1987) to model discrete-event systems. The behavior of an uncontrolled system is described by a regular language L over an alphabet Σ and its equivalent finite automaton representation:

$$M_L = (Q, \Sigma, T, q_0),$$

where Q is a finite set of states; Σ is a finite alphabet of symbols; $T \subseteq Q \times \Sigma \times Q$ is the transition relation; and q_0 is the initial state.We will write $q_1 \xrightarrow{\sigma} q_2$ if there is a transition labelled σ from state q_1 to state q_2 (i.e., $(q_1, \sigma, q_2) \in T$). We can compose transitions and

write $q_1 \xrightarrow{\sigma_1\sigma_2\ldots\sigma_m} q_r$ if there is a sequence of events $\sigma_1\sigma_2\ldots\sigma_m \in \Sigma^*$ that begin in state q_1 and end in state q_r. The prefix closure of a language L is defined as $\overline{L} := \{s \in \Sigma^* | \exists s' \in \Sigma^* \text{ such that } ss' \in L\}$. If L is prefix-closed, then $L = \overline{L}$. We assume that all languages in this paper are prefix-closed.

In supervisory control, a supervisor or *controller* is designed such that the uncontrolled system, under the control decisions of the controller, performs only behavior in a given specification language $K \subseteq L$ (with a corresponding automaton M_K). Thus, a controller, upon detecting that a behavior in $L \setminus K$ is about to occur, will prevent or *disable* controllable events that take the system from K to $L \setminus K$.

We are interested in problems that require the synthesis of $n \geq 2$ controllers (Rudie and Wonham (1992)). Let $I = \{1, \ldots, n\}$ denote the index set for our controllers. Then we define observable and controllable events for $i \in I$: $\Sigma = \Sigma_{o,i} \uplus \Sigma_{uo,i}$ and $\Sigma = \Sigma_{c,i} \uplus \Sigma_{uc,i}$. Let $I_c(\sigma) = \{i \in I | \sigma \in \Sigma_{c,i}\}$ be the set of controllers that control event σ. It will be useful also to define the following subsets of T based on the partitions of Σ: $T_{o,i} = \{(q, \sigma, q') \in T \mid \sigma \in \Sigma_{o,i}\}$ and $T_{c,i} = \{(q, \sigma, q') \in T \mid \sigma \in \Sigma_{c,i}\}$.

A *natural projection* $\pi_i : \Sigma^* \to \Sigma_{o,i}^*$ is used to illustrate the partial view of the system behavior for a controller i. In other words, π_i removes the events not observable to controller i from a sequence $s \in \Sigma^*$. The *inverse* projection $\pi^{-1} : \Sigma_{o,i}^* \to 2^{\Sigma^*}$ is defined for $s' \in \Sigma_{o,i}^*$ as $\pi_i^{-1}(s') = \{u \in \Sigma^* \mid \pi_i(u) = s'\}$. We will use the notation $[\![s]\!]_i$ to refer to $\pi_i^{-1}[\pi_i(s)]$.

To synthesize controllers that will solve the control problem in the absence of communication between controllers, the specification $K \subseteq L$ has to be *controllable*, and *co-observable* (Rudie and Wonham (1992)). The specification K is *controllable* wrt L and Σ_{uc} if $\overline{K}\Sigma_{uc} \cap L \subseteq \overline{K}$. Then K is *co-observable* wrt L, $\Sigma_{o,i}$, and $\Sigma_{c,i}$ if $(\forall s \in \overline{K})(\forall \sigma \in \Sigma_c) s\sigma \in L \setminus \overline{K} \Rightarrow (\exists i \in I_c(\sigma))[\![s]\!]_i \sigma \cap \overline{K} = \emptyset$. When $I = \{1\}$, K is said to be *observable* (Lin and Wonham (1988)).

When K is not co-observable, but *is* observable, it is possible to synthesize synchronous communication protocols so that with the additional information provided through the content of received messages for controller i, denoted here by $\Sigma_i^? \subseteq \Sigma_o \setminus \Sigma_{o,i}$, all the correct control decisions can be taken. There are a variety of techniques for synthesizing synchronous communication protocols (see van Schuppen (2004) for an overview). Since our focus here is not the synthesis of such protocols, in the sequel, we will simply assume that we are given a synchronous communication protocol Φ that is derived from a set of messages that are sent by each controller, which we denote here by $T_i^! \subseteq T_{o,i}$, and instructs each controller whether or not to send a message to another controller.We define the content of the message as $\Sigma_i^! = \{\sigma | \exists (q, \sigma, q') \in T_i^!\}$.

Definition 1. Given $T_i^! = T_{i,1}^! \cup \ldots \cup T_{i,n}^!$, for all $i \in I$, we define a ***communication protocol*** *$\Phi_i = \langle \phi_{i,1}, \ldots, \phi_{i,n} \rangle$, with $\phi_{i,j} : L \to \Sigma_i^! \cup \{\varepsilon\}$ as follows. If $q_0 \xrightarrow{s} q' \xrightarrow{\sigma_m} q'' \in T$ then*

$$\phi_{i,j}(s\sigma_m) = \begin{cases} \sigma_m, & \text{if } (q', \sigma_m, q'') \in T_{i,j}^!; \\ \varepsilon, & \text{otherwise.} \end{cases}$$

We update the natural projection to include received messages as $\pi_i^? : (\Sigma \cup \Sigma_i^! \cup \Sigma_i^?)^* \to (\Sigma_{o,i} \cup \Sigma_i^?)^*$. Hence a communication protocol is *feasible* if $\pi_i^?(s_1\sigma_m) = \pi_i^?(s_2\sigma_m) \Rightarrow \phi_{i,j}(s_1\sigma_m) = \phi_{i,j}(s_2\sigma_m)$.

Note that with synchronous communication, the messages arrive without delay, and there is no need to distinguish a sent message from a received message. The protocol simply instructs the controller whether or not to issue a message after sequence $s\sigma_m$ occurs.

The basic idea for this class of problems is that we can synthesize $T_i^!$ (and thus $\Sigma_i^?$) such that K is co-observable wrt L, $\Sigma_{o,i} \cup \Sigma_i^?$, and $\Sigma_{c,i}$ (Ricker (2008)).

We illustrate this class of decentralized control problems with an example.

Example 1. We are given M_L and M_K as shown in Fig. 1. Let $n = 3$ and suppose that $\Sigma_{o,1} = \{\mathsf{a}_1\}$, $\Sigma_{o,2} = \{\mathsf{b}_1\}$, and $\Sigma_{o,3} = \{\mathsf{c}_1\}$. Further, let $I_c(\sigma) = \{1, 2, 3\}$. Let $s = \mathsf{b}_1\mathsf{c}_1\mathsf{a}_1$

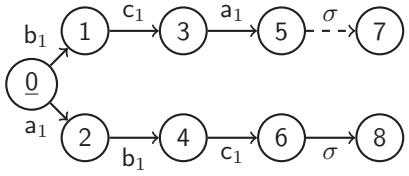

Fig. 1. A joint M_L (all transitions) and M_K (only solid line transitions); the initial state is underlined.

and $s' = \mathsf{a}_1\mathsf{b}_1\mathsf{c}_1$. Note that K is not co-observable wrt π_i since for all $i \in I_c(\sigma)$, $\pi_i(s) = \pi_i(s')$ and, thus, no single controller can take the correct control decision regarding σ. There are many different sets of sent messages that give rise to synchronous communication protocols that will solve this problem (short of communicating all observations). Here we choose $T_{1,3}^! = \{(3, \mathsf{a}_1, 5), (0, \mathsf{a}_1, 2)\}$ (i.e., $\Sigma_3^? = \{\mathsf{a}_1\}$) and $T_{2,1}^! = \{(0, \mathsf{b}_1, 1), (2, \mathsf{b}_1, 4)\}$ (i.e., $\Sigma_1^? = \{\mathsf{b}_1\}$) to be the only non-empty sets of messages. While $\pi_1(s) = \pi_1(s')$, by extending controller 1's information to $\Sigma_{o,1} \cup \Sigma_1^?$ via Φ, we have $\pi_1^?(s) = \mathsf{b}_1\mathsf{a}_1$ whereas now $\pi_1^?(s') = \mathsf{a}_1\mathsf{b}_1$, leading to K being co-observable wrt $\Sigma_{o,i} \cup \Sigma_i^?$. Similarly, controller 3 can also distinguish s from s' after receiving its messages. ◇

We are interested in determining how robust such a synchronous communication policy is in the face of an unknown but bounded delay in the communication channels. Note that we are not strictly interested in the robustness of optimal communication policies. Rather, we insist that the protocol allows the controllers to solve the control problem correctly; however, we are assuming that the protocols we are examining represent some sort of reduced communication from the "communicate every observation" strategy.

3. ROBUST SYNCHRONOUS COMMUNICATION UNDER BOUNDED DELAY

A synchronous communication protocol is *robust under a bounded delay* $[1..d]$ if messages received by controller i, even d clock cycles late, continue to allow the correct control decisions to be taken. Instead of synthesizing communication protocols that will solve the decentralized control problem in the presence of bounded delay, we are interested in determining how robust a synchronous

Download English Version:

https://daneshyari.com/en/article/714226

Download Persian Version:

https://daneshyari.com/article/714226

Daneshyari.com