Delay-Robustness for Localization-Based
Distributed Control of Timed Discrete-Event
Systems
Renyuan Zhang, Kai Cai, Yongmei Gan, W.M. Wonham

Abstract

This paper identifies properties of bounded and unbounded delay-robustness for distributed supervisory control for timed discrete-event systems (TDES) with communication delay. In our previous work, for untimed DES we have proposed an effective verification tool to identify delay-robustness for distributed controllers constructed by the supervisor localization procedure. Delay-robustness means that the overall system behavior controlled by distributed controllers with communication delay is equivalent to its delay-free counterpart. Further, determine delay-robustness to be bounded or unbounded by the standard controllability checking procedure. In this paper, we first apply the timed localization procedure to obtain a set of local controllers and tick preemptors; second, we model each inter-agent channel as a 2-state TDES in which the clock event tick is added to each state, and thus the time delay is represented by an exact number of ticks; third, we introduce the TDES delay-robustness. We distinguish bounded or unbounded delay-robustness of the system, and for those events identified as bounded delay-robust, we propose an algorithm to determine the maximal delay bound in terms of the number of tick events, rather than the number of untimed events that may occurred at random. Based on these studies, we present another algorithm to compute a set of maximal delay bounds for the communication events, under the condition that the overall system behavior with communication delay preserves the global optimality and nonblocking property. Our results are illustrated by the example of an up-load tap-changing transformer (ULTC).

Keywords

R. Zhang and Y. Gan are with School of Electrical Engineering, Xian Jiaotong University, China. K. Cai and W.M. Wonham are with the Systems Control Group, Department of Electrical and Computer Engineering, University of Toronto, Canada. (Emails: r.yuan.zhang@gmail.com; kai.cai@scg.utoronto.ca; ymgan@mail.xjtu.edu.cn; wonham@control.utoronto.ca).
Timed Discrete-Event Systems, Distributed Supervisory Control, Supervisor Localization, Delay-robustness.

I. INTRODUCTION

Distributed control is pervasive in engineering practice, either by geographical necessity or to circumvent the complexity of centralized control. In distributed control architecture of discrete-event systems (DES), component agents have their own local observation and control strategies, concurrently and collectively achieve a global given task (specification) through inter-agent communications. In practice, since distributed controllers are linked by a physical network subject to delays, to model and appraise these delays is essential for the correct implementation of control strategies.

For untimed DES, distributed supervisory control with communication delay has been dealt with widely in the DES literature\[1–8\]. In particular in \[2–6\], the existence of distributed controllers in the unbounded delay case is proved to be undecidable, and in \[2–5, 7\], distributed controllers are synthesized under the condition that communication delays are bounded. Recently in \[9\] and its conference precursor\[10\], we propose a new approach to deal with distributed supervisory control with communication delay: first we apply the supervisor localization procedure proposed in \[11, 12\] to decompose each synthesized supervisor into local controllers for individual plant components, in this process determining the set of communication events (i.e. events to be communicated); then we propose the concept of delay-robustness to identify that the overall controlled behavior of the system interconnected by communication channels is equivalent to its delay-free counterpart, and design a test to verify for which channeled events (i.e. selected communication events that are transmitted by channels) the system is delay-robust. In this approach, the communication channel for each channeled event is modeled as a 2-state DES; this channel requires that the fresh occurrence of the channeled event be permitted only when the channel is idle; thus for uncontrollable events the requirement may violate the modeling assumption that uncontrollable events cannot be delayed or disabled. Hence, an additional checking procedure (based on the standard controllability checking procedure) is presented to check if the assumption is violated, and thereby we determine delay-robustness to be bounded or unbounded. However, in untimed DES, the delay bound only can be measured by the number of untimed events, not an exact time; this disadvantage motivates our study on determining the delay bound by the exact number of clock events tick in TDES model.

In this paper, we study delay-robustness in the TDES in the Brandin-Wonham (BW) framework\[13, 14\]. Specifically, we follow the approach described above for untimed DES to investigate the distributed supervisory control of TDES with communication delay. First, we apply the timed supervisor localization
procedure proposed recently in [15]. This procedure decomposes each synthesized supervisor into a set of local controllers for prohibiting controllable events and a set of local preemptors (formally defined in [15]) for preempting the clock event \textit{tick}. In this process, a set of communication events is determined. Second, we build a communication channel modelled by a 2-state TDES by augmenting the 2-state DES model in [9] with the clock event \textit{tick}, and adopt the test tool presented in [9] to verify delay-robustness for the channeled events. Third, we distinguish bounded or unbounded delay robustness of the system, and for the latter we develop an algorithm to compute the maximal delay bound (in terms of the number of \textit{tick} events) of each channeled event.

The first contribution of this paper is the adaption of the delay-robustness concept and verification procedure from untimed DES to TDES. Specific to TDES, we introduce the 2-state TDES channel model to transmit each channeled event, in which the delay time is represented by an exact number of \textit{ticks}. The second contribution is the development of an algorithm to compute the maximal delay bound for events that are bounded delay-robust. In the TDES model, the maximal delay bound is exactly measured by the number of clock events \textit{tick}, rather than the arbitrarily occurred untimed event in untimed DES model. Based on these studies, we present another algorithm to compute a set of maximal delay bounds for the communication events, under the condition that the overall system behavior is globally optimal and nonblocking. These results are illustrated by a ULTC (up-load tap-changing transformer) example.

The paper is organized as follows. Sect. II provides a review of the BW TDES framework and briefly introduces the supervisor localization procedure for TDES. In Sect. III we first present the concept and verification procedure for TDES delay-robustness, and then we distinguish unbounded and bounded delay-robustness by the standard controllability checking procedure. For those events identified to be bounded delay-robust, we present in Sect. IV an algorithm to compute the maximal delay bound. In Sect. V we present another algorithm to compute a set of maximal delay bounds, preserving the global optimality and nonblocking of the overall system. Sect. VI studies a distributed control solution for the voltage control problem for a ULTC, and finally Sect. VII presents our conclusions.

II. DISTRIBUTED CONTROL BY SUPERVISOR LOCALIZATION OF TDES

A. Preliminaries on TDES

The TDES model proposed by Brandin and Wonham [13, 14], is evolved from the untimed DES model ‘generator’ in the RW framework. A TDES is given by

$$G := (Q, \Sigma, \delta, q_0, Q_m).$$

(1)
Here \( Q \) is the finite set of \emph{states}; \( \Sigma \) is the finite set of events including a distinguished event \emph{tick}, which represents the “tick of the global clock”; \( \delta : Q \times \Sigma \to Q \) is the (partial) \emph{state transition function}; this is derived from the corresponding activity transition function; the reader is referred to detailed updating rules given in \cite{13, 14}; \( q_0 \) is the \emph{initial state}; and \( Q_m \subseteq Q \) is the set of \emph{marker states}. Note that the event set \( \Sigma \) is comprised of \( \Sigma_{act} \) and the clock event \emph{tick}, where \( \Sigma_{act} \) is a set of timed event. Each timed event \( \sigma \in \Sigma_{act} \) is assigned a lower bound \( l_\sigma \in \mathbb{N} \) (\( \mathbb{N} \) denotes the natural numbers) and an upper bound \( u_\sigma \in \mathbb{N} \) such that \( l_\sigma \leq u_\sigma \).

Let \( \Sigma^* \) be the set of all finite strings of elements in \( \Sigma = \Sigma_{act} \cup \{\text{tick}\} \), including the empty string \( \epsilon \). For \( \Sigma' \subseteq \Sigma \), the \emph{natural projection} \( P : \Sigma^* \to \Sigma'^* \) is defined by

\[
P(\epsilon) = \epsilon, \quad \epsilon \text{ is the empty string;}
\[
P(\sigma) = \begin{cases} 
\epsilon, & \text{if } \sigma \notin \Sigma', \\
\sigma, & \text{if } \sigma \in \Sigma';
\end{cases}
\]

\[
P(s\sigma) = P(s)P(\sigma), \quad s \in \Sigma^*, \sigma \in \Sigma.
\]

As usual, \( P \) is extended to \( P : Pwr(\Sigma^*) \to Pwr(\Sigma'^*) \), where \( Pwr(\cdot) \) denotes powerset. Write \( P^{-1} : Pwr(\Sigma'^*) \to Pwr(\Sigma^*) \) for the \emph{inverse-image function} of \( P \).

We introduce the languages generated by TDES \( G \) in \cite{1}. The transition function is extended to \( \delta : Q \times \Sigma^* \to Q \) in the usual way. The \emph{closed behavior} of \( G \) is the language \( L(G) := \{ s \in \Sigma^* | \delta(q_0, s) \} \) and the \emph{marked behavior} is \( L_m(G) := \{ s \in L(G) | \delta(q_0, s) \in Q_m \} \subseteq L(G) \). We say that \( G \) is \emph{nonblocking} if the \emph{prefix closure} \( \bar{L}_m(G) = L(G) \).

To use TDES \( G \) in \cite{1} for supervisory control, we first define a subset of events, denoted by \( \Sigma_{hib} \), as the \emph{prohibitible} events which can be disabled by an external supervisor. Next, and specific to TDES, we bring in another category of events, called the \emph{forcible} events, which can \emph{preempt} event \emph{tick}; let \( \Sigma_{for} \subseteq \Sigma_{act} \) denote the set of forcible events. Now it is convenient to define the \emph{controllable} event set \( \Sigma_c := \Sigma_{hib} \cup \{\text{tick}\} \). The \emph{uncontrollable} event set is \( \Sigma_u := \Sigma - \Sigma_c \).

We introduce the notion of controllability as follows. For a string \( s \in L(G) \), define \( Elig_G(s) := \{ \sigma \in \Sigma | s\sigma \in L(G) \} \) to be the subset of events ‘eligible’ to occur (i.e. defined) at the state \( q = \delta(q_0, s) \).

Consider an arbitrary language \( F \subseteq L(G) \) and a string \( s \in F \); similarly define the eligible event subset
Elig_F(s) := \{ \sigma \in \Sigma | s_\sigma \in F \}. We say F is controllable with respect to G if, for all s \in \bar{F},

\[
Elig_F(s) \supseteq \begin{cases} 
Elig_G(s) \cap (\Sigma_u \cup \{\text{tick}\}) & \text{if } Elig_F(s) \cap \Sigma_for = \emptyset, \\
Elig_G(s) \cap \Sigma_u & \text{if } Elig_F(s) \cap \Sigma_for \neq \emptyset.
\end{cases}
\] (3)

Whether or not F is controllable, we denote by C(F) the set of all controllable sublanguages of F. Then C(F) is nonempty, closed under arbitrary set unions, and thus contains a unique element denoted by supC(F) \[13, 14\]. Now consider a specification language E \subseteq \Sigma^* imposed on the timed behavior of G; E may represent a logical and/or temporal requirement. Let the TDES

\[
\text{SUP} = (X, \Sigma, \xi, x_0, X_m)
\] (4)

be the corresponding monolithic supervisor that is optimal (i.e., maximally permissive) and nonblocking in the following sense: SUP’s marked language L_m(SUP) is

\[
L_m(SUP) = supC(E \cap L_m(G)) \subseteq L_m(G)
\]

and moreover its closed language L(SUP) is L(SUP) = \bar{L}_m(SUP).

B. Supervisor Localization of TDES

In this subsection, we introduce the supervisory localization procedure, which was initially proposed in DES framework \[11\] and then adapted to TDES framework \[15\]. By this procedure, a set of local controllers and preempts is obtained and shown to be control equivalent to the monolithic supervisor SUP in (4). By allocating these constructed local controllers and preempts to each component agent, we build a distributed supervisory control architecture.

Let TDES G in (1) be the plant to be controlled and E be a specification language. Synthesize as in \[14\] the monolithic optimal and nonblocking supervisor SUP. Supervisor SUP’s control action includes (i) disabling prohibitable events in \Sigma_{hib} and (ii) preempting tick via forcible events in \Sigma_{for}.

By the supervisor localization procedure, a set of local controllers \{LOC_C^\alpha | \alpha \in \Sigma_{hib} \} (with alphabet \Sigma_\alpha \subseteq \Sigma) one for disabling each prohibitive event and a set of local preempts \{LOC_P^\beta | \beta \in \Sigma_{for} \} (with alphabet \Sigma_\beta \subseteq \Sigma) one for preempting tick via each forcible event \beta are constructed; these are

\[1\] For the detailed definition of ‘local controller’ and ‘local preemt’, the reader is referred to \[15\].

5
control equivalent to $\text{SUP}$ (with respect to $G$) in the following sense:

$$L(G) \cap \left( \bigcap_{\alpha \in \Sigma_{j,for}} P_{\alpha}^{-1} L(\text{LOC}_{\alpha}^P) \right)$$
$$\cap \left( \bigcap_{\beta \in \Sigma_{h,ib}} P_{\beta}^{-1} L(\text{LOC}_{\beta}^C) \right) = L(\text{SUP}),$$

(5)

$$L_m(G) \cap \left( \bigcap_{\alpha \in \Sigma_{j,for}} P_{\alpha}^{-1} L_m(\text{LOC}_{\alpha}^P) \right)$$
$$\cap \left( \bigcap_{\beta \in \Sigma_{h,ib}} P_{\beta}^{-1} L_m(\text{LOC}_{\beta}^C) \right) = L_m(\text{SUP}).$$

(6)

Here $P_{\alpha} : \Sigma^* \to \Sigma_{\alpha}^*$ and $P_{\beta} : \Sigma^* \to \Sigma_{\beta}^*$ are the natural projections as in (2).

Now, using the constructed local controllers and local preemptors, we build a distributed supervisory control architecture without communication delay for a multi-agent TDES plant. Consider that $G$ consists of $n$ ($n \in \mathbb{N}$) components $G_i$ ($i \in N := \{1, 2, \ldots, n\}$) with event set $\Sigma_i$ where $G_i$ are a priori independent in the sense that their corresponding timed event sets $\Sigma_{i,act}$ are pairwise disjoint, i.e. $\Sigma_{i,act} \cap \Sigma_{j,act} = \emptyset$ ($i \neq j$). A convenient allocation policy is described as follows. For fixed $G_i$, let $\Sigma_{i,for}$, $\Sigma_{i,hib} \subseteq \Sigma_i$ be its corresponding forcible event set and prohibitable event set, respectively. As exemplified in Fig. [1] let each controller/preemptor be owned by its corresponding agent $G_i$, namely $G_i$ is controlled by a set of local preemptors $\text{LOC}^P_i := \{\text{LOC}^P_{\alpha} | \alpha \in \Sigma_{i,for}\}$ and a set of local controllers $\text{LOC}^C_i := \{\text{LOC}^C_{\beta} | \beta \in \Sigma_{i,hib}\}$. This allocation creates a distributed control architecture for the multi-agent plant, in which each agent acts semi-autonomously while interacting with other agents through communication of shared events. For agent $G_i$, the set of communication events that are transmitted from other agents is

$$\Sigma_{i,com} := \left( \bigcup_{\alpha \in \Sigma_{i,for}} \Sigma_{\alpha} - \Sigma_i \right) \cap \left( \bigcup_{\beta \in \Sigma_{i,hib}} \Sigma_{\beta} - \Sigma_i \right)$$

(7)

where $\Sigma_{\alpha}$ and $\Sigma_{\beta}$ are the event sets of $\text{LOC}^P_{\alpha}$ and of $\text{LOC}^C_{\beta}$ respectively.

However, this distributed control architecture is built under the assumption that the communication delays of communicating events (in $\Sigma_{i,com}$) are negligible. While simplifying the design of distributed controllers, this assumption may be unrealistic in practice, where controllers are linked by a physical network subject to delays. In the rest of this paper, we investigate how the communication delays affect the control strategies and thus the overall system behavior.

---

2The closed behavior and marked behavior of $G$ are $L(G) = \|\{L(G_i) | i \in N\}$ and $L_m(G) = \|\{L_m(G_i) | i \in N\}$ where $\|$ denotes synchronous product of languages[14].
III. DELAY-ROBUSTNESS FOR DISTRIBUTED CONTROL OF TDES

In this section, we extend delay-robustness concept and verification procedure presented in [9] to TDES. First, we build a 2-state TDES channel model for each channeled event selected from (7). Second, we adopt the test tool presented in [9] to verify delay-robustness for the channeled event. Finally, we determine delay-robustness to be bounded or unbounded by the standard controllability checking procedure.

A. Definition and Verification of Delay-Robustness in TDES

In this subsection, we define delay-robustness for distributed control of TDES and exploit the test tool presented in [9] to verify the delay-robustness.

First, we compute the local controlled behavior of an agent, for instance $G_i$. We call the agent $G_i$ controlled by its corresponding controllers/preemptors a module, written $\text{SUP}_i$; according to the described allocation policy,

$$\text{SUP}_i = G_i \parallel \left( \bigparallel_{\alpha \in \Sigma_{i,for}} \text{LOC}_{\alpha}^P \right) \parallel \left( \bigparallel_{\beta \in \Sigma_{i,hib}} \text{LOC}_{\beta}^C \right) \quad (8)$$

Here `||` denotes the synchronous product of two TDESs in the following sense: the closed and marked behaviors of $\text{TDES} = \text{TDES}_1 || \text{TDES}_2$ are $L(\text{TDES}) = L(\text{TDES}_1) || L(\text{TDES}_2)$ and $L_m(\text{TDES}) = L_m(\text{TDES}_1) || L_m(\text{TDES}_2)$. Under the condition that the communication delay among modules are
negligible, the overall system behavior is represented by \( \bigoplus_{i \in N}^{i} \). According to (5) and (6),

\[
\sup \cong \bigoplus_{i \in N}^{i} \sup
\]

where ‘\( \cong \)’ denotes that two DESs \( TDES1 \) and \( TDES2 \) have the identical marked and closed behavior, i.e. \( L(TDES1) = L(TDES2) \) and \( L_m(TDES1) = L_m(TDES2) \) both hold. So we use \( \sup \) to represent the overall behavior of the system \( G \) controlled by the local controller/preemptors without communication delay.

Next, we model the way that selected communication events are transmitted with indefinite time delays, and call such events *channeled events*. For instance, let \( \sigma \in \Sigma_j \) and assume that \( \sigma \) is needed to transmit from \( G_j \) to \( \sup_i \) (\( i \neq j \)). Denoted the set of events communicated from \( G_j \) to \( \sup_i \) by \( \Sigma_{j,\text{com},i} \), then\( \Sigma_{j,\text{com},i} = \Sigma_j \cap \Sigma_{\text{com},i} \), (10) and thus \( \sigma \in \Sigma_{j,\text{com},i} \). We model the communication channel for \( \sigma \) as a 2-state TDES \( CH(j, \sigma, i) \) shown in Fig. 2 and thus \( \sigma \) is a channeled event. Compared with the 2-state DES model shown in [9, Fig. 1], \( CH(j, \sigma, i) \) has event \( \text{tick-selfloop} \) being added to each state, and thus the alphabet of \( CH(j, \sigma, i) \) is \( \{ \sigma, \text{tick}, \sigma' \} \). We emphasize that, in the TDES model, the delay time can be measured by the number of \( \text{ticks} \), a more precise measure than the (perhaps random) number of events that might occur in the untimed DES model. We can thus determine an exact delay bound for bounded delay-robustness, as will be discussed in Sect. [IV].

In the transition structures of local controllers \( LOC_i^C \) and local preemptors \( LOC_i^P \) controlling \( G_i \), we replace every instance of event \( \sigma \) with event \( \sigma' \) (we call \( \sigma' \) the *signal event* of \( \sigma \)); let \( LOC_i^{C'} \) and \( LOC_i^{P'} \) represent these newly modified controllers and preemptors respectively. Now the local behavior of \( G_i \) is represented by

\[
\sup'_i = G_i \bigoplus_{\alpha \in \Sigma, \sigma_r}^{\alpha} \bigoplus_{\beta \in \Sigma_{i,hib}}^{\beta} LOC_i^{P'} \bigoplus_{\alpha \in \Sigma_{i,hib}}^{\alpha} LOC_i^{C'}
\]

(11)

Note that the capacity of \( CH(j, \sigma, i) \) is 1\(^4\). So only when the last occurrence of event \( \sigma \) is received by its recipient, will the channel send a fresh instance of \( \sigma \). Hence, the channel model permits the attendance of communication event \( \sigma \) to change to \( \sigma' \), which is how the local controller or local preemptor can be updated. The new state of \( \sigma \) is received from its recipient (i.e. \( \sup_i \)). The capacity of a buffer storing a communication event can be enlarged to any natural number. In that case, the delay-robustness concept and verification procedure are also applicable; however, the maximal communication delay tolerated by the system to preserve delay-robustness may be smaller.

\[^{3}\text{Specifically, if } TDES1 \text{ and } TDES2 \text{ are both state minimal, } TDES1 \cong TDES2 \text{ iff } TDES1 \text{ and } TDES2 \text{ are isomorphic} [14].\]

\[^{4}\text{In this paper, the bound of time delay indicates the maximal number of } \text{tick} \text{ events in the transmitting process, not the capacity of a buffer storing a communication event. Also, the capacity can be enlarged to any natural number. In that case, the delay-robustness concept and verification procedure are also applicable; however, the maximal communication delay tolerated by the system to preserve delay-robustness may be smaller.}\]
Fig. 2. Communication channel model $\text{CH}(j, \sigma, i)$ transmitting event $\sigma$ from $G_j$ to $\text{SUP}_i$ with indefinite time delay. The label $\sigma'$ represents that the occurrence of $\sigma$ has been received by the recipient. The number of events $\text{tick}$ that have occurred between $\sigma$ and $\sigma'$ represents the exact communication time delay.

reoccurrence of $\sigma$ only when the channel is idle. If $\sigma$ is controllable, its reoccurrence can be disabled or delayed by its corresponding local controller. However, if $\sigma$ is uncontrollable, this requirement may violate the modeling assumption that an uncontrollable event cannot be disabled or delayed by external agent. In this semantics, the only external agent added is the channel model, and thus the violation means that the occurrence of the uncontrollable event is blocked by its channel. Thus, the problem arises of checking whether or not such blocking occurs; we will address this issue in Sect. III-B.

Now, we compute the overall controlled behavior of system with communication delay of $\sigma$, and formalize its delay-robustness property. Here $\text{CH}(i, \sigma, j)$ is not considered as a control device, but rather an intrinsic component of the physical system being modeled. Thus it will be ‘hard-wired’ into the system by synchronous product with the agents and controllers/preemptors, and the overall controlled behavior of $G$ is (for $i, j, \sigma$ fixed)

$$
\text{SUP}' = \text{SUP}'_i \parallel \text{CH}(j, \sigma, i) \parallel \bigwedge_{k \in \mathbb{N}, k \neq i} \text{SUP}_k
$$

(12)

where $\text{SUP}'_i$ is defined in (11). $\text{SUP}'$ is defined over the alphabet $\Sigma' = \Sigma \cup \{\sigma'\}$. We refer to $\text{SUP}'$ as the channeled behavior of $\text{SUP}$ with $\sigma$ being the channeled event.

As defined in [9], delay-robustness says that $\text{SUP}'$ has the same behavior as $\text{SUP}$, namely, when observed through the natural projection $P : \Sigma^* \rightarrow \Sigma^*$, $\text{SUP}'$ is correct and complete. Here correctness means that no $P$-projection of anything $\text{SUP}'$ can do is disallowed by $\text{SUP}$, and completeness means that anything $\text{SUP}$ can do is the $P$-projection of something $\text{SUP}'$ can do. In terms of language, delay-robustness is defined as follows.

**Definition 1.** For $\text{SUP}'$ given in (12) and channeled event $\sigma$ selected from $\Sigma_{j,\text{com},i}$ (in (10)), $\text{SUP}$ in
is delay-robust relative to $\sigma$ if the following three conditions all hold,

$$PL(SUP') = L(SUP)$$  \hspace{1cm} (13)

$$PL_m(SUP') = L_m(SUP)$$  \hspace{1cm} (14)

$$(\forall s \in \Sigma^*)(\forall w \in \Sigma^*) \; s \in L(SUP') \; \& \; (Ps)w \in L_m(SUP)$$

$$\Rightarrow (\exists v \in \Sigma^*) \; P_L = w \; \& \; sv \in L_m(SUP')$$  \hspace{1cm} (15)

In this definition, conditions (13) and (14) describe the equivalence of closed and marked behaviors between SUP and the $P$-projection of SUP'. Condition (15), which is required for completeness, states that if SUP' executes a string $s$ whose projection $Ps$ in SUP can be extended by a string $w$ to a marked string of SUP, then SUP' can further execute a string $v$ whose projection $Pv$ is $w$ and such that $sv$ is marked in SUP'. Roughly, an observationally consistent inference about coreachability at the “operating” level of SUP' can be drawn from coreachability at the abstract (projected) level of SUP.

The following proposition, adapted from [9, Theorem 1], verifies whether or not SUP is delay-robust relative to $\sigma$.

**Proposition 1.** SUP is delay-robust relative to $\sigma$ if and only if

$$QCSUP' = Supqc(SUP', Null[r'])$$  \hspace{1cm} (16)

is structurally deterministic and has the identical closed and marked behavior to SUP (i.e. $QCSUP' \cong SUP$).

Here the operator $Supqc(TDES, Null(\Sigma_{uo}))$ [14, Sect. 6.7], computes the ‘supermal quasi-congruence’ relation on a given TDES over the alphabet $\Sigma'$ and projects the TDES to QCTDES, the corresponding quotient TDES over $\Sigma^* = P(\Sigma'^*)$ where $\Sigma = \Sigma' - \Sigma_{uo}$.

Proposition 1 supplies a test tool using $Supqc$ to verify delay-robustness of a system. In the next subsection, we determine delay-robustness to be bounded or unbounded.

---

5Let QCDES = $(Z, \Sigma, \zeta, z_0, Z_m)$. QCDES is said to be structurally deterministic if, for all $z \in Z$ and $s \in \Sigma^*$, we have $\zeta(z, s) \neq \emptyset \Rightarrow |\zeta(z, s)| = 1$.

6We remark that the algorithm for $Supqc(TDES, \cdot)$ in [14, Sect. 6.7], can be estimated to have time complexity $O(kn^4)$ where $(k, n)$ is the (alphabet, state) size of TDES.
B. Bounded and Unbounded Delay-Robustness

If a channeled event $\sigma$ is controllable, and SUP is delay-robust relative to $\sigma$, then we say that SUP is unbounded delay-robust relative to $\sigma$. Otherwise, we say that SUP is bounded delay-robust relative to $\sigma$. So Proposition 1 is sufficient to determine bounded or unbounded delay-robustness of the system for a channeled controllable event.

However, if a channeled event $\sigma$ is uncontrollable, as mentioned earlier, its reoccurrence is possibly blocked by the communication channel, thereby violating the modeling assumption that an uncontrollable event cannot be disabled or delayed by any external agent. This subsection employs the standard controllability checking procedure to verify whether a channeled uncontrollable event is blocked by its channel (previously mentioned in Sect. III-A), and thereby we distinguish the corresponding bounded or unbounded delay-robustness of the system.

First, we define formally the case that an uncontrollable event is blocked by its channel. Consider a single channeled event $\sigma \in \Sigma_{j,com,i}$, and let NSUP represent the system behavior without the channel model $\text{CH}(j, \sigma, i)$, namely

$$\text{NSUP} = \text{SUP}'_i \parallel \bigoplus_{k \in \mathbb{N}, k \neq i} \text{SUP}_k$$

(17)

The alphabet of NSUP is $\Sigma' = \Sigma \cup \{\sigma'\}$. Let

$$P_{ch} : \Sigma'^* \to \{\sigma, \text{tick}, \sigma'\}^*$$

(18)

be the natural projection (as in (2)) that projects the overall system behavior on $\{\sigma, \text{tick}, \sigma'\}^*$ and $P_{ch}^{-1} : \text{Pwr}(\{\sigma, \text{tick}, \sigma'\}^*) \to \text{Pwr}(\Sigma'^*)$ be its inverse image function. By (12) and the associativity of synchronous product[14], we have

$$L(\text{SUP}') = L(\text{NSUP}) \cap P_{ch}^{-1} L(\text{CH}(j, \sigma, i))$$

(19)

In terms of language, for SUP' given in (12), NSUP in (17) and $\sigma \in \Sigma_u$, if there exists $s \in L(\text{SUP'})$ such that $s\sigma \in L(\text{NSUP})$, but $s\sigma \notin L(\text{SUP'})$, then we say that $\sigma$ is blocked by its communication channel $\text{CH}(j, \sigma, i)$.

To check whether or not $\sigma$ is blocked by $\text{CH}(j, \sigma, i)$, we check if $P_{ch}^{-1} L(\text{CH}(j, \sigma, i))$ is NSUP-controllable with respect to event $\sigma$, i.e.

$$P_{ch}^{-1} L(\text{CH}(j, \sigma, i)) \{\sigma\} \cap L(\text{NSUP}) \subseteq P_{ch}^{-1} L(\text{CH}(j, \sigma, i)).$$

For this, we employ the standard algorithm that checks controllability [14]; the algorithm has complexity $O(mn)$ where $m$ and $n$ represent the state numbers of $\text{CH}(j, \sigma, i)$ and NSUP, respectively. Here $m = 2$ and thus the complexity of controllability checking is $O(n)$. 
Now, by Proposition 1 and the standard controllability checking procedure, we identify unbounded delay-robustness relative to an uncontrollable event as follows.

**Definition 2.** For \( \text{SUP}' \) given in (12), \( \text{NSUP} \) in (17), and an uncontrollable channeled event \( \sigma \) selected from \( \Sigma_{j,com,i} \) (in (10)), if \( \text{SUP} \) is delay-robust relative to \( \sigma \), and \( P_{ch}^{-1} L(\text{CH}(j, \sigma, i)) \) is \( \text{NSUP} \)-controllable with respect to \( \sigma \), then we say that \( \text{SUP} \) is unbounded delay-robust relative to \( \sigma \).

If \( \text{SUP} \) is not unbounded delay-robust relative to \( \sigma \), then we say that \( \text{SUP} \) is bounded delay-robust relative to \( \sigma \). According to Definition 2, Proposition 1 and the controllability checking procedure together are sufficient to determine bounded or unbounded delay-robustness for a channeled uncontrollable event. It is of interest, in the case of bounded delay-robustness, to determine the maximal delay bound; we address this issue in the next section.

**IV. MAXIMAL DELAY BOUND FOR BOUNDED DELAY-ROBUSTNESS**

In untimed DES, the delay bound for preserving bounded delay-robustness is measured only by the number of arbitrarily occurred untimed event, but not an exact time. However, in TDES, we can exactly measure the time delay by counting the number of ticks. This section presents an algorithm to compute the maximal delay bound for an uncontrollable event such that the delay-robustness property and the controllability condition are both satisfied.

Before that, we define \( d \)-bounded delay-robustness relative to an uncontrollable event \( \sigma \) for a given \( d \in \mathbb{N} \). For this purpose, we introduce a TDES model \( \text{CH}_d(j, \sigma, i) \) shown in Fig. 3. In this model, the transmission of event \( \sigma \) must be completed before the \( d \)th tick occurs where \( d \in \mathbb{N}^+ \); we will discuss the case \( d = 0 \) separately.

Connected by \( \text{CH}_d(j, \sigma, i) \), the overall system behavior with \( d \)-bounded communication delay is
represented by

\[ \text{SUP}'_d = \text{SUP}'_i \parallel \text{CH}_d(j, \sigma, i) \parallel (\parallel \text{SUP}_k) \quad (20) \]

The alphabet of \( \text{SUP}'_d \) is \( \Sigma' = \Sigma \cup \{\sigma'\} \). Note that here \( \text{SUP}'_i \) is defined in (11), and \( \sigma' \) is the signal event of \( \sigma \). By (20) and the association of synchronous product, we have

\[ \text{SUP}'_d = \text{NSUP}||\text{CH}_d(j, \sigma, i) \quad (21) \]

By Proposition 1 and the controllability checking procedure to \( \text{SUP}'_d \), we define \( d \)-bounded delay-robustness relative to an uncontrollable event as follows.

**Definition 3.** For \( \text{SUP}'_d \) given in (20) where \( d \in \mathbb{N}^+ \), \( \text{NSUP} \) in (17), and an uncontrollable channeled event \( \sigma \) selected from \( \Sigma_{j, \text{com}, i} \) (in (10)), if \( \text{SUP} \) is delay-robust relative to \( \sigma \) (verified by Proposition 1 with \( \text{SUP}'_i \) being replaced by \( \text{SUP}'_d \)), and \( P_{ch}^{-1}L(\text{CH}_d(j, \sigma, i)) \) is \( \text{NSUP} \)-controllable with respect to \( \sigma \), then we say that \( \text{SUP} \) is \( d \)-bounded delay-robust relative to uncontrollable event \( \sigma \).

Next, we consider the case that \( d = 0 \). In this case, the transmission of \( \sigma \) must be completed instantaneously, i.e. the communication delay is negligible. So the overall system behavior is identical to \( \text{SUP} \) as shown in (9), and we say that \( \text{SUP} \) is \( 0 \)-bounded delay-robust with respect to \( \sigma \). Obviously, \( \text{SUP} \) is \( 0 \)-bounded delay-robust relative to any channeled event.

Now, we determine the maximal delay bound \( d \) such that bounded delay-robustness is preserved. Intuitively, the smaller the communication delay, the controllers are easier to implement correct control actions in time, and thus the bounded delay-robustness will be easier to preserve. So, there exists a maximal delay bound for bounded delay-robustness. To compute the maximal delay bound, we need the following theorem.

**Theorem 1.** For any \( d, k \in \mathbb{N} \), if \( \text{SUP} \) is \( d \)-bounded delay-robust relative to \( \sigma \) and \( k \leq d \), then it is \( k \)-bounded delay-robust relative to \( \sigma \).

This result can be obtained immediately from the following proposition; its proof is in Appendix A.

**Proposition 2.** For any \( d \in \mathbb{N} \), if \( \text{SUP} \) is \( (d+1) \)-bounded delay-robust relative to \( \sigma \), then it is \( d \)-bounded delay-robust relative to \( \sigma \).

By Theorem 1 if \( \text{SUP} \) is bounded delay-robust relative to \( \sigma \), then there exists a maximal delay bound, \( d_{max} \in \mathbb{N} \), such that \( \text{SUP} \) is \( d_{max} \)-bounded delay-robust, but not \( (d_{max} + 1) \)-bounded delay-robust. We
present the following algorithm by increasing the delay bound iteratively to compute the maximal delay bound, $d_{\text{max}} \in \mathbb{N}$.

**Algorithm 1.** Input $\text{SUP}$ in (4), $\text{NSUP}$ in (17) and uncontrollable $\sigma$ selected from $\Sigma_{j,\text{com},i}$ (in (10)).

1. Initially set $d = 1$.
2. Create TDES model $\text{CH}_d(j, \sigma, i)$, and compute the overall system behavior $\text{SUP}'_d = \text{NSUP} \parallel \text{CH}_d(j, \sigma, i)$.
3. For $\text{SUP}'_d$, $\text{NSUP}$ and $\sigma$, verify according to Definition 3 whether or not $\text{SUP}$ is $d$-bounded delay-robust relative to $\sigma$. If not, let $d = d - 1$ and go to Step 4; otherwise advance $d$ to $d + 1$ and $\text{NSUP}$ to $\text{SUP}'_d$, and go to Step 2.
4. Output $d_{\text{max}} = d \in \mathbb{N}$.

Note that in this algorithm, we start from $d = 1$, because $\text{SUP}$ is 0-bounded delay-robust relative to any selected $\sigma$.

Let $d_{\text{max}}$ be the output of Algorithm 1. $\text{SUP}$ is $d_{\text{max}}$-bounded delay-robust relative to $\sigma$, but not $(d_{\text{max}} + 1)$-bounded delay-robust.

The correctness of Algorithm 1 is confirmed by Theorem 1, i.e. $d_{\text{max}}$ will be uniquely determined by Algorithm 1. For the effectiveness, since when $d_{\text{max}} \in \mathbb{N}$, the algorithm returns at the $(d_{\text{max}} + 1)$th step, and thus Algorithm 1 will terminate in finite steps.

**Remark 1.** For a controllable channeled event, if the system is bounded delay-robust, Algorithm 1 is also applicable to compute the maximal delay bound. In this case the controllability checking procedure is not needed for verifying the $d$-bounded delay-robustness.

In conclusion, for any channeled event selected from $\Sigma_{j,\text{com},i}$, to preserve the global optimality and nonblocking of the system and not violate the modeling assumption on uncontrollable events, the transmission of this event must be completed either (1) in a bounded time (corresponding to bounded delay-robustness), or (2) in the unbounded time (corresponding to unbounded delay-robustness).

V. **DISTRIBUTED SUPERVISORY CONTROL ARCHITECTURE WITH COMMUNICATION DELAY**

In previous sections, the communication delay of each event (controllable or uncontrollable) is investigated separately, and an algorithm is presented to compute the maximal delay bound. However, if the communication delays of multiple events are considered simultaneously, the maximal delay bound of each event tolerated by the system (for preserving optimality and nonblocking) will be interfered each other. So in this section, we present an algorithm to compute a set of maximal delay bounds for the
communication events, under the condition that the overall system behavior preserves optimality and nonblocking.

Based on the study on delay-robustness for each channeled event, we compute the maximal delay bounds for all the communication events in this way: first, we consider the communication delay of one selected event, and by Algorithm 1 we compute its maximal delay bound; second, hard-wiring the corresponding channel into the system, we obtain a new system; third, consider the communication delay of another selected event, and compute its maximal delay bound respected to the new system. By repeating the second and third steps, a set of maximal delay bounds for all the communication events will be obtained. We formally present the algorithm as follows.

Write \( \Sigma_{j,\text{com},i} \) (in (10)) as \( \{ \sigma_1, \sigma_2, \ldots, \sigma_{m_{j,i}} \} \) where \( m_{j,i} = |\Sigma_{j,\text{com},i}| \). For each \( \sigma_k \) \( (k = 1, \ldots, m_{j,i}) \), let \( d_{\text{max}}' (j, \sigma_k, i) \) represent its maximal delay bound when the communication delays are considered separately.

\[ \text{Algorithm 2. Input } \text{SUP (in (4)) and } \Sigma_{\text{com}} = \bigcup_{i,j \in N} \Sigma_{j,\text{com},i}. \]

(i) Set \( i = 1, j = 1 \) and \( k = 1 \).

(ii) If \( k > m_{j,i} \), advance \( j \) to \( j + 1 \) and set \( k = 1 \). If \( j > n \), advance \( i \) to \( i + 1 \), and set \( j = k = 1 \). If \( i > n \), go to Step (iv). Apply Algorithm 1 with inputs \( \text{SUP}, \text{NSUP} \) and \( \sigma_k \) to obtain its maximal delay bound \( d_{\text{max}}' (j, \sigma_k, i) \).

(iii) If \( d_{\text{max}}' (j, \sigma_k, i) \in \mathbb{N}^+ \), hard-wire \( \text{CH}_{d_{\text{max}}' (j, \sigma_k, i)} \) (where \( d_{\text{max}}' \) represents \( d_{\text{max}} (j, \sigma_k, i) \)), and advance \( \text{NSUP} \) to \( \text{NSUP} || \text{CH}_{d_{\text{max}}' (j, \sigma_k, i)} \). If \( d_{\text{max}}' (j, \sigma_k, i) = \infty \), advance \( \text{NSUP} \) to \( \text{NSUP} || \text{CH} (j, \sigma_k, i) \).

If \( d_{\text{max}}' (j, \sigma_k, i) = 0 \), no channel is hard-wired into the system. Now advance \( k \) to \( k + 1 \) and go to Step (ii).

(iv) Output a set of maximal delay bounds \( \{ d_{\text{max}}' (j, \sigma_k, i) \mid \sigma_k \in \Sigma_{j,\text{com},i}, i, j \in N \} \).

By this algorithm, we will obtain a set of maximal delay bounds \( d_{\text{max}}' (j, \sigma, i) \in \mathbb{N} \cup \{ \infty \} \), one for each \( \sigma \in \Sigma_{j,\text{com},i} \) (transmitted from \( G_j \) to \( \text{SUP}_i \)). This result is not optimal in the meaning of achieving the maximal delay when each event is considered separately, namely \( d_{\text{max}}' (j, \sigma, i) = d_{\text{max}} (j, \sigma, i) \) does not hold for every event. However, the result will be uniquely determined by the element sequence in \( \Sigma_{j,\text{com},i} \).

In the rest of this section, we verify that (1) the overall system behavior with communication delays computed by Algorithm 2 is optimal and nonblocking, and (2) every uncontrollable event will not be blocked by any channel. The latter statement is true and the reason is as follows. First, in each iteration of Algorithm 2 the maximal delay bound of a channeled event requires that the event will not be blocked...
by its corresponding channel; second, by inspecting the transition diagram of one channel model, say \( \text{CH}(j, \sigma, i) \) as shown in Fig. 2, the occurrence any event other than \( \sigma \) will not be blocked by \( \text{CH}(j, \sigma, i) \). So, we only need to verify statement (1).

First, we compute the overall behavior of the system with communication delay. We remark here that all the channels work independently. In particular, if \( \sigma \) needs to be transmitted from \( G_i \) to \( \text{SUP}_j \) and \( \text{SUP}_k \) \((k \neq i)\) with unbounded delay simultaneously, two channels \( \text{CH}(i, \sigma, j) \) and \( \text{CH}(i, \sigma, k) \) will be built to transmit \( \sigma \) to \( \text{SUP}_j \) and \( \text{SUP}_k \) simultaneously.

Let \( d'_{\text{max}}(j, \sigma, i) \) be the maximal delay bounds for \( \sigma_k \in \Sigma_{j, \text{com}, i} \) returned by Algorithm 2; then \( d'_{\text{max}}(j, \sigma, i) \in \mathbb{N} \cup \{\infty\} \). If \( d'_{\text{max}}(j, \sigma, i) = \infty \), we use \( \text{CH}(j, \sigma, i) \) representing unbounded delay to transmit \( \sigma_k \). If \( d'_{\text{max}}(j, \sigma, i) \in \mathbb{N}^+ \), we use \( \text{CH}_{d'_{\text{max}}}(j, \sigma, i) \) representing \( d'_{\text{max}}(j, \sigma, i) \)-bounded delay. In these two cases, \( \sigma_k \) is transmitted to \( \text{SUP}_i \) by channel model, and we use \( \Sigma_{\text{chn}} \) to represent the set of such events, i.e.

\[
\Sigma_{\text{chn}} = \{\sigma_k \in \Sigma_{j, \text{com}, i} | i, j \in \mathbb{N}, d'_{\text{max}}(j, \sigma, i) \in \mathbb{N}^+ \cup \{\infty\}\}
\]

and \( \Sigma_{\text{sig}} \) the set of their corresponding signal event \( \sigma'_k \), i.e. \( \Sigma_{\text{sig}} = \{\sigma'_k | \sigma_k \in \Sigma_{\text{chn}}\} \). In \( \text{SUP}_i \), we replace each instance of \( \sigma \) in all the corresponding local controllers and preempts by \( \sigma' \), and then obtain \( \text{SUP}'_i \). If \( d'_{\text{max}}(j, \sigma, i) = 0 \), the transmission of \( \sigma_k \) must be completed instantly; so in this case, we don’t change \( \text{SUP}_i \). Considering the communication delays returned by Algorithm 2 the overall system behavior is

\[
\text{SUP}'_{\text{com}} = (||\{\text{SUP}'_i | i \in \mathbb{N}, \Sigma_i \cap \Sigma_{\text{chn}} \neq \emptyset\}) \]

\[
\text{SUP}'_{\text{com}} = (||\{\text{SUP}'_i | i \in \mathbb{N}, \Sigma_i \cap \Sigma_{\text{chn}} = \emptyset\}) \]

\[
\text{SUP}'_{\text{com}} = (||\{\text{CH}(j, \sigma, i) | \sigma_k \in \Sigma_{\text{chn}}, d'_{\text{max}}(j, \sigma, i) = \infty\}) \]

\[
\text{SUP}'_{\text{com}} = (||\{\text{CH}_{d'_{\text{max}}}(j, \sigma, i) | \sigma_k \in \Sigma_{\text{chn}}, d'_{\text{max}}(j, \sigma, i) \in \mathbb{N}^+\}) \]

The alphabet of \( \text{SUP}'_{\text{com}} \) is \( \Sigma'_{\text{com}} = \Sigma \cup \Sigma_{\text{sig}} \).

Now, we turn to verify that \( \text{SUP}'_{\text{com}} \) has the same behavior with \( \text{SUP} \) in (4) under the projection of \( P_{\text{com}} : \Sigma'_{\text{com}} \rightarrow \Sigma^* \). Using the concept of delay-robustness, we confirm this statement by the following theorem.
Theorem 2. For SUP
\textsubscript{\text{com}}\textsuperscript{'} in [23], SUP is delay-robust relative to Σ\textsubscript{\text{com}}, in the sense that

\[
\text{Supqc}(\text{SUP}\textsubscript{\text{com}}, \text{Null}(\Sigma\textsubscript{\text{sig}}))
\]

is structurally deterministic, and has the identical marked and closed behavior to SUP.

For transparency, we consider the case that \( R = \{\alpha, \beta\} \subseteq \Sigma\textsubscript{chn} \). Assume that \( \alpha \) is transmitted from \( G_i \) to SUP\( j \) by CH\( d_{\alpha}(i, \alpha, j) \) and \( \beta \) is transmitted from \( G_i \) to SUP\( k \) by CH\( d_{\beta}(i, \beta, k) \). In SUP\( j \) (resp. SUP\( k \)), replacing each instance of \( \alpha \) (resp. \( \beta \)) by \( \alpha' \) (resp. \( \beta' \)), we obtain SUP\( j' \) (resp. SUP\( k' \)).

The overall system behavior with communication delay of \( \alpha \) is represented by

\[
\text{SUP}\textsubscript{\text{com}}' = \text{SUP}\textsubscript{\text{com}}' \| (\text{CH}_{d_{\alpha}(i, \alpha, j)} \| \text{SUP}_{l})_{l \in N, l \neq j}
\]

with alphabet \( \Sigma'_{\text{com}} = \Sigma \cup \{\alpha'\} \), and the overall system behavior with communication delays of \( \alpha \) and \( \beta \) is represented by

\[
\text{SUP}'_{\text{com}} = \text{SUP}'_{\text{com}} \| (\text{CH}_{d_{\alpha}(i, \alpha, j)} \| \text{SUP}_{l})_{l \in N, l \neq j, l \neq k}
\]

with alphabet \( \Sigma''_{\text{com}} = \Sigma \cup \{\alpha', \beta'\} \). We shall need the natural projections

\[
\begin{align*}
P'_{\alpha} &: \Sigma'_{\text{com}}^* \rightarrow \Sigma^* \\
P'_{\beta} &: \Sigma''_{\text{com}}^* \rightarrow \Sigma'_{\text{com}}^* \\
P''_{R} &: \Sigma''_{\text{com}}^* \rightarrow \Sigma^*.
\end{align*}
\]

According to domains of these projections, we have

\[
P''_{R} = P'_{\alpha} P'_{\beta}.
\]

Considering only the communication delays of \( \alpha \) and \( \beta \), we have the following result; its proof is in Appendix B.

Proposition 3. For SUP\( \text{com}' \) and SUP\( \text{com}'' \) described above, if SUP in (4) is delay-robust relative to \( \alpha \) and SUP\( \text{com}' \) is delay-robust relative to \( \beta \), then SUP is delay-robust relative to \( R \) in the sense that (13), (14), (15) hold with \( P = P''_{R} \) and SUP\( \text{com}' \) replaced by SUP\( \text{com}''_{R} \).

Using Proposition 3 we can easily prove Theorem 2, which confirms that the overall system behavior with communication delays computed by Algorithm 2 is optimal and nonblocking. We illustrate this result by the ULTC example in Sect. VI.
VI. DISTRIBUTED SUPERVISORY CONTROL FOR ULTC

A. Model Description for ULTC

Transformers with tap-changing facilities constitute an important mean of controlling voltage throughout electrical power systems at all voltage levels. Fig. 4 shows the block diagram of a ULTC described in [16]. This ULTC consists of two components: Voltmeter and Tap-Changer. We consider a voltage control problem of the ULTC: when the voltage is not ‘normal’, design controllers to recover the voltage through controlling tap ratio after a time delay to recover the voltage. The timing behavior suggests a TDES approach to solve the voltage control problem. Each component is modeled as a TDES shown in Fig. 5 and associated events are listed in Table I.

Here, we consider a control logic (specification) described in [16]. There are two modes of operations: Automatic and Manual. In the automatic mode, the tap-changer works according to the following logic:
(i) If the voltage deviation $|\Delta V| > ID$ and $\Delta V$ is negative (positive) then the timer will start and when it times out a ‘tap increase (decrease) event’ will occur and the timer will reset.

(ii) If the voltage returns to the dead-band, because of a tap change or some other reason, then no tap change will occur.

(iii) If the voltage exceeds the value $V_{max}$, then the lowering tap command without delay happens instantaneously.

In the manual mode, the system is waiting for ‘Tap-up’, ‘Tap-down’, or ‘Automatic’ commands. In this paper, we are concerned with an Automatic/Manual mode, in which an operator joins in the system and the operator can change the operation model from one to the other. Fig. 6 shows the TDES model for the control specification in Automatic/Manual mode.

Note that since the tap increase (decrease) and lowering tap command would preempt the occurrence of $tick$, events 31, 33 and 35 are defined as forcible events.
Now the plant to be controlled is the synchronized behavior of Volmeter (VOLT), Tap-changer (TAP) and Operator (OPTR), i.e.

$$\text{PLANT} = \text{VOLT} \parallel \text{TAP} \parallel \text{OPTR}.$$  \hfill (25)

In the next subsection, monolithic and distributed supervisory controllers will be designed to satisfy the given specification and achieve an optimal (maximally permissive) and nonblocking behavior.

### B. Distributed Control Design for ULTC

We first synthesize the monolithic supervisor $\text{TDES SUP}$ in the sense that its marked behavior

$$L_m(\text{SUP}) = \text{SupC}(L_m(\text{SPEC}) \cap L_m(\text{VOLT}))$$  \hfill (26)

and its closed behavior $L(\text{SUP}) = L_m(\text{SUP})$. SUP has 231 states and 543 transitions, and contains disabling actions for all the prohibitive events and preemption actions on tick for all the forcible events. From this ‘monolithic’ supervisor SUP, it is difficult to figure out the control strategies.

Next, we solve the voltage control problem of ULTC in a distributed fashion by supervisor localization. As shown in Table II the prohibitive event set $\Sigma_{hib} = \{11, 31, 33, 35, 41, 43\}$ and the forcible event set $\Sigma_{for} = \{31, 33, 35\}$. Applying localization procedure described in Sect II-B we obtain a set of local controllers $\text{LOC}_{11}^C$, $\text{LOC}_{31}^C$, $\text{LOC}_{33}^C$, $\text{LOC}_{35}^C$, $\text{LOC}_{41}^C$ and $\text{LOC}_{43}^C$ with state sizes being 7, 7, 7, 2, 2 and 2 respectively, and a set of local preemptors $\text{LOC}_{31}^P$, $\text{LOC}_{33}^P$ and $\text{LOC}_{35}^P$ with state sizes being 7, 7 and 2 respectively, as shown in Fig. 7. These local controllers and preemptors have much smaller state sizes than SUP (with 231 states).
Using these constructed local controllers/preemptors and applying the allocation policy described in Sect. II-B we build a distributed control architecture of ULTC as displayed in Fig. 8. Each component is controlled by its corresponding local controllers and preemptors. Their local controlled behaviors are

\[
\text{SUP}_V = \text{VOLT} || \text{LOC}_{C_{11}};
\]

\[
\text{SUP}_T = \text{TAP} || (\text{LOC}_{C_{31}} || \text{LOC}_{C_{33}} || \text{LOC}_{C_{35}}) || (\text{LOC}_{P_{31}} || \text{LOC}_{P_{33}} || \text{LOC}_{P_{35}});
\]

\[
\text{SUP}_O = \text{OPTR} || (\text{LOC}_{C_{41}} || \text{LOC}_{C_{43}}).
\]

It is guaranteed by supervisor localization of TDES (15) that the ULTC under the control of these local controllers and preemptors, has the identical closed and marked behavior to SUP in (26), i.e \( L(\text{SUP}) = L(\text{SUP}_V) || L(\text{SUP}_T) || L(\text{SUP}_O) \) and \( L_m(\text{SUP}) = L_m(\text{SUP}_V) || L_m(\text{SUP}_T) || L_m(\text{SUP}_O) \).

C. Delay-Robustness Verification for ULTC

Next, we verify the delay-robustness of the obtained distributed supervisory control architecture shown in Fig. 8 relative to the communication events. For illustration, we consider the following cases.
(i) Event 30

This case shows that \( \text{SUP} \) is unbounded delay-robust relative to the uncontrollable event 30.

Considering the communication delay of event 30 from \( \text{TAP} \) to \( \text{SUP}_O \), first create a communication channel model \( \text{CH}(T, 30, O) \) (where \( T \) and \( O \) represent \( \text{TAP} \) and \( \text{SUP}_O \) respectively) shown in Fig. 9 using a new event label \( 30' \) to represent the corresponding signal event. Second, use \( 30' \) to replace 30 in \( \text{LOC}_{41} \) and \( \text{LOC}_{43} \) that control \( \text{OPTR} \), and obtain the new local controlled behavior \( \text{SUP}'_O \). Third, compute the channeled behavior \( \text{SUP}'_{30} \) with 30 being the channeled event, according to

\[
\text{SUP}'_{30} = \text{SUP}_V || \text{SUP}_T || \text{CH}(T, 30, O) || \text{SUP}'_O
\]

over the augmented alphabet \( \{10, 11, ..., 43, 30'\} \).

To check delay-robustness, first we project \( \text{SUP}'_{30} \) modular supremal quasi-congruence with nulled event \( 30' \), to get, say

\[
\text{QCSUP}'_{30} = \text{Supqc}(\text{SUP}'_{30}, \text{Null}[30'])
\]

with 56 states and 130 transitions. Then, we verify that \( \text{QCSUP}'_{30} \) has the identical closed and marked behavior to \( \text{SUP} \), i.e. \( \text{QCSUP}'_{30} \cong \text{SUP} \). Finally, since 30 is an uncontrollable event, we verify by controllability checking procedure \textit{Condat} in [17] that 30 is not blocked by its channel \( \text{CH}(T, 30, O) \). By Definition 2, \( \text{SUP} \) is unbounded delay-robust relative to 30.
For an physical interpretation, consider the case that the voltmeter reported an increase in voltage, the tap has received a tap up command, but the tap up operation failed (i.e. events 11, 10, tick, tick, tick, tick, 33 and 30 have occurred in sequence). On the one hand, if the operator knows that the tap up operation failed (i.e. SUP receives the occurrence of 30), the system will enter the automatic mode (event 41 is enabled). On the other hand, if the operator does not know the failure (i.e. SUP does not receive the occurrence of 30), the events are potentially generated in the system are 11, tick, 35, 41 and 43, but the event enabled by the local controllers is only tick; only after that SUP receives the occurrence of 30, event 41 will be enabled. In both cases, the system will stay still and will not go back to the automatic mode until the operator has known the failure operation. Hence, despite of the indefinite communication delay of event 30, the system behavior is 'correct' and 'complete', as required by delay-robustness. Furthermore, if the system does not enter the automatic mode (event 41 occurs), the operation failure generated by the tap-changer will not happen again; this is to say that the reoccurrence of event 30 depends on the occurrence of event 41, which can be disabled by its corresponding controller. So in this case event 30 will not blocked by its channel. Applying Similar analysis to other cases, we conclude that SUP is unbounded delay-robust relative to 30.

(ii) Event 10

This case shows that SUP is 5-bounded delay-robust relative to the uncontrollable event 10.

Considering the communication delay of event 10 from VOLT to SUP, similar to (i), CH(V, 10, O) shown in Fig. is created to transmit event 10 and we can verify that SUP is delay-robust relative to event 10. Again, since event 10 is uncontrollable, we verify by Condat that 10 is blocked by CH(V, 10, O). So, we apply Algorithm and obtain that the maximal delay bound for 10 is 5. According to Definition to preserve the global optimality and nonblocking of the overall system, the transmission of event 10 must be completed before the fifth tick occurs.

For illustration, inspect the system behavior when an increase in the voltage is reported (i.e. event 11 and 10 has occurred sequentially). In this case, the event that is enabled by the local controllers is only tick. This situation will not change until the fifth tick occurs. If the operator knows the voltage increase before the fifth tick occurs, it will switch the system to manual model (i.e. event 43 occurs); otherwise, event 43 will be disabled and the system behavior will violate the specification. Hence, if the transmission of event 10 is completed in time (before the fifth tick occurs) the system behavior is still optimal and nonblocking.

By the same method, one can verify that SUP is 5-bounded delay-robust relative to event 14.

(iii) Event 33
This case shows that event 33 is delay-critical with respect to SUP, i.e. the transmission of 33 must be completed instantly.

First, we verified by Proposition 1 that SUP is not delay-robust relative to event 33. Second, applying Algorithm 1, we obtain the maximal delay bound for event 33 is zero. So, we conclude that event 33 is delay-critical with respect to SUP.

For an physical interpretation, consider the case that the voltmeter reported an increase in voltage and the tap has received a tap up command (i.e. events 11, 10, tick, tick, tick, tick, tick, and 33 have occurred in sequence). Now the events that are eligible to occur in the plant and enabled by the local controllers are events 30 and 32. In the one hand, if the operator knows that the tap-changer has received the voltage increase command (i.e. SUP_O knows the occurrence of 33), and the tap-up operation failed (i.e. 30 occurs), the system will enter the manual model by the operation failure (the occurrence of 30). However, if SUP_O does not receives the occurrence of 33, but 30 still occurs, event 43 will be enabled. In this case, the operator forces the system into the manual mode. Hence, the indefinite communication delay of event 33 leads to a different control decisions on event 43. Also, since tick does not occur before the different decisions are made, we conclude that event 33 is delay-critical with respect to the system.

D. Distributed Supervisory Control Architecture for ULTC with Communication Delay

In this subsection, we first apply Algorithm 2 to compute a set of maximal delay bounds for the communication events in ULTC, and then verify that the overall behavior system with these communication delays is optimal and nonblocking.

First, by Algorithm 2, we obtain that $d'_{max}(T, 30, O) = \infty$ ($d'_{max}(T, 30, O)$ represents the maximal delay bound for event 30 transmitted from TAP to SUP_O; similar notations is applicable to others), $d'_{max}(V, 10, O) = d'_{max}(V, 14, O) = 5$, and for the rest events, $d'_{max} = 0$.

Next, we compute the overall behavior of the system with these communication delays. For this purpose, we use $\text{CH}(T, 30, O)$ shown in Fig. 9 to transmit event 30, $\text{CH}_5(V, 10, O)$ and $\text{CH}_5(V, 14, O)$ shown in Fig. 10 to transmit events 10 and 14. So we relabel events 10, 14, 30 in their local controllers by $10'$, $14'$ and $30'$ respectively, and obtain SUP''_O. For the rest events, their transmission must be completed instantly, no channel is needed. Now, the overall system behavior is

$$\text{SUP}_{com}' = \text{SUP}_V || \text{SUP}_T || \text{CH}_5(V, 10, O) || \text{CH}_5(V, 14, O) || \text{CH}(T, 30, O) || \text{SUP}''_O$$

over the augmented alphabet \{10, 11, ..., 43, 10', 14', 30'\}.
Finally, we verify that SUP is delay-robust relative to \( \{10, 14, 30\} \). Let 

\[
QCSUP'_{com} = \text{Supqc}(SUP'_{com}, \text{Null}[10', 14', 30']);
\]

\( QCSUP'_{com} \) has 56 states and 130 transitions and \( QCSUP'_{com} \cong \text{SUP} \). Also, we verify by Condat that events 10, 14, 30 are not be blocked by their corresponding channels. Hence, we conclude that SUP is delay-robust relative to \( \{10, 14, 30\} \), namely the overall system \( SUP'_{com} \) is optimal and nonblocking within the computed delay bounds.

\[ \text{VII. Conclusions} \]

In this paper we have studied distributed control obtained by supervisor localization in TDES on the relaxed assumption (compared to previous literature \cite{11,12}) that inter-agent communication of selected ‘communication events’ (channeled events) may be subject to unknown time delays. For this distributed architecture, first we have identified properties of ‘delay-robustness’ which guarantees that the properties of our delay-free distributed control continue to be enforced in the presence of delay, albeit with possibly degraded temporal behavior. Then, we have introduced the verification tool for checking delay-robustness in \cite{9} and the standard controllability checking procedure in \cite{14} to determine the bounded and unbounded delay-robustness. Next, for those events that are identified as bounded delay-robust, we have proposed an algorithm to determine the maximal delay bound \( d_{max} \) such that the system is \( d_{max} \)-bounded delay-robust, but not \( (d_{max} + 1) \)-bounded delay-robust. Finally, we have presented another algorithm to compute a set of maximal delay bounds for one for each communication event, and verified that the overall system behavior with these computed communication delays is optimal and nonblocking. An ULTC example has
exemplified these results, showing how to verify the bounded and unbounded delay-robustness, determine
the maximal delay bound for bounded delay-robustness, and compute the maximal delay bounds for
preserving the optimality and nonblocking of the system with communication delay.

With the definitions and tests reported here as basic tools, future work should include the investigation
of alternative channel models and, of especial interest, global interconnection properties of a distributed
system of DES and TDES which render delay-robustness more or less likely to be achieved.

APPENDIX A

PROOF OF PROPOSITION [2]

To prove Proposition [2] we first introduce the following Lemmas [1] and [2]

Lemma 1. For any $d \in \mathbb{N}$,

\begin{align}
L(\text{SUP}) & \subseteq P_{ch}L(\text{SUP}'_d) \quad (27) \\
L_m(\text{SUP}) & \subseteq P_{ch}L_m(\text{SUP}'_d) (28)
\end{align}

Note that here even with different delay bound $d$, the alphabets of $\text{SUP}'_d$ and $\text{CH}_d(j, \sigma, i)$ are
constantly $\Sigma' = \Sigma \cup \{\sigma'\}$ and $\{\sigma, \text{tick}, \sigma'\}$, respectively. Define natural projection $P_{ch}: \Sigma'^* \to \{\sigma, \sigma'\}^*$
as in (2). The intuitive reason for Lemma [1]is that for any behavior of $\text{SUP}$, it can be achieved in $\text{SUP}'_d$
if the transmission of $\sigma$ is completed instantly.

Proof of Lemma [1]: Here we only prove that $L(\text{SUP}) \subseteq P_{ch}L(\text{SUP}'_d)$; (28) can be easily proved by
replacing $L$ by $L_m$ throughout.

Let $s \in L(\text{SUP})$; we must show that there exists string $t \in L(\text{SUP}'_d)$ such that $P_{ch}(t) = s$. First, we
assume that only one instance of $\sigma$ appeared in $s$, and write $s = x_1\sigma x_2$ where $x_1, x_2$ are free of $\sigma$. By
(20) and observing that $\text{SUP}'_d$ is obtained by replacing each instance of $\sigma$ by $\sigma'$, it is easy to see that
t := $x_1\sigma\sigma'x_2 \in L(\text{SUP}'_d)$, and thus $P_{ch}(t) = s$. So, $L(\text{SUP}) \subseteq P_{ch}L(\text{SUP}'_d)$. This result can be easily
extended to the general case that $s$ has multiple instance of $\sigma$. The reason is that $\sigma$ is transmitted by the
channel model and the reoccurrence of $\sigma$ is permitted only when the transmission of last $\sigma$ is completed.

Hence, if letting $s = x_1\sigma x_2\sigma..., x_{k-1}\sigma x_k$, it is easy to find out a string $t = x_1\sigma\sigma'x_2\sigma\sigma'..., x_{k-1}\sigma\sigma'x_k$
such that $t \in L(\text{SUP}'_d)$ and $P_{ch}t = s$. Hence, $L(\text{SUP}) \subseteq P_{ch}L(\text{SUP}'_d)$ is proved.

Lemma 2. Let $t = x_1\sigma x_2\sigma'x_4 \in L_m(\text{SUP}'_d)$ where $x_1, x_2, x_3$ and $x_4$ are strings free of $\sigma$ and $\sigma'$,
i.e. $x_1, x_2, x_3, x_4 \in (\Sigma - \{\sigma\})^*$. Then $t' := x_1\sigma x_2\sigma' x_3 x_4 \in L_m(\text{SUP}'_d)$. 26
Lemma 2 says that if the transmission of $\sigma$ completed after string $x_2x_3$ will lead to a legal state in $\text{SUP}'_d$, then the transmission completed after $x_2$ in a shorter time will also lead to a legal state in $\text{SUP}'_d$.

**Proof of Lemma 2**: Recall that $\text{SUP}'_i$ is $\text{SUP}_i$ with transitions labeled $\sigma$ relabeled $\sigma'$. By definition of synchronous product, $x_2$, $x_3$ and $\sigma'$ can be re-ordered without affecting the membership of $t$ in $L_m(\text{SUP}'_d)$, namely the strings $t'$ formed from $t$ by the successive replacement

$$x_1\sigma x_2 x_3 \sigma' x_4 \rightarrow x_1 \sigma x_2 x_3 x_4$$

$$\rightarrow x_1 \sigma x_2 \sigma' x_3 x_4$$

will belong to $L_m(\text{SUP}'_d)$ as well. In other words, if the transmission of $\sigma$ is completed after a string shorter than $x_2x_3$ (hence before the bound $d$ is met), the behavior also belongs to $L_m(\text{SUP}'_d)$.

With Lemmas 1 and 2 we prove Proposition 2 as follows.

**Proof of Proposition 2**: To prove $\text{SUP}$ is $(d-1)$—bounded delay-robust relative to $\sigma$, we must verify (13), (14), (15) with $P = P_{ch}$ (defined in (18)) and $\text{SUP}'$ replaced by $\text{SUP}'_{d-1}$, and if $\sigma \in \Sigma_u$, $P_{ch}^{-1} L(\text{CH}_{d-1}(j, \sigma, i)) \sigma \cap L(\text{NSUP}) \subseteq P_{ch}^{-1} L(\text{CH}_{d-1}(j, \sigma, i))$.

For (13), we prove that $P_{ch} L(\text{SUP}_{d-1}') \supseteq L(\text{SUP})$ and $P_{ch} L(\text{SUP}_{d-1}') \subseteq L(\text{SUP})$ in sequence. $P_{ch} L(\text{SUP}_{d-1}') \supseteq L(\text{SUP})$ is obtained from Lemma 1 immediately. Now we prove that $P_{ch} L(\text{SUP}_{d-1}') \subseteq L(\text{SUP})$. Inspecting the transition diagram of bounded channel model in Fig. 3 $L(\text{CH}_{d-1}(j, \sigma, i)) \subseteq L(\text{CH}_{d}(j, \sigma, i))$. By (20),

$$L(\text{SUP}'_{d-1}) \subseteq L(\text{SUP}'_d).$$

Since $\text{SUP}$ is $d$—bounded delay-robust relative to $\sigma$, $P_{ch} L(\text{SUP}'_d) \subseteq L(\text{SUP})$. Hence $P_{ch} L(\text{SUP}_{d-1}') \subseteq L(\text{SUP})$.

For (14), it can be confirmed from the proof of (13) by replacing $L$ by $L_m$ throughout.

For (15), we must show that for all $s \in \Sigma^*$ and $w \in \Sigma^*$

$$s \in L(\text{SUP}'_{d-1}) \land (P_{ch}s)w \in L_m(\text{SUP})$$

$$\Rightarrow (\exists v \in \Sigma^*) P_{ch}v = w \land sv \in L_m(\text{SUP}'_{d-1})$$

By (29), that $s \in L(\text{SUP}'_{d-1})$ derives $s \in L(\text{SUP}'_d)$. Since $\text{SUP}$ is $d$—bounded delay-robust relative to $\sigma$, there exists $u \in \Sigma^*$ (to distinguish with $v$ for $\text{SUP}'_{d-1}$) such that $P_{ch}u = w$ and $su \in L_m(\text{SUP}'_d)$.

As mentioned in Proof of Lemma 1, the transmissions of two instances of $\sigma$ will not be interfered by each other. Thus here we consider the case that only one instance of $\sigma$ (and correspondingly only one
\( \sigma' \) exists in \( su \); the general cases can be dealt similarly. Hence, \( su \) can be rewritten as \( su = s_1\sigma s_2\sigma' s_3u \) with \( u \) is free of \( \sigma \) and \( \sigma' \), \( su = s_1\sigma s_2u_1\sigma' u_2 \), or \( su = su_1\sigma u_2\sigma' u_3 \) with \( s \) is free of \( \sigma \) and \( \sigma' \), where \( s_1, s_2, s_3, u_1, u_2, u_3 \) are free of \( \sigma \) and \( \sigma' \). Next, we prove (30) from the following three cases.

(i) \( su = s_1\sigma s_2\sigma' s_3u \). By (21), it follows from \( su \in L_m(\text{SUP}_{d-1}') \) that \( su \in L_m(\text{NSUP}) \). By the same reason, that \( s \in L(\text{SUP}_{d-1}') \) derives that \( P_{ch}(s) \in L(\text{CH}_{d-1}(j, \sigma, i)) \). Also, from \( s = s_1\sigma s_2\sigma' s_3 \), we have \( P_{ch}(s) = \sigma \sigma' \); so \( P_{ch}(s) \in L_m(\text{CH}_{d-1}(j, \sigma, i)) \). Moreover, \( u \) is free of \( \sigma \) and \( \sigma' \), implying that \( P_{ch}u = \epsilon \). So \( P_{ch}(su) = P_{ch}(s) \in L_m(\text{CH}_{d-1}(j, \sigma, i)) \). Again, by (21), \( su \in L_m(\text{SUP}_{d-1}') \), namely there exists \( v = u \) such that (30) holds.

(ii) \( su = s_1\sigma s_2u_1\sigma' u_2 \). By Lemma 2 from \( su \in L_m(\text{SUP}_{d-1}') \), it derives that \( s_1\sigma s_2u_1u_2 \in L_m(\text{SUP}_{d-1}') \). Similar to (i), we have \( s_1\sigma s_2\sigma' u_1u_2 \in L_m(\text{SUP}_{d-1}') \). Let \( v = \sigma' u_1u_2 \); then \( P_{ch}v = \sigma' = P_{ch'} = w \) and (30) holds.

(iii) \( su = su_1\sigma u_2\sigma' u_3 \). By Lemma 2 we have \( su_1\sigma u_2\sigma' u_3 \in L_m(\text{SUP}_{d-1}') \). Similar to (i), we have \( su_1\sigma u_2u_3 \in L_m(\text{SUP}_{d-1}') \). Let \( v = u_1\sigma u_2u_3 \); then \( P_{ch}v = \sigma' = P_{ch}u = w \) and (30) holds.

By these three steps, (30) is confirmed.

Finally, we show that if \( \sigma \in \Sigma_u \), \( P_{ch}^{-1} L(\text{CH}_{d-1}(j, \sigma, i)) \cap L(\text{NSUP}) \subseteq P_{ch}^{-1} L(\text{CH}_{d-1}(j, \sigma, i)) \). Let \( s \in P_{ch}^{-1} L(\text{CH}_{d-1}(j, \sigma, i)) \) and \( s \sigma \in L(\text{NSUP}) \). We claim that \( s \sigma \in P_{ch}^{-1} L(\text{CH}_{d-1}(j, \sigma, i)) \); otherwise, \( \sigma \) is blocked after string \( s \). Write \( \text{CH}_d(j, \sigma, i) = (C_d, \Sigma, \text{com}, \tau_d, c_{d,0}, \{c_{d,0}\}) \) for any \( d \in \mathbb{N} \) where \( \Sigma_{\text{com}} = \{\sigma, \text{tick}, \sigma'\} \); then \( \tau_{d-1}(c_{d-1,0}, P_{ch}s) \neq c_{d-1,0} \) \( c_{d-1,0} \) is the only marker state). Inspecting the transition diagram of \( \text{CH}_d(j, \sigma, i) \) shown in Fig. 3 only \( \text{dth} \) \( \text{tick} \) (not \( \sigma \)) is defined in \( \text{CH}_d(j, \sigma, i) \), but not in \( \text{CH}_{d-1}(j, \sigma, i) \). So from \( \tau_{d-1}(c_{d-1,0}, P_{ch}s) \neq c_{d-1,0} \), we obtain that \( \tau_d(c_{d,0}, P_{ch}s) \neq c_{d,0} \). Again, since \( \sigma \) is only defined at \( c_{d,0} \), the occurrence of \( \sigma \) after \( s \) will be blocked by \( \text{CH}_d(j, \sigma, i) \); hence a contradiction to that \( \text{SUP} \) is \( d \)-bounded delay-robust relative to \( \sigma \). The claim is confirmed.

**APPENDIX B**

**PROOF OF PROPOSITION 3**

**Proof of Proposition 3** For (13), we prove that \( P''_R L(\text{SUP}_R') \supseteq L(\text{SUP}) \) and \( P''_R L(\text{SUP}_R'') \subseteq L(\text{SUP}) \) in sequence. First, \( P''_R L(\text{SUP}_R') \supseteq L(\text{SUP}) \) is obtained directly from Lemma 1. Then \( P''_R L(\text{SUP}_R'') \subseteq L(\text{SUP}) \) is derived from \( P''_R L(\text{SUP}_R'') \subseteq L(\text{SUP}_R') \) (by delay-robustness of \( \text{SUP}_R' \) relative to \( \beta \)), \( P''_R L(\text{SUP}_R') \subseteq L(\text{SUP}) \) (by delay-robustness of \( \text{SUP} \) relative to \( \alpha \)), and \( P''_R = P''_R \) (as in (24)).

For (14), we must show that \( P''_R L_m(\text{SUP}') = L_m(\text{SUP}) \); it can be confirmed by replacing \( L \) by \( L_m \) throughout.
For (15), we must show that for all $s \in \Sigma_R''$ and $w \in \Sigma^*$,

$$s \in L(SUP_R'') \& (P''_R s) w \in L_m(SUP) \Rightarrow (\exists v \in \Sigma_R'') P'_R v = w \& sv \in L_m(SUP_R'').$$

Let $t = P'_R s$, then from $s \in L(SUP_R'')$ it derives that $t \in P'_R(L(SUP_R'')) \subseteq L(SUP_R')$ (by SUP$_R'$ is delay-robust relative to \( \beta \)), we have $P'_R(L(SUP_R'')) \subseteq L(SUP'_R)$; it follows from $(P''_R s) w \in L_m(SUP)$ and (24) that $(P'_R t) w = (P''_R s) w \in L_m(SUP)$. So by the delay-robustness of SUP relative to \( \alpha \), there exists $u \in \Sigma_R''$ such that $P'_R u = w$ and $tu \in L_m(SUP_R')$. Furthermore, we already have $t = P'_R s$, so $P'_R s \in L_m(SUP_R')$. Again, since SUP$_R'$ is delay-robust relative to \( \beta \), there exists $v \in \Sigma_R''$, such that $P'_R v = u$ such that $sv \in L_m(SUP_R'')$. However, since $P'_R u = w$ and $P''_R = P'_R P'_R$, $P''_R v = P'_R u = w$, as required by (31).

REFERENCES


