Supervisor Localization of Timed Discrete-Event Systems Under Partial Observation

Renyuan Zhang, *Member, IEEE*, and Kai Cai, *Senior Member, IEEE*

***Abstract*—We study *supervisor localization* for timed discrete-event systems under partial observation in the Brandin–Wonham framework. First, we employ timed relative observability to synthesize a partial-observation monolithic supervisor; the control actions of this supervisor include not only disabling action of prohibitible events (as that of controllable events in the untimed case) but also "clock-preempting" action of forcible events. Accordingly, we decompose the supervisor into a set of partial-observation local controllers one for each prohibitible event, as well as a set of partial-observation local preemptors one for each forcible event. We prove that these local controllers and preemptors collectively achieve the same controlled behavior as the partial-observation monolithic supervisor does. The above-mentioned results are illustrated by a timed workcell example.**

***Index Terms*—Partial observation, supervisor localization, timed discrete-event systems (TDES).**

I. Introduction

In [1]–[3], we developed a top-down approach, called *supervisor localization*, to the distributed control synthesis of multicomponent discrete-event systems (DES). The essence of localization is the decomposition of the monolithic (optimal and nonblocking) supervisor into local controllers for the individual components. In [4], we extended supervisor localization to timed DES (TDES) in the Brandin–Wonham framework [5]; in addition to local controllers (corresponding to disabling actions), a set of local preemptors is obtained corresponding to clock-preempting actions. More recently in [6], we have extended the untimed supervisor localization to the case of partial observation. In particular, we combined localization with relative observability [7] to first synthesize a partial-observation monolithic supervisor, and then decompose the supervisor into local controllers whose state changes are caused only by observable events.

In this technical note and its conference precursor [8], we generalize supervisor localization to study distributed control of multicomponent TDES under partial observation. We study partial-observation supervisor localization for TDES in the Brandin–Wonham framework, thereby extending both [4] and [6]. We propose to first synthesize a partial-observation monolithic supervisor using the concept of timed relative observability [9]. Timed relative observability is proved to be generally stronger than timed observability [10], weaker than normality [10], and closed under set union. Therefore, the supremal timed relatively observable (and controllable) sublanguage of a given language exists and may be effectively computed [9]. Since this supremal sublanguage is timed observable and controllable, it may be implemented by a partial-observation (feasible and nonblocking) supervisor [10]. We then suitably extend the localization procedure in [4] to decompose the supervisor into partial-observation local controllers and local preemptors for individual components, and prove that the derived local controlled behavior is equivalent to the monolithic one and is therefore globally observable and controllable.

The main contributions of this technical note are as follows. First, the proposed timed supervisor localization under partial observation extends the untimed counterpart in [6] and the full-observation counterpart in [4]. Compared with [6], not only is the monolithic supervisor's disabling action localized (as in the untimed case), but also its preemptive action is localized with respect to individual forcible events. While compared with [4], the new concepts of *partial-observation control cover* and *partial-observation preemption cover* are defined on the *powerset* of the monolithic supervisor's state set. In this way, in the transition structure of the resulting local controllers/preemptors, only observable events can lead to state changes. It is important to stress that the proposed timed supervisor localization under partial observation cannot be obtained directly by combining [4] and [6], because the treatment of the clock event tick is new and cannot be found in [4] or [6].

Second, the proposed supervisor localization for TDES provides a top-down, computationally effective approach to the distributed control of TDES under partial observation, which was not available in the literature. By the allocation policy described in Section III, the partial-observation local preemptors/controllers derived by the proposed localization procedures are allocated to each plant component, thereby building a purely distributed control architecture.

This technical note improves [8] in the following aspects.

1) We add a detailed explanation (in Section III) of the allocation policy of building from local controllers/preemptors a distributed control architecture under partial observation and present an example for illustration in Section V.
2) We provide a proof of Theorem 1 to explain that tick-preemptions by a set of constructed partial-observation local preemptors are consistent with those by the monolithic supervisor.
3) We add remarks on potential combinations of the proposed localization procedure with existing methods to solve different problems in TDES.

II. Preliminaries

This section reviews supervisory control of TDES in the Brandin–Wonham framework [5], [11, Ch. 9]. First, consider the untimed DES model $\mathbf{G}_{\text{act}} = (A, \Sigma_{\text{act}}, \delta_{\text{act}}, a_0, A_m)$; here A is the finite set of *activities*, Σ_{act} the finite set of *events*, $\delta_{\text{act}} : A \times \Sigma_{\text{act}} \to A$ the (partial) *transition function*, $a_0 \in A$ the *initial activity*, and $A_m \subseteq A$ the set of *marker activities*. Let $\mathbb{N}$ denote the set of natural numbers $\{0, 1, 2, \ldots\}$ and introduce *time* into $\mathbf{G}_{\text{act}}$ by assigning to each event $\sigma \in \Sigma_{\text{act}}$ a *lower bound* $l_{\mathbf{G},\sigma} \in \mathbb{N}$ and an *upper bound* $u_{\mathbf{G},\sigma} \in \mathbb{N} \cup \{\infty\}$ such that $l_{\mathbf{G},\sigma} \leq u_{\mathbf{G},\sigma}$. Also, introduce a distinguished event, written tick,

Manuscript received January 11, 2018; revised June 22, 2018, November 29, 2018, March 15, 2019, and April 4, 2019; accepted April 11, 2019. Date of publication April 18, 2019; date of current version December 27, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61573289 and Grant 11772264 and in part by the JSPS KAKENHI under Grant JP16K18122. Recommended by Associate Editor S. Takai. *(Corresponding author: Renyuan Zhang.)*

R. Zhang is with the School of Automation, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: ryzhang@nwpu.edu.cn).

K. Cai is with the Department of Electrical and Information Engineering, Osaka City University, Osaka 558-8585, Japan (e-mail: kai.cai@eng.osaka-cu.ac.jp).

Digital Object Identifier 10.1109/TAC.2019.2912008

to represent "tick of the global clock". Then, a TDES model

$$\mathbf{G} := (Q, \Sigma, \delta, q_0, Q_m) \tag{1}$$

is constructed from $\mathbf{G}_{\text{act}}$ (refer to [5] and [11, Ch. 9] for detailed construction) such that Q is the finite set of *states*, $\Sigma := \Sigma_{\text{act}} \,\dot\cup\, \{\text{tick}\}$ the finite set of events, $\delta : Q \times \Sigma \to Q$ the (partial) *state transition function*, q_0 the *initial state*, and Q_m the set of *marker states*.

Let Σ^* be the set of all finite strings of elements in $\Sigma = \Sigma_{\text{act}} \,\dot\cup\, \{\text{tick}\}$, including the empty string ϵ. The transition function δ is extended to $\delta : Q \times \Sigma^* \to Q$ in the usual way. The *closed behavior* of $\mathbf{G}$ is the language $L(\mathbf{G}) := \{s \in \Sigma^* | \delta(q_0, s)!\}$ and the *marked behavior* is $L_m(\mathbf{G}) := \{s \in L(\mathbf{G}) | \delta(q_0, s) \in Q_m\} \subseteq L(\mathbf{G})$. Let $K \subseteq \Sigma^*$ be a language; its *prefix closure* is $\overline{K} := \{s \in \Sigma^* | (\exists t \in \Sigma^*)\ st \in K\}$. K is said to be $L_m(\mathbf{G})$*-closed* if $\overline{K} \cap L_m(\mathbf{G}) = K$. TDES $\mathbf{G}$ is *nonblocking* if $\overline{L_m(\mathbf{G})} = L(\mathbf{G})$.

A TDES $\mathbf{G}$ can be graphically represented by both its *activity transition graph* (ATG), namely the ordinary transition graph of $\mathbf{G}_{\text{act}}$, and its *timed transition graph* (TTG), namely the ordinary transition graph of $\mathbf{G}$, incorporating the tick transition explicitly.

For two TDES $\mathbf{G}_1$ and $\mathbf{G}_2$ with ATG $\mathbf{G}_{1,\text{act}}$ and $\mathbf{G}_{2,\text{act}}$ defined on $\Sigma_{1,\text{act}}$ and $\Sigma_{2,\text{act}}$, respectively, their *composition* $\mathbf{Comp}(\mathbf{G}_1, \mathbf{G}_2)$ is a new TDES $\mathbf{G}$ such that $\mathbf{G}_{\text{act}} = \mathbf{G}_{1,\text{act}} || \mathbf{G}_{2,\text{act}}$, where "$||$" denotes the synchronous product of two generators [11]. The time bounds on the events of $\mathbf{G}$ are determined by: if $\sigma \in \Sigma_{1,\text{act}} \cap \Sigma_{2,\text{act}}$, then $l_{\mathbf{G},\sigma} = \max(l_{\mathbf{G}_1,\sigma}, l_{\mathbf{G}_2,\sigma})$ and $u_{\mathbf{G},\sigma} = \min(u_{\mathbf{G}_1,\sigma}, u_{\mathbf{G}_2,\sigma})$; if $\sigma \in \Sigma_{1,\text{act}} \setminus \Sigma_{2,\text{act}}$, then $l_{\mathbf{G},\sigma} = l_{\mathbf{G}_1,\sigma}$ and $u_{\mathbf{G},\sigma} = u_{\mathbf{G}_1,\sigma}$; and if $\sigma \in \Sigma_{2,\text{act}} \setminus \Sigma_{1,\text{act}}$, then $l_{\mathbf{G},\sigma} = l_{\mathbf{G}_2,\sigma}$ and $u_{\mathbf{G},\sigma} = u_{\mathbf{G}_2,\sigma}$. If this leads to $l_{\mathbf{G},\sigma} > u_{\mathbf{G},\sigma}$, the composition $\mathbf{G}$ does not exist. Composition of more than two TDES can be similarly constructed.[1]

To use TDES $\mathbf{G}$ in (1) for supervisory control, first designate a subset of events, denoted by $\Sigma_{\text{hib}} \subseteq \Sigma_{\text{act}}$, to be the *prohibitible* events that can be disabled by an external supervisor. Next, and specific to TDES, specify a subset of *forcible* events, denoted by $\Sigma_{\text{for}} \subseteq \Sigma_{\text{act}}$, which can *preempt* the occurrence of event tick. Now, it is convenient to define the *controllable* event set $\Sigma_c := \Sigma_{\text{hib}} \,\dot\cup\, \{\text{tick}\}$. The *uncontrollable* event set is $\Sigma_{uc} := \Sigma \setminus \Sigma_c$. A sublanguage $K \subseteq L_m(\mathbf{G})$ is *controllable* if, for all $s \in \overline{K}$

$$\text{Elig}_K(s) \supseteq \begin{cases} \text{Elig}_{\mathbf{G}}(s) \cap (\Sigma_{uc} \,\dot\cup\, \{\text{tick}\}) \\ \qquad \text{if } \text{Elig}_K(s) \cap \Sigma_{\text{for}} = \emptyset \\ \text{Elig}_{\mathbf{G}}(s) \cap \Sigma_{uc} \\ \qquad \text{if } \text{Elig}_K(s) \cap \Sigma_{\text{for}} \neq \emptyset \end{cases}$$

where $\text{Elig}_{\mathbf{G}}(s) := \{\sigma \in \Sigma | s\sigma \in L(\mathbf{G})\}$ and $\text{Elig}_K(s) := \{\sigma \in \Sigma | s\sigma \in \overline{K}\}$ are the subsets of eligible events after string s in $L(\mathbf{G})$ and K, respectively.

For partial observation, Σ is partitioned into Σ_o, the subset of observable events, and Σ_{uo}, the subset of unobservable events (i.e., $\Sigma = \Sigma_o \,\dot\cup\, \Sigma_{uo}$). Bring in the *natural projection* $P : \Sigma^* \to \Sigma_o^*$ defined by: (i) $P(\epsilon) = \epsilon$; (ii) $P(\sigma) = \sigma$ if $\sigma \in \Sigma_o$ and otherwise $P(\sigma) = \epsilon$; and (iii) for all $s \in \Sigma^*$ and $\sigma \in \Sigma$, $P(s\sigma) = P(s)P(\sigma)$. As usual, P is extended to $P : \text{Pwr}(\Sigma^*) \to \text{Pwr}(\Sigma_o^*)$, where $\text{Pwr}(\cdot)$ denotes powerset. Write $P^{-1} : \text{Pwr}(\Sigma_o^*) \to \text{Pwr}(\Sigma^*)$ for the *inverse image function* of P. A language $K \subseteq L_m(\mathbf{G})$ is *observable* if for every pair of strings $s, s' \in \Sigma^*$ with $P(s) = P(s')$ there holds

$$(\forall \sigma \in \Sigma_{\text{act}} \cup \{\text{tick}\}) s\sigma \in \overline{K}, s' \in \overline{K}, s'\sigma \in L(\mathbf{G}) \Rightarrow s'\sigma \in \overline{K}$$

where $P : \Sigma^* \to \Sigma_o^*$ is the corresponding natural projection.

A *supervisor* V under partial observation is any map $V : P(L(\mathbf{G})) \to \text{Pwr}(\Sigma)$. Then, the closed-loop system is $V/\mathbf{G}$ with closed behavior $L(V/\mathbf{G})$ and marked behavior $L_m(V/\mathbf{G})$ $(:= L(V/\mathbf{G}) \cap L_m(\mathbf{G}))$ [10]. A supervisor V is *nonblocking* if $\overline{L_m(V/\mathbf{G})} = L(V/\mathbf{G})$ and *admissible* if for each $s \in L(V/\mathbf{G})$, (i) $\Sigma_{uc} \subseteq V(P(s))$ and

$$\text{(ii) } \text{Elig}_{\mathbf{G}}(s) \cap V(P(s)) \cap \Sigma_{\text{for}} = \emptyset, \text{tick} \in \text{Elig}_{\mathbf{G}}(s) \\ \Rightarrow \text{tick} \in V(P(s)).$$

It has been proved in [10] that a nonblocking, admissible supervisory control V exists that synthesizes a (nonempty) sublanguage $K \subseteq L_m(\mathbf{G})$ such that $L_m(V/\mathbf{G}) = K$ if and only if K is (timed) observable, controllable, and $L_m(\mathbf{G})$-closed. While controllability and $L_m(\mathbf{G})$-closedness are properties closed under set union, observability is not; consequently when K is not observable, there generally does not exist the supremal observable (controllable and $L_m(\mathbf{G})$-closed) sublanguage of K.

Recently in [9], we have proposed a new concept of *timed relative observability*, which is stronger than timed observability, but it permits the existence of the supremal relatively observable sublanguage. Let $C \subseteq L_m(\mathbf{G})$. A language $K \subseteq C$ is *timed relatively observable* (or timed C-observable), if for every pair of strings $s, s' \in \Sigma^*$ with $P(s) = P(s')$ there holds

$$(\forall \sigma \in \Sigma_{\text{act}} \cup \{\text{tick}\}) \\ s\sigma \in \overline{K}, s' \in \overline{C}, s'\sigma \in L(\mathbf{G}) \Rightarrow s'\sigma \in \overline{K}. \tag{2}$$

In this technical note, only timed relative observability (or timed C-observability) is used; thus, for simplicity, we shall henceforth often omit the word "timed."

For an arbitrary sublanguage $E \subseteq L_m(\mathbf{G})$, write $\mathcal{CO}(E)$ for the family of C-observable, controllable, and $L_m(\mathbf{G})$-closed sublanguages of E. Then, $\mathcal{CO}(E)$ is nonempty (the empty language $\emptyset$ belongs) and is closed under set union; $\mathcal{CO}(E)$ has a unique supremal element $\sup \mathcal{CO}(E)$ given by

$$\sup \mathcal{CO}(E) = \bigcup \{K | K \in \mathcal{CO}(E)\}$$

which may be effectively computed [7], [9]. Note that since relative observability is stronger than observability, $\sup \mathcal{CO}(E)$ is observable (controllable and $L_m(\mathbf{G})$-closed), and since relative observability is weaker than normality, $\sup \mathcal{CO}(E)$ is generally larger than its normality counterpart.

III. Formulation of Partial-Observation Supervisor Localization Problem

Let the plant $\mathbf{G}$ be comprised of N component TDES

$$\mathbf{G}_k = (Q_k, \Sigma_k, \delta_k, q_{0,k}, Q_{m,k}), \quad k = 1, \dots, N. \tag{3}$$

Then $\mathbf{G} = \mathbf{Comp}(\mathbf{G}_1, \dots, \mathbf{G}_N)$, where $\mathbf{Comp}$ is the composition operator defined in Section II, which is used to build complex TDES from simpler ones. Let $\Sigma_o \subseteq \Sigma (:= \Sigma_1 \cup \cdots \cup \Sigma_N)$ be the subset of observable events and $P : \Sigma^* \to \Sigma_o^*$ the corresponding natural projection. Note that Σ_k are not pairwise disjoint, because event tick is shared by all components $\mathbf{G}_k$ (each TTG $\mathbf{G}_k$ is constructed from its ATG $\mathbf{G}_{k,\text{act}}$ and the corresponding time bounds by the rules in [5] and [11] and thus contains event tick); and tick may or may not be observable.

These components are implicitly coupled through a specification language $E \subseteq \Sigma^*$ that imposes a constraint on the global behavior of $\mathbf{G}$ (E may itself be the composition of multiple component specifications). For the plant $\mathbf{G}$ and the imposed specification E, let the generator

[1] There also exist *generalized* TDES (as defined in [11, Sec. 9.11]) that are represented by only TTG including tick in the alphabet. Namely, a generalized TDES does not have a corresponding ATG or timer information, and is simply an ordinary finite-state generator whose event set includes tick. Generalized TDES are often adopted to model temporal specifications and supervisors, and represent controlled plant behaviors. To compose two or more generalized TDES, we use the synchronous product "$||$," rather than $\mathbf{Comp}$.

$\mathbf{SUP} = (X, \Sigma, \xi, x_0, X_m)$ be such that

$$L_m(\mathbf{SUP}) := \sup \mathcal{CO}(E \cap L_m(\mathbf{G})).^2 \tag{4}$$

We call $\mathbf{SUP}$ the *controllable and observable behavior*. Note that $\mathbf{SUP}$ is not a "partial-observation supervisor" (to be defined in Section IV), which can only contain observable events as state changers. To rule out the trivial case, we assume that $L_m(\mathbf{SUP}) \neq \emptyset$.

The control actions of $\mathbf{SUP}$ include: first, disabling prohibitible events in Σ_{hib}; and second, preempting event tick via forcible events in Σ_{for}. Accordingly, the localization of $\mathbf{SUP}$'s control actions under partial-observation is with respect to not only each prohibitible event's disabling action (just as the untimed counterpart in [6]) but also each forcible event's preemptive action. The latter is specific to TDES, for which we introduce the new concept of "partial-observation local preemptor" as follows.

Let $\alpha \in \Sigma_{\text{for}}$ be an arbitrary forcible event, which may or may not be observable. We say that a generator

$$\mathbf{LOC}_\alpha^P = (Y_\alpha, \Sigma_\alpha, \eta_\alpha, y_{0,\alpha}, Y_{m,\alpha}),\ \Sigma_\alpha \subseteq \Sigma_o \cup \{\alpha, \text{tick}\}$$

is a *partial-observation local preemptor* for α if (i) $\mathbf{LOC}_\alpha^P$ preempts event tick consistently with $\mathbf{SUP}$ when tick is preempted by α and (ii) if $\sigma \in \{\alpha, \text{tick}\}$ is unobservable, then σ-transitions can only be selfloops in $\mathbf{LOC}_\alpha^P$ (other unobservable events in $\Sigma \setminus \Sigma_\alpha$ are not defined in, and thus not selfloops in, $\mathbf{LOC}_\alpha^P$).

First, condition (i) means that for all $s \in \Sigma^*$ if $s\alpha \in L(\mathbf{SUP})$, there holds

$$P_\alpha(s).\text{tick} \in L(\mathbf{LOC}_\alpha^P),\, s.\text{tick} \in L(\mathbf{G}) \Leftrightarrow s.\text{tick} \in L(\mathbf{SUP}) \tag{5}$$

where $P_\alpha : \Sigma^* \to \Sigma_\alpha^*$ is the natural projection. Notation s.tick means that event tick occurs after string s and will be used henceforth. Note that specific to TDES, only when $s\alpha \in L(\mathbf{SUP})$ can tick-occurrence after s be preempted by α in $\mathbf{LOC}_\alpha^P$. Also note that $\mathbf{LOC}_\alpha^P$ is not required to preempt tick consistently with $\mathbf{SUP}$ when tick is preempted by another forcible event α'; thus, $\mathbf{LOC}_\alpha^P$ is only responsible for the preemption of tick by α. Second, condition (ii) requires that only observable events may cause state change in $\mathbf{LOC}_\alpha^P$, i.e.,

$$(\forall y, y' \in Y_\alpha, \forall \sigma \in \Sigma_\alpha)\ y' = \eta_\alpha(y, \sigma)!, y \neq y' \Rightarrow \sigma \in \Sigma_o. \tag{6}$$

This requirement is a distinguishing feature of a partial-observation local preemptor as compared to its full-observation counterpart in [4].

Note that the event set Σ_α of $\mathbf{LOC}_\alpha^P$ in general satisfies

$$\{\alpha, \text{tick}\} \subseteq \Sigma_\alpha \subseteq \Sigma_o \cup \{\alpha, \text{tick}\}$$

in typical cases, both subset containments are strict. In fact, the events in $\Sigma_\alpha \backslash \{\alpha, \text{tick}\}$ are communication events that may be critical to achieve synchronization with other partial-observation local preemptors/controllers. Σ_α is not fixed *a priori*, but will be determined as part of the localization result presented in Section IV.

Next, let $\beta \in \Sigma_{\text{hib}}$ be an arbitrary prohibitible event, which may or may not be observable. A generator

$$\mathbf{LOC}_\beta^C = (Y_\beta, \Sigma_\beta, \eta_\beta, y_{0,\beta}, Y_{m,\beta}),\ \Sigma_\beta \subseteq \Sigma_o \cup \{\beta\}$$

is a *partial-observation local controller* for β if (i) $\mathbf{LOC}_\beta^C$ enables/disables the event β (and only β) consistently with $\mathbf{SUP}$ and (ii) if β is unobservable, then β-transitions can only be selfloops in $\mathbf{LOC}_\beta^C$. Here condition (i) means that for all $s \in \Sigma^*$ there holds

$$P_\beta(s)\beta \in L(\mathbf{LOC}_\beta^C),\, s\beta \in L(\mathbf{G}) \Leftrightarrow s\beta \in L(\mathbf{SUP}) \tag{7}$$

where $P_\beta : \Sigma^* \to \Sigma_\beta^*$ is the natural projection. Note that $\mathbf{LOC}_\beta^C$ is not required to disable/enable other prohibitible event β' consistently with $\mathbf{SUP}$; thus $\mathbf{LOC}_\beta^C$ is only responsible for the disablement/enablement of β. Condition (ii) imposes the same requirement (ii) of $\mathbf{LOC}_\alpha^P$ on $\mathbf{LOC}_\beta^C$, i.e., (6) holds for all $y, y' \in Y_\beta$ and $\sigma \in \Sigma_\beta$.

The event set Σ_β of $\mathbf{LOC}_\beta^C$ in general satisfies $\{\beta\} \subseteq \Sigma_\beta \subseteq \Sigma_o \cup \{\beta\}$; in typical cases, both subset containments are strict. Like Σ_α as mentioned earlier, Σ_β will be generated as part of our localization result.

The definition of partial-observation local controller differs from that of partial-observation local preemptor in condition (i) (conditions (ii) are identical because they are required for partial observation). Condition (i) of partial-observation local preemptor specially requires that the consistency on tick preemption is considered only when a forcible event α is enabled. Since every forcible event may preempt tick, there will exist a set of partial-observation local preemptors responsible for preempting the event tick, one for each relevant forcible event. While for any prohibitible event in Σ_{hib}, there is only one partial-observation local controller responsible for disabling/enabling it.

We are now ready to formulate the *partial-observation supervisor localization problem*.

Construct a set of partial-observation local preemptors $\{\mathbf{LOC}_\alpha^P | \alpha \in \Sigma_{\text{for}}\}$ and a set of partial-observation local controllers $\{\mathbf{LOC}_\beta^C \mid \beta \in \Sigma_{\text{hib}}\}$ with

$$L(\mathbf{LOC}) := \left(\bigcap_{\alpha \in \Sigma_{\text{for}}} P_\alpha^{-1} L(\mathbf{LOC}_\alpha^P) \right) \cap \left(\bigcap_{\beta \in \Sigma_{\text{hib}}} P_\beta^{-1} L(\mathbf{LOC}_\beta^C) \right) \tag{8}$$

$$L_m(\mathbf{LOC}) := \left(\bigcap_{\alpha \in \Sigma_{\text{for}}} P_\alpha^{-1} L_m(\mathbf{LOC}_\alpha^P) \right) \cap \left(\bigcap_{\beta \in \Sigma_{\text{hib}}} P_\beta^{-1} L_m(\mathbf{LOC}_\beta^C) \right) \tag{9}$$

such that the collective controlled behavior of $\mathbf{LOC}$ is equivalent to the controllable and observable controlled behavior $\mathbf{SUP}$ in (4) with respect to $\mathbf{G}$, i.e.,

$$L(\mathbf{G}) \cap L(\mathbf{LOC}) = L(\mathbf{SUP})$$
$$L_m(\mathbf{G}) \cap L_m(\mathbf{LOC}) = L_m(\mathbf{SUP}).$$

Having a set of partial-observation local preemptors $\{\mathbf{LOC}_\alpha^P | \alpha \in \Sigma_{\text{for}}\}$ and a set of partial-observation local controllers $\{\mathbf{LOC}_\beta^C \mid \beta \in \Sigma_{\text{hib}}\}$, we build for the TDES plant $\mathbf{G}$ with multiple components $\mathbf{G}_k$ $(k = 1, \ldots, N)$ [as in (3)] a nonblocking distributed control architecture under partial observation. Let $\Sigma_{\text{for},k} = \Sigma_k \cap \Sigma_{\text{for}}$ and $\Sigma_{\text{hib},k} = \Sigma_k \cap \Sigma_{\text{hib}}$ be the subset of forcible events and subset of prohibitible events of $\mathbf{G}_k$, respectively. One way of allocating the local preemptors/controllers to the components is as follows. First, construct a set of disjoint subsets of forcible events $\{\hat{\Sigma}_{\text{for},k} | k = 1, \ldots, N\}$

[2] $\mathbf{SUP}$ can be computed by an algorithm presented in [9] (Algorithm 2), with the ambient language C set to be the supremal controllable and $L_m(\mathbf{G})$-closed sublanguage of $E \cap L_m(\mathbf{G})$.

according to

$$\begin{aligned}\hat{\Sigma}_{\text{for},1} &:= \Sigma_{\text{for},1}\\ \hat{\Sigma}_{\text{for},2} &:= \Sigma_{\text{for},2} \setminus \hat{\Sigma}_{\text{for},1}\\ &\ \ \vdots\\ \hat{\Sigma}_{\text{for},N} &:= \Sigma_{\text{for},N} \setminus \big(\hat{\Sigma}_{\text{for},1} \cup \hat{\Sigma}_{\text{for},2} \cup \cdots \cup \hat{\Sigma}_{\text{for},N-1}\big). \qquad (10)\end{aligned}$$

Similarly, a set of disjoint subsets of prohibitible events $\{\hat{\Sigma}_{\text{hib},k} | k = 1,\ldots,N\}$ can be constructed. Second, let each local preemptor (respectively controller) belong to the component $\mathbf{G}_k$ such that $\hat{\Sigma}_{\text{for},k}$ (respectively $\hat{\Sigma}_{\text{hib},k}$) contains the corresponding forcible (respectively prohibitible) event. By this allocation policy, each local preemptor/controller will be owned by exactly one component, thereby we build a distributed control architecture for $\mathbf{G}$. Note that different orders of choosing $\hat{\Sigma}_{\text{for},k}$ and $\hat{\Sigma}_{\text{hib},k}$ generally lead to different allocation policies, the choice of which is case dependent. We shall use this allocation rule in the example (timed workcell) in Section V below.

IV. Partial-Observation Localization Procedure

We solve the partial-observation supervisor localization problem of TDES by developing a partial-observation localization procedure for the preemptive and disabling actions. The procedure extends the untimed counterpart in [6]. In particular, localizing the preemption of event tick with respect to each forcible event under partial observation is novel in the current TDES setup, for which we introduce the concept of "partial-observation preemption cover" as follows.

Let $\mathbf{G} = (Q, \Sigma, \delta, q_0, Q_m)$ be the TDES plant, $\Sigma_o \subseteq \Sigma$ the subset of observable events, and $P : \Sigma^* \to \Sigma_o^*$ the corresponding natural projection. Also let $\mathbf{SUP} = (X, \Sigma, \xi, x_0, X_m)$ be controllable and observable behavior [as defined in (4)]. We present the localizations of preemptive and disabling actions in the sequel. To this end, we need the concept of *uncertainty set*.

For $s \in L(\mathbf{SUP})$, let $U(s)$ be the subset of states of $\mathbf{SUP}$ that may be reached by some string s' that looks like s, i.e.,

$$U(s) = \{x \in X | (\exists s' \in \Sigma^*) P(s) = P(s'), x = \xi(x_0, s')\}.$$

We call $U(s)$ the *uncertainty set* [6] of the state $\xi(x_0, s)$ associated with string s. Let $\mathcal{U}(X) := \{U(s) \subseteq X | s \in L(\mathbf{SUP})\}$, i.e., $\mathcal{U}(X)$ is the set of uncertainty sets of all states (associated with strings in $L(\mathbf{SUP})$) in X. The size of $\mathcal{U}(X)$ is in general $|\mathcal{U}(X)| \leq 2^{|X|}$.

The transition function associated with $\mathcal{U}(X)$ is $\hat{\xi} : \mathcal{U}(X) \times \Sigma_o \to \mathcal{U}(X)$ given by

$$\hat{\xi}(U, \sigma) = \bigcup \{\xi(x, u_1 \sigma u_2) | x \in U, u_1, u_2 \in \Sigma_{uo}^*\}. \qquad (11)$$

With $\mathcal{U}(X)$ and $\hat{\xi}$, define the *partial-observation monolithic supervisor* [11], [12]

$$\mathbf{SUPO} = (\mathcal{U}(X), \Sigma_o, \hat{\xi}, U_0, U_m) \qquad (12)$$

where $U_0 = U(\epsilon)$ and $U_m = \{U \in \mathcal{U}(X) | U \cap X_m \neq \emptyset\}$. $\mathbf{SUPO}$ can be constructed by the well-known subset construction algorithm in [13] and it is known [11] that $L(\mathbf{SUPO}) = P(L(\mathbf{SUP}))$ and $L_m(\mathbf{SUPO}) = P(L_m(\mathbf{SUP}))$.

Now, let $U \in \mathcal{U}(X)$, $x \in U$ be any state in $\mathbf{SUP}$ and $\sigma \in \Sigma_c$ $(= \Sigma_{\text{hib}} \dot{\cup} \{\text{tick}\})$ be a controllable event. We say that the following statements hold:

i) σ is *enabled* at $x \in U$ if σ is defined at x in $\mathbf{SUP}$;

ii) σ $(\neq$ tick$)$ is *disabled* at $x \in U$ if it is not defined at x in $\mathbf{SUP}$, but is defined at some state q in $\mathbf{G}$ that *corresponds* to $x \in U$ (i.e., there exists a string $s \in \Sigma^*$ such that $\xi(x_0, s) = x$, and $\delta(q_0, s) = q$);

iii) σ is *not defined* at $x \in U$ if it is not defined at x in $\mathbf{SUP}$, and also not defined at any state in $\mathbf{G}$ that corresponds to x; and

iv) $\sigma =$ tick is *preempted* at $x \in U$ if tick is not defined at x in $\mathbf{SUP}$, but is defined at some state q in $\mathbf{G}$ that corresponds to x, and additionally there must exist a forcible event σ_f that is defined at x in $\mathbf{SUP}$.

The formal definitions of (i)–(iii) can be found in [6]. Since (iv) is specific to TDES (under partial observation), we define it as follows: σ $(=$ tick$)$ is *preempted* at $x \in U$ if $\neg\xi(x, \text{tick})!$ and

$$\begin{aligned}(\exists s \in \Sigma^*)(\exists \sigma_f \in \Sigma_{\text{for}}) \xi(x_0, s) = x \,\&\, \hat{\xi}(U_0, Ps) = U&\\ \&\, \xi(x, \sigma_f)! \,\&\, \delta(q_0, s.\text{tick})!.&\end{aligned}$$

Lemma 1: Given $\mathbf{SUP}$ in (4), let $U \in \mathcal{U}(X)$, $x \in U$, and $\sigma \in \Sigma_c$. If σ is enabled at $x \in U$, then for all $x' \in U$, either σ is also enabled at $x' \in U$ or σ is not defined at $x' \in U$. On the other hand, if σ is disabled (respectively preempted) at $x \in U$, then for all $x' \in U$, either σ is also disabled (respectively preempted) at $x' \in U$ or σ is not defined at $x' \in U$.

For the proof of Lemma 1, see [14].

A. Partial-Observation Localization of Preemptive Action

Under partial observation, the preemptive action after string $s \in L(\mathbf{SUP})$ depends not only on the single state $\xi(x_0, s)$, but also on the uncertainty set $U(s)$.

Fix an arbitrary forcible event $\alpha \in \Sigma_{\text{for}}$. First define $E_{\text{tick}} : \mathcal{U}(X) \to \{0, 1\}$ according to

$$(\forall U \in \mathcal{U}(X))\, E_{\text{tick}}(U) = \begin{cases} 1, & \text{if } (\exists x \in U) \xi(x, \text{tick})! \\ 0, & \text{otherwise.} \end{cases}$$

Thus, $E_{\text{tick}}(U) = 1$ means that tick is enabled at some state $x \in U$, i.e., tick is eligible to occur and its occurrence is not preempted by any forcible events. Then, by Lemma 1, at any other state $x' \in U$, tick is either enabled or not defined. Then, define $F_\alpha : \mathcal{U}(X) \to \{0, 1\}$ according to

$$(\forall U \in \mathcal{U}(X))$$

$$F_\alpha(U) = \begin{cases} 1, & \text{if } (\exists x \in U)\, \xi(x, \alpha)! \,\&\, \neg\xi(x, \text{tick})! \,\&\\ & ((\exists s \in \Sigma^*) \xi(x_0, s) = x \,\&\, \hat{\xi}(U_0, Ps) = U\\ & \qquad \&\, \delta(q_0, s.\text{tick})!)\\ 0, & \text{otherwise.} \end{cases}$$

Hence $F_\alpha(U) = 1$ means that tick is preempted by the occurrence of α at some state $x \in U$, i.e., there exists a state $x \in U$ such that tick is eligible to occur at some state in $\mathbf{G}$ that corresponds to x, but its occurrence is effectively preempted by α that has already been enabled at x. Again by Lemma 1, at any other state $x' \in U$, tick is either preempted or not defined. Note that at state x, α need not be the only forcible event that preempts tick, for there can be other forcible events, say α', defined at x. In that case, $F_{\alpha'}(U) = 1$ holds as well.

Based on the preemption information captured by E_{tick} and F_α mentioned earlier, we define the preemption consistency relation $\mathcal{R}_\alpha^P \subseteq \mathcal{U}(X) \times \mathcal{U}(X)$ (for α) as follows.

Definition 1: For $U, U' \in \mathcal{U}(X)$, we say that U and U' are *preemption consistent* with respect to α, written $(U, U') \in \mathcal{R}_\alpha^P$, if

$$E_{\text{tick}}(U) \cdot F_\alpha(U') = 0 = E_{\text{tick}}(U') \cdot F_\alpha(U).$$

Thus, a pair of uncertainty sets (U, U') satisfies $(U, U') \in \mathcal{R}_\alpha^P$ if tick is defined at some state of U, but not preempted by α at any state of U', and vice versa. It is easily verified that $\mathcal{R}_\alpha^P$ is reflexive and symmetric, but not transitive. Hence, $\mathcal{R}_\alpha^P$ is not an equivalence relation. This fact leads to the definition of a *partial-observation preemption cover*. Recall that a *cover* on a set $\mathcal{U}(X)$ is a family of nonempty subsets (or *cells*) $\mathcal{U}_i$ ($i \in I_\alpha$, I_α is an index set) of $\mathcal{U}(X)$ whose union is $\mathcal{U}(X)$, i.e., $\mathcal{U}(X) = \bigcup\{\mathcal{U}_i | \mathcal{U}_i \subseteq \mathcal{U}(X), \mathcal{U}_i \neq \emptyset, i \in I_\alpha\}$.

Definition 2: Let I_α be some index set, and $\mathcal{C}_\alpha^P = \{\mathcal{U}_i \subseteq \mathcal{U}(X) | i \in I_\alpha\}$ be a cover on $\mathcal{U}(X)$. We say that $\mathcal{C}_\alpha^P$ is a *partial-observation preemption cover* with respect to α if

(i) $(\forall i \in I_\alpha, \forall U, U' \in \mathcal{U}_i)\, (U, U') \in \mathcal{R}_\alpha^P$ and

(ii) $(\forall i \in I_\alpha, \forall \sigma \in \Sigma_o)(\exists U \in \mathcal{U}_i)\, \hat{\xi}(U, \sigma) \neq \emptyset$

$$\Rightarrow \big((\exists j \in I_\alpha)(\forall U' \in \mathcal{U}_i)\, \hat{\xi}(U', \sigma) \neq \emptyset \Rightarrow \hat{\xi}(U', \sigma) \in \mathcal{U}_j\big).$$

A partial-observation preemption cover $\mathcal{C}_\alpha^P$ lumps the uncertainty sets $U \in \mathcal{U}(X)$ into (possibly overlapping) *cells* $\mathcal{U}_i \in \mathcal{C}_\alpha^P$, $i \in I_\alpha$, according to the following: (i) the uncertainty sets U that reside in the same cell $\mathcal{U}_i$ must be pairwise preemption consistent and (ii) for every observable event $\sigma \in \Sigma_o$, the uncertainty sets U' that can be reached from any uncertainty set $U \in \mathcal{U}_i$ by a one-step transition σ must be covered by the same cell $\mathcal{U}_j$. Inductively, two uncertainty sets U and U' belong to a common cell of $\mathcal{C}_\alpha^P$ if and only if U and U' are preemption consistent, and two future uncertainty sets that can be reached from U and U' by a given observable string are again preemption consistent.

The partial-observation preemption cover $\mathcal{C}_\alpha^P$ differs from its full-observation counterpart in [4] in two aspects. First, $\mathcal{C}_\alpha^P$ is defined on $\mathcal{U}(X)$, not on X; this is due to state uncertainty caused by partial observation. Second, in condition (ii) of $\mathcal{C}_\alpha^P$, only observable events in Σ_o are considered, not Σ; this is to generate partial-observation local preemptors whose state transitions are triggered only by observable events. We call $\mathcal{C}_\alpha^P$ a *partial-observation preemption congruence* if $\mathcal{C}_\alpha^P$ happens to be a partition on $\mathcal{U}(X)$.

Having defined a partial-observation preemption cover $\mathcal{C}_\alpha^P$ on $\mathcal{U}(X)$, we construct a generator $\mathbf{J}_\alpha = (I_\alpha, \Sigma_o, \zeta_\alpha, i_{0,\alpha}, I_{m,\alpha})$ and two functions $\psi_\alpha : I_\alpha \to \{0, 1\}$ and $\psi_{\text{tick}} : I_\alpha \to \{0, 1\}$ as follows. Recall from (12) that $U_0 = U(\epsilon)$ and thus $x_0 \in U_0$

$$\text{(i)} \quad i_{0,\alpha} \in I_\alpha \text{ such that } U_0 \in \mathcal{U}_{i_{0,\alpha}} \tag{13}$$

$$\text{(ii)} \quad I_{m,\alpha} := \{i \in I_\alpha | (\exists U \in \mathcal{U}_i) X_m \cap U \neq \emptyset\} \tag{14}$$

$$\text{(iii)} \quad \zeta_\alpha : I_\alpha \times \Sigma_o \to I_\alpha \text{ with } \zeta_\alpha(i, \sigma) = j$$

$$\text{if } (\exists U \in \mathcal{U}_i)\, \hat{\xi}(U, \sigma) \in \mathcal{U}_j \tag{15}$$

$$\text{(iv)} \quad \psi_\alpha(i) = 1 \text{ iff } (\exists U \in \mathcal{U}_i)(\exists x \in U)\, \xi(x, \alpha)! \tag{16}$$

$$\text{(v)} \quad \psi_{\text{tick}}(i) = 1 \text{ iff } (\exists U \in \mathcal{U}_i)\, E_{\text{tick}}(U) = 1. \tag{17}$$

The function $\psi_\alpha(i) = 1$ means that forcible event α is defined at state i of $\mathbf{J}_\alpha$, and the function $\psi_{\text{tick}}(i) = 1$ means that event tick is eligible to occur and its occurrence is not preempted at state i of $\mathbf{J}_\alpha$. Note that owing to cell overlapping, the choices of $i_{0,\alpha}$ and ζ_α may not be unique, and consequently $\mathbf{J}_\alpha$ may not be unique. In that case, we simply pick an arbitrary instance of $\mathbf{J}_\alpha$.

Finally, we define the *partial-observation local preemptor* $\mathbf{LOC}_\alpha^P = (Y_\alpha, \Sigma_\alpha, \eta_\alpha, y_{0,\alpha}, Y_{m,\alpha})$ as follows.

Step (i): $Y_\alpha = I_\alpha$, $y_{0,\alpha} = i_{0,\alpha}$, and $Y_{m,\alpha} = I_{m,\alpha}$. Thus, the function ψ_α is $\psi_\alpha : Y_\alpha \to \{0, 1\}$ and the function ψ_{tick} is $\psi_{\text{tick}} : Y_\alpha \to \{0, 1\}$.

Step (ii): $\Sigma_\alpha = \{\alpha, \text{tick}\} \cup \Sigma_{\text{com},\alpha}$, where

$$\Sigma_{\text{com},\alpha} := \{\sigma \in \Sigma_o \setminus \{\alpha, \text{tick}\} \mid (\exists i, j \in I_\alpha)\, i \neq j \;\&$$
$$\zeta_\alpha(i, \sigma) = j\}.$$

Thus, $\Sigma_{\text{com},\alpha}$ is the set of observable events that are not merely selfloops in $\mathbf{J}_\alpha$ (i.e., these events will cause state changes in $\mathbf{LOC}_\alpha^P$). It holds by definition that $\{\alpha, \text{tick}\} \subseteq \Sigma_\alpha \subseteq \Sigma_o \cup \{\alpha, \text{tick}\}$, and $\Sigma_{\text{com},\alpha}$ represents the set of *communication events* that need to be communicated to $\mathbf{LOC}_\alpha^P$. Note that a communication event $\sigma \in \Sigma_{\text{com},\alpha}$ can be nonforcible or nonprohibitible.

Step (iii): If $\alpha \in \Sigma_o$, then $\eta_\alpha = \zeta_\alpha|_{Y_\alpha \times \Sigma_\alpha} : Y_\alpha \times \Sigma_\alpha \to Y_\alpha$, i.e., η_α is the restriction of ζ_α to $Y_\alpha \times \Sigma_\alpha$. If $\alpha \in \Sigma_{uo}$, first obtain $\eta_\alpha = \zeta_\alpha|_{Y_\alpha \times \Sigma_\alpha}$, then add α-selfloops $\eta_\alpha(y, \alpha) = y$ to those $y \in Y_\alpha$ with $\psi_\alpha(y) = 1$.

Step (iv): If tick $\in \Sigma_{uo}$, then add tick-selfloops $\eta_\alpha(y, \text{tick}) = y$ to those $y \in Y_\alpha$ with $\psi_{\text{tick}}(y) = 1$.

Lemma 2: The generator $\mathbf{LOC}_\alpha^P$ is a partial-observation local preemptor for α, i.e., (5) and (6) hold.

For the proof of Lemma 2, see [14].

By the same procedure, we generate a set of partial-observation local preemptors $\mathbf{LOC}_\alpha^P$, one for each forcible event $\alpha \in \Sigma_{\text{for}}$. We will verify in the following that these generated preemptors collectively achieve the same tick-preemptive action as $\mathbf{SUP}$ did.

B. Partial-Observation Localization of Disabling Action

Next, we turn to the localization of disabling action, which is analogous to the treatment in [6] for the untimed case. Fix an arbitrary prohibitible event $\beta \in \Sigma_{\text{hib}}$. Define $E_\beta : \mathcal{U}(X) \to \{0, 1\}$ according to

$$(\forall U \in \mathcal{U}(X))\, E_\beta(U) = 1 \text{ iff } (\exists x \in U)\xi(x, \beta)!.$$

So, $E_\beta(U) = 1$ if event β is enabled at some state $x \in U$. Also define $D_\beta : \mathcal{U}(X) \to \{0, 1\}$ according to

$$(\forall U \in \mathcal{U}(X))$$
$$D_\beta(U) = \begin{cases} 1, & \text{if } (\exists x \in U)\neg\xi(x, \beta)! \;\& \\ & ((\exists s \in \Sigma^*)\xi(x_0, s) = x \;\& \\ & \quad \hat{\xi}(U_0, Ps) = U \& \; \delta(q_0, s\beta)!) \\ 0, & \text{otherwise.} \end{cases}$$

Hence, $D_\beta(U) = 1$ if β is disabled at some state $x \in U$. Now, define $M : \mathcal{U}(X) \to \{0, 1\}$ by $M(U) = 1$ iff there exists $x \in U$ such that $x \in X_m$, and $T : \mathcal{U}(X) \to \{0, 1\}$ by $T(U) = 1$ iff there exists $s \in \Sigma^*$ such that $\xi(x_0, s) \in U$, $\hat{\xi}(U_0, Ps) = U$, and $\delta(q_0, s) \in Q_m$.

We define the *control consistency relation* $\mathcal{R}_\beta^C \subseteq \mathcal{U}(X) \times \mathcal{U}(X)$ with respect to β according to $(U, U') \in \mathcal{R}_\beta^C$ iff

$$E_\beta(U) \cdot D_\beta(U') = 0 = E_\beta(U') \cdot D_\beta(U)$$
$$T(U) = T(U') \Rightarrow M(U) = M(U').$$

Let I_β be some index set, and $\mathcal{C}_\beta^C = \{\mathcal{U}_i \subseteq \mathcal{U}(X) | i \in I_\beta\}$ a cover on $\mathcal{U}(X)$. We say that $\mathcal{C}_\beta^C$ is a *partial-observation control cover* with respect to β if

(i) $(\forall i \in I_\beta, \forall U, U' \in \mathcal{U}_i)\, (U, U') \in \mathcal{R}_\beta^C$

(ii) $(\forall i \in I_\beta, \forall \sigma \in \Sigma_o)(\exists U \in \mathcal{U}_i)\hat{\xi}(U, \sigma) \neq \emptyset$

$$\Rightarrow \big((\exists j \in I_\beta)(\forall U' \in \mathcal{U}_i)\hat{\xi}(U', \sigma) \neq \emptyset \Rightarrow \hat{\xi}(U', \sigma) \in \mathcal{U}_j\big).$$

With the control cover $\mathcal{C}_\beta^C$ on $\mathcal{U}(X)$, we construct, by the Steps (i)–(iii) mentioned earlier for a local preemptor, a partial-observation local controller $\mathbf{LOC}_\beta^C = (Y_\beta, \Sigma_\beta, \eta_\beta, y_{0,\beta}, Y_{m,\beta})$ for prohibitible event β. Here, the event set Σ_β is $\Sigma_\beta = \{\beta\} \cup \Sigma_{\text{com},\beta}$, where

$$\Sigma_{\text{com},\beta} := \{\sigma \in \Sigma_o \setminus \{\beta\} \mid (\exists i, j \in I_\beta) i \neq j, \zeta_\beta(i, \sigma) = j\}.$$

It holds by definition that $\{\beta\} \subseteq \Sigma_\beta \subseteq \Sigma_o \cup \{\beta\}$, and $\Sigma_{\text{com},\beta}$ represents the set of *communication events* that need to be communicated to $\mathbf{LOC}_\beta^C$. Similar to the events in $\Sigma_{\text{com},\alpha}$, a communication event $\sigma \in \Sigma_{\text{com},\beta}$ can be nonforcible or nonprohibitible.

Lemma 3: The generator $\mathbf{LOC}_\beta^C$ is a partial-observation local controller for prohibitile event β.

For the proof of Lemma 3, see [6, Lemma 2].

By the same procedure, we generate a set of partial-observation local controllers $\mathbf{LOC}_\beta^C$, one for each prohibitive event $\beta \in \Sigma_{\text{hib}}$. We will verify in the following that these generated controllers collectively achieve the same disabling action as $\mathbf{SUP}$ did.

C. Main Result

Here is the main result of this section, which states that the collective behavior of the partial-observation local preemptors and local controllers generated by the localization procedure above is identical to the monolithic controllable and observable $\mathbf{SUP}$.

Theorem 1: The set of partial-observation local preemptors $\{\mathbf{LOC}_\alpha^P | \alpha \in \Sigma_{\text{for}}\}$ and the set of partial-observation local controllers $\{\mathbf{LOC}_\beta^C | \beta \in \Sigma_{\text{hib}}\}$ constructed above solve the partial-observation supervisor localization problem, i.e.,

$$L(\mathbf{G}) \cap L(\mathbf{LOC}) = L(\mathbf{SUP}) \tag{18}$$

$$L_m(\mathbf{G}) \cap L_m(\mathbf{LOC}) = L_m(\mathbf{SUP}) \tag{19}$$

where $L(\mathbf{LOC})$ and $L_m(\mathbf{LOC})$ are as defined in (8) and (9), respectively.

Since for every partial-observation preemption cover (respectively control cover), the presented procedure constructs a local preemptor (respectively local controller), Theorem 1 asserts that every set of preemption and control covers together generates a solution to the partial-observation supervisor localization problem. The localization algorithm in [6] for untimed DES can easily be adapted in the current TDES case, the only modification being to use the new definitions of partial-observation preemption and control consistency given in Sections IV-A and IV-B. The complexity of the localization algorithm is $O(n^4)$; since the size n of $\mathcal{U}(X)$ is $n \leq 2^{|X|}$ in general, the algorithm is exponential in $|X|$.

Remark 1: As presented in [1]–[3], for large-scale timed DES in practice, we may combine our proposed partial-observation supervisor localization with an efficient decentralized/hierarchical supervisor synthesis approach [15], by exploiting modularities that often exist in practical systems and extending the approach in [15] from untimed to timed DES. A systematic investigation on this topic is left for our future work.

Having these obtained partial-observation local preemptors/controllers, by the allocation policy described in Section III, we build a distributed control architecture for the multicomponent TDES $\mathbf{G}$ in (3). As asserted by Theorem 1, the distributed controlled behavior is identical to the monolithic one, as represented by $\mathbf{SUP}$.

Proof of Theorem 1: First, we prove ($\subseteq$) of (18), i.e., $L(\mathbf{G}) \cap L(\mathbf{LOC}) \subseteq L(\mathbf{SUP})$, by induction on the length of strings.

For the *base step*, note that none of $L(\mathbf{G})$, $L(\mathbf{LOC})$, and $L(\mathbf{SUP})$ is empty; and thus the empty string ϵ belongs to all of them. For the *inductive step*, suppose that $s \in L(\mathbf{G}) \cap L(\mathbf{LOC})$, $s \in L(\mathbf{SUP})$, and $s\sigma \in L(\mathbf{G}) \cap L(\mathbf{LOC})$ for arbitrary event $\sigma \in \Sigma$; we must show that $s\sigma \in L(\mathbf{SUP})$. Since $\Sigma = \Sigma_{uc} \dot\cup \Sigma_{\text{hib}} \dot\cup \{\text{tick}\}$, σ may belong to Σ_{uc}, Σ_{hib}, or be equal to tick. The proof for $\sigma \in \Sigma_{uc}$ and $\sigma \in \Sigma_{\text{hib}}$ is similar to that in [6] for untimed DES; in the following, we consider the case $\sigma = \text{tick}$, which is specific to TDES.

By the hypothesis that $s, s.\text{tick} \in L(\mathbf{LOC})$, for every forcible event $\alpha \in \Sigma_{\text{for}}$, $s, s.\text{tick} \in P_\alpha^{-1} L(\mathbf{LOC}_\alpha^P)$, i.e., $P_\alpha(s), P_\alpha(s).\text{tick} \in L(\mathbf{LOC}_\alpha^P)$. Let $y = \eta_\alpha(y_{0,\alpha}, P_\alpha(s))$; then $\eta_\alpha(y, \text{tick})!$. Since tick may be observable or unobservable, we consider the following two cases.

i) $\text{tick} \in \Sigma_{uo}$. It follows from the construction rule (iv) of $\mathbf{LOC}_\alpha^P$ that $\eta_\alpha(y, \text{tick})!$ implies that for the state $i \in I$ of the generator $\mathbf{J}_\alpha$ corresponding to y (i.e., $i = \zeta_\alpha(i_0, P(s))$), there holds $\psi_{\text{tick}}(i) = 1$. By the definition of ψ_{tick} in (17), there exists an uncertainty set $U \in \mathcal{U}_i$ such that $E_{\text{tick}}(U) = 1$. Let $U' = \hat\xi(U_0, Ps)$; by (15) and $i = \zeta_\alpha(i_0, Ps)$, $U' \in \mathcal{U}_i$. According to (11), $\xi(x_0, s) \in U'$. Since U and U' belong to the same cell $\mathcal{U}_i$, by the definition of partial-observation preemption cover they must be preemption consistent, i.e., $(U, U') \in \mathcal{R}_\alpha^P$. Thus, $E_{\text{tick}}(U) \cdot F_\alpha(U') = 0$, which implies that $F_\alpha(U') = 0$. The latter means that for all state $x \in U'$, (a) $\neg\xi(x, \alpha)!$, or (b) $\xi(x, \text{tick})!$, or (c) $\neg(\exists s' \in \Sigma^*)$ $(\xi(x_0, s') = x$, $\hat\xi(U_0, Ps') = U'$ and $\delta(q_0, s'.\text{tick})!)$. First, Case (c) is impossible, because we already have $\xi(x_0, s) \in U'$, $\hat\xi(U_0, Ps) = U'$ and $s.\text{tick} \in L(\mathbf{G})$ (namely string s falsifies the logical statement of Case (c)). Next, Case (b) means directly that $s.\text{tick} \in L(\mathbf{SUP})$. Finally, Case (a) implies that $\alpha \notin \text{Elig}_{L_m(\mathbf{SUP})}(s)$; note that this holds for all $\beta \in \Sigma_{\text{for}}$. Hence $\text{Elig}_{L_m(\mathbf{SUP})}(s) \cap \Sigma_{\text{for}} = \emptyset$. Then, by the fact that $L_m(\mathbf{SUP})$ is controllable and $s.\text{tick} \in L(\mathbf{G})$, $\text{tick} \in \text{Elig}_{L_m(\mathbf{SUP})}(s)$, i.e., $s.\text{tick} \in L(\mathbf{SUP})$.

ii) $\text{tick} \in \Sigma_o$. In this case, for the state $i \in I$ of the generator $\mathbf{J}_\alpha$ corresponding to y (i.e., $i = \zeta_\alpha(i_0, P(s))$), there holds $\zeta_\alpha(i, \text{tick})!$. By the definition of ζ_α in (15), there exists an uncertainty set $U \in \mathcal{U}_i$ such that $\hat\xi(U, \text{tick})!$. So $E_{\text{tick}}(U) = 1$. The rest of the proof is identical to Case (i) earlier, and we conclude that $s.\text{tick} \in L(\mathbf{SUP})$ as well.

The ($\supseteq$) direct of (18), as well as (19), can be established similarly to [6]. ■

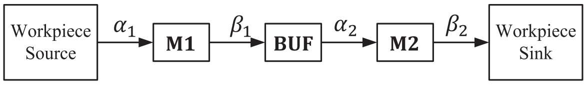

Fig. 1. Workcell: system configuration.

V. Case Study: Timed Workcell

We illustrate the proposed partial-observation supervisor localization procedure by a timed workcell example, adapted from [11, Ch. 9]. As displayed in Fig. 1, the workcell consists of two machines $\mathbf{M1}$ and $\mathbf{M2}$, linked by a one-slot buffer $\mathbf{BUF}$; additionally, a worker $\mathbf{WK}$ is responsible for repairing $\mathbf{M1}$ and $\mathbf{M2}$. The workcell operates as follows. Initially, the buffer is empty. With the event α_1, $\mathbf{M1}$ takes a workpiece from the infinite workpiece source. Subsequently, $\mathbf{M1}$ either breaks down (event λ_1) or successfully completes its work cycle, deposits the workpiece in the buffer (event β_1). $\mathbf{M2}$ operates similarly, but takes its workpiece from the buffer (event α_2), and deposits it when finished in the infinite workpiece sink. If a machine $\mathbf{M}_i$, $i = 1$ or 2, breaks down (event λ_i), then the worker $\mathbf{WK}$ will start to repair the machine (event μ_i) and finish the repair (event η_i) in due time. Assign lower and upper time bounds to each event, with notation (event, lower bound, upper bound), as follows:

$\mathbf{M1}$'s timed events :

$$(\alpha_1, 0, \infty)\ (\beta_1, 1, 2)\ (\lambda_1, 0, 2)\ (\mu_1, 0, \infty)\ (\eta_1, 1, \infty)$$

$\mathbf{M2}$'s timed events :

$$(\alpha_2, 0, \infty)\ (\beta_2, 1, 1)\ (\lambda_2, 0, 1)\ (\mu_2, 0, \infty)\ (\eta_2, 2, \infty)$$

$\mathbf{WK}$'s timed events :

$$(\mu_1, 0, \infty)\ (\eta_1, 1, 2)\ (\mu_2, 0, \infty)\ (\eta_2, 2, 3).$$

Then, the TDES models of the two machines and the worker can be generated [11]; their joint behavior is the composition of the three TDES, which is the plant $\mathbf{PLANT}$ to be controlled, i.e.,

$$\mathbf{PLANT} = \mathbf{Comp}(\mathbf{M1}, \mathbf{M2}, \mathbf{WK}).$$

Note that $\mathbf{Mi}$ $(i = 1, 2)$ shares events μ_i and η_i with $\mathbf{WK}$; so according to the composition rule described in Section II, the lower and upper bounds of μ_i and η_i are unified as: $(\mu_1, 0, \infty)$ $(\eta_1, 1, 2)$ $(\mu_2, 0, \infty)$ $(\eta_2, 2, 3)$.

To impose behavioral constraints on the two machine's joint behavior, we take $\Sigma_{\text{for}} = \Sigma_{\text{hib}} = \{\alpha_i, \mu_i | i = 1, 2\}$ and $\Sigma_{uc} = \{\beta_i, \lambda_i, \eta_i | i = 1, 2\}$. We impose the following control specifications: (S1) $\mathbf{BUF}$

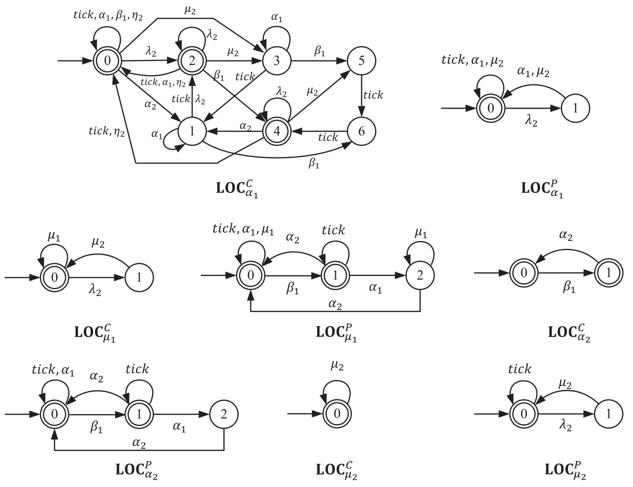

Fig. 2. Local preemptors and local controllers under partial observation with $\Sigma_{uo} = \{\mu_1, \eta_2\}$.

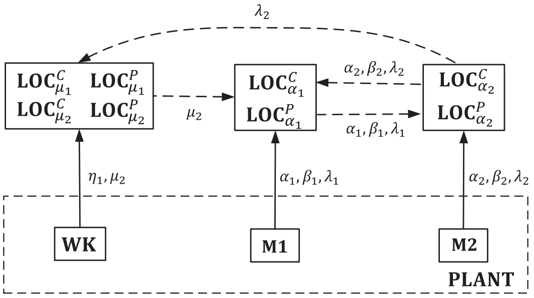
λ_2
$\mathbf{LOC}^C_{\mu_1}$ $\mathbf{LOC}^P_{\mu_1}$
$\mathbf{LOC}^C_{\mu_2}$ $\mathbf{LOC}^P_{\mu_2}$
μ_2
$\mathbf{LOC}^C_{\alpha_1}$
$\mathbf{LOC}^P_{\alpha_1}$
$\alpha_2, \beta_2, \lambda_2$
$\alpha_1, \beta_1, \lambda_1$
$\mathbf{LOC}^C_{\alpha_2}$
$\mathbf{LOC}^P_{\alpha_2}$
η_1, μ_2
$\alpha_1, \beta_1, \lambda_1$
$\alpha_2, \beta_2, \lambda_2$
WK
M1
M2
PLANT

Fig. 3. Distributed control architecture under partial observation.

must not overflow or underflow; and (S2) if $\mathbf{M2}$ goes down, its repair must be started "immediately," and prior to starting repair of $\mathbf{M1}$ if $\mathbf{M1}$ is currently down. These two specifications are formalized as generators $\mathbf{BUFSPEC}$ and $\mathbf{BRSPEC}$. So the overall specification imposed on the $\mathbf{PLANT}$ is represented by $\mathbf{SPEC} = \mathbf{BUFSPEC}||\mathbf{BRSPEC}$, where "||" denotes the synchronous product of two generators [11].

For partial observation, we set $\Sigma_{uo} = \{\mu_1, \eta_2\}$, namely the event of starting repair of $\mathbf{M1}$ and event of finishing the repair of $\mathbf{M2}$ are unobservable. Note that μ_1 is both prohibitible and forcible, whereas η_2 is uncontrollable. We first compute as in (4) the controllable and observable behavior $\mathbf{SUP}$, which has 77 states and 169 transitions. Then, we apply the proposed partial-observation supervisor localization procedure to construct partial-observation local preemptors and partial-observation local controllers for each forcible event and each prohibitile event. The computation is done by an algorithm adapted from [6], as discussed in Section IV-C. The results are displayed in Fig. 2; it is inspected from the TTG of the local preemptors/controllers that none of the unobservable events (in $\Sigma_{uo} = \{\mu_1, \eta_2\}$) causes state change. It is also verified that the collective controlled behavior of these local preemptors and controllers is identical to the controllable and observable behavior $\mathbf{SUP}$.

Finally, according to the allocation policy described in Section III, we build a distributed control architecture for the timed workcell, as displayed in Fig. 3. Here, $\hat{\Sigma}_{\text{for},\mathbf{WK}} = \hat{\Sigma}_{\text{hib},\mathbf{WK}} = \{\mu_1, \mu_2\}$, $\hat{\Sigma}_{\text{for},\mathbf{M1}} = \hat{\Sigma}_{\text{hib},\mathbf{M1}} = \{\alpha_1\}$, and $\hat{\Sigma}_{\text{for},\mathbf{M2}} = \hat{\Sigma}_{\text{hib},\mathbf{M2}} = \{\alpha_2\}$. A local preemptor/controller either directly observes an observable event generated by the plant component owning it, as denoted by solid lines in Fig. 3, or imports an observable event by communication from other local preemptors/controllers, as denoted by the dashed lines. Those events imported by communication may be subject to delay when using physical channels, and we present in [14] a synthesis approach combining timed relative co-observability and the partial-observation localization procedure, which constructs partial-observation local controllers and preemptors tolerant of the subjected communication delays.

VI. Conclusion

In this technical note, we have developed a partial-observation supervisor localization procedure to solve the distributed control problem of multicomponent TDES. A synthesized monolithic supervisor is decomposed into a set of partial-observation local controllers and a set of partial-observation local preemptors, whose state changes are caused only by observable events. We have proved that the resulting local controllers/preemptors collectively achieve the same controlled behavior as the monolithic supervisor does.

In future research, we shall study an alternative approach that first synthesizes the full-observation centralized supervisor and then performs localization to respect the observable event subsets specified *a priori*.

References

[1] K. Cai and W. Wonham, "Supervisor localization: A top-down approach to distributed control of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 55, no. 3, pp. 605–618, Mar. 2010.

[2] K. Cai and W. Wonham, "Supervisor localization for large discrete-event systems: Case study production cell," *Int. J. Adv. Manuf. Technol.*, vol. 50, no. 9–12, pp. 1189–1202, 2010.

[3] K. Cai and W. Wonham, *Supervisor Localization: A Top-Down Approach to Distributed Control of Discrete-Event Systems* (Lecture Notes in Control and Information Sciences), vol. 459. New York, NY, USA: Springer, 2015.

[4] R. Zhang, K. Cai, Y. Gan, Z. Wang, and W. Wonham, "Supervision localization of timed discrete-event systems," *Automatica*, vol. 49, no. 9, pp. 2786–2794, 2013.

[5] B. Brandin and W. Wonham, "Supervisory control of timed discrete-event systems," *IEEE Trans. Autom. Control*, vol. 39, no. 2, pp. 329–342, Feb. 1994.

[6] R. Zhang, K. Cai, and W. Wonham, "Supervisor localization of discrete-event systems under partial observation," *Automatica*, vol. 81, pp. 142–147, 2017.

[7] K. Cai, R. Zhang, and W. Wonham, "Relative observability of discrete-event systems and its supremal sublanguages," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 659–670, Mar. 2015.

[8] R. Zhang and K. Cai, "Supervisor localization of timed discrete-event systems under partial observation," in *Proc. 55th IEEE Conf. Decis. Control*, Las Vegas, NV, USA, 2016, pp. 4752–4757.

[9] K. Cai, R. Zhang, and W. Wonham, "Relative observability and coobservability of timed discrete-event systems," *IEEE Trans. Autom. Control*, vol. 61, no. 11, pp. 3382–3395, Nov. 2016.

[10] F. Lin and W. Wonham, "Supervisory control of timed discrete-event systems under partial observation," *IEEE Trans. Autom. Control*, vol. 40, no. 3, pp. 558–562, Mar. 1995.

[11] W. Wonham and K. Cai, *Supervisory Control of Discrete-Event Systems*, Springer, 2019.

[12] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. New York, NY, USA: Springer, 2008.

[13] J. Hopcroft, R. Motwani, and J. Ullman, *Introduction to Automata Theory Languages and Computation*. London, U.K.: Pearson Educ., 2014.

[14] R. Zhang and K. Cai, "Supervisor localization of timed discrete-event systems under partial observation and communication delay," Tech. Rep., 2019. [Online]. Available: http://arxiv.org/abs/1603.02023

[15] L. Feng and W. Wonham, "Supervisory control architecture for discrete-event systems," *IEEE Trans. Autom. Control*, vol. 53, no. 6, pp. 1449–1461, Jul. 2008.