12th IFAC/IEEE Workshop on Discrete Event Systems
Cachan, France. May 14-16, 2014

Local Modular Supervisory Control of Timed Discrete-Event Systems ⋆

Germano Schafaschek * **Max H. de Queiroz** * **José E. R. Cury** *

* *Automation and Systems Department, Federal University of Santa Catarina, Florianópolis, Brazil (e-mail: germano.s@posgrad.ufsc.br; {max.queiroz,jose.cury}@ufsc.br).*

Abstract: This paper presents a local modular approach which reduces computational efforts in the synthesis of supervisors for timed discrete-event systems. We exploit the modularity commonly inherent to large scale systems, constructing local models that comprise only those parts of the system affected by the given specifications. Modular supervisors are then designed over these local models, and conditions are presented under which their concurrent action achieves nonblocking optimal global behavior.

Keywords: Supervisory control; timed discrete-event systems; modular control; composed systems; large scale systems.

1. INTRODUCTION

State explosion is a recurrent problem concerning the modelling of large scale Discrete-Event Systems (DES). A consequence of this problem is the complexity of computing controllers for such systems. In the Ramadge-Wonham (RW) supervisory control framework — Ramadge and Wonham (1987) — the model of the plant to be controlled is commonly obtained through the composition of several automata, and its size grows exponentially with the number of components. This poses an obstacle for the synthesis of supervisors in many applications, motivating efforts to overcome such difficulties. In Wonham and Ramadge (1988), a modular approach is presented which allows global optimality to be achieved by the independent synthesis of multiple supervisors, one for each different specification. In Queiroz and Cury (2000, 2002), this strategy is enhanced by taking advantage of the modularity of the system, whose global behavior results from the shuffle product of smaller subsystems. Local plants are obtained by composing only those subsystems affected by each specification, so that supervisors may be synthesized locally, i. e., considering only the corresponding local model.

In the context of Timed Discrete-Event Systems (TDES), the consideration of an extra event representing the passage of time may cause a dramatic increase in the size of the models, further aggravating the issue of state explosion. In Brandin and Wonham (1994, 1993), the RW supervisory control techniques are extended to TDES. In Saadatpoor and Wonham (2007), a state based synthesis approach for timed supervisors using binary decision diagrams (BDD) is proposed. Although supervisors are synthesized in a centralized way, exploiting structural information for a proper BDD representation may significantly reduce computational effort. In Zhang et al. (2013), supervisor localization is extended to the timed context; this approach is yet to be combined with a modular strategy so as to be suitable for large scale TDES.

In Brandin and Wonham (1993), the results on decentralized supervisory control from Lin and Wonham (1988) are extended to the timed context, yielding the so-called modular supervision of TDES under partial observation. It consists in a top-down approach, starting from a global model of the system and building local subplants by projection of the global behavior onto subalphabets. With each subplant is associated a specification to be implemented by a local supervisor, whose success at the global level depends on the property of normality. The overall computational effort required for the design of such supervisors is generally greater in comparison with equivalent centralized designs.

In this paper, we extend the local modular supervisory control from Queiroz and Cury (2000) to the timed context. The system's global model is again seen as the composition of smaller subsystems, which in this case are quasi independent, being synchronized only by a common global clock. Instead of obtaining local models by projection of the global behavior, in our bottom-up approach a local plant is built for each specification by the synchronous composition of only the affected subsystems. Modular supervisors are then synthesized considering just the local plants. The global model of the plant is not required to be computed during the synthesis procedure, and the property of normality is not regarded, being guaranteed by construction. Sufficient conditions are established for the absence of blocking as well as for global optimality.

2. PRELIMINARIES

In this section, some preliminary definitions and results are presented. For a more detailed discussion on the theory of TDES, please refer to Brandin and Wonham (1994)

⋆ The first author has received financial support from the Brazilian National Council for Scientific and Technological Development (CNPq).

or Wonham (July 2013). Some prior knowledge about the supervisory control of untimed DES is assumed. See Ramadge and Wonham (1987) and Wonham and Ramadge (1988) for the seminal work on the subject.

2.1 Languages

An *alphabet* Σ is a finite, nonempty set of symbols. A *string* s is a finite sequence of such symbols. The *empty string* is denoted by ε. Σ^+ denotes the set of all nonempty strings formed by elements of Σ, and $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$. Given strings $s, u \in \Sigma^*$, u is said to be a *prefix* of s in case $\exists v \in \Sigma^* \mid s = uv$. A *language* L over Σ is any subset of Σ^*. The *prefix-closure* of L is defined as $\overline{L} = \{u \in \Sigma^* \mid \exists v \in \Sigma^*, uv \in L\}$. For each $s \in L$ define the set $\Sigma_L(s) = \{\sigma \in \Sigma \mid s\sigma \in \overline{L}\}$.

For alphabets $\Sigma_i \subset \Sigma$, the *natural projection* $P_i : \Sigma^* \to \Sigma_i^*$ is defined, $\forall \sigma \in \Sigma$ and $\forall s \in \Sigma^*$, by

$$P_i(\varepsilon) = \varepsilon\,;$$
$$P_i(s\sigma) = \begin{cases} P_i(s) & \text{if } \sigma \notin \Sigma_i\,, \\ P_i(s)\sigma & \text{if } \sigma \in \Sigma_i\,. \end{cases}$$

The definition can be extended to languages, with $P_i : 2^{\Sigma^*} \to 2^{\Sigma_i^*}$ and $P_i^{-1} : 2^{\Sigma_i^*} \to 2^{\Sigma^*}$ given by

$$P_i(L) = \{s_i \in \Sigma_i^* \mid \exists s \in L, P_i(s) = s_i\}\,;$$
$$P_i^{-1}(L_i) = \{s \in \Sigma^* \mid P_i(s) \in L_i\}\,.$$

Given a collection of languages L_i, $i \in \{1, \ldots, n\}$, their *synchronous product* is defined as

$$\mathop{||}_{i=1}^{n} L_i = \bigcap_{i=1}^{n} P_i^{-1}(L_i)\,.$$

2.2 Timed Discrete-Event Systems

As in Brandin and Wonham (1994), let a TDES be modelled by a finite-state automaton $\mathbf{G} = (Q, \Sigma, \delta, q_0, Q_m)$, with *closed behavior* $L(\mathbf{G}) = \{s \in \Sigma^* \mid \exists q \in Q, \delta(q_0, s) = q\}$ and *marked behavior* $L_m(\mathbf{G}) = \{s \in \Sigma^* \mid \exists q \in Q_m, \delta(q_0, s) = q\}$. Time in the system is measured with a global digital clock, and an event *tick* is introduced which represents the passing of one unit of clock time. The event set is then given by $\Sigma = \Sigma_{\text{act}} \,\dot\cup\, \{tick\}$, where Σ_{act} consists of the "regular" events of the system. Each element of Σ_{act} is equipped with a *lower time bound* $\ell_\sigma \in \mathbb{N}$ and an *upper time bound* $u_\sigma \in \mathbb{N} \cup \{\infty\}$, representing, respectively, the minimum and maximum delays before an event occurs. The set Σ_{act} is partitioned as $\Sigma_{\text{act}} = \Sigma_{\text{rem}} \,\dot\cup\, \Sigma_{\text{spe}}$, where $\Sigma_{\text{rem}} = \{\sigma \in \Sigma_{\text{act}} \mid u_\sigma = \infty\}$ is the set of *remote* events and $\Sigma_{\text{spe}} = \{\sigma \in \Sigma_{\text{act}} \mid u_\sigma < \infty\}$ is the set of *prospective* events. Any event $\sigma \in \Sigma_{\text{act}}$ can occur provided it remains enabled during at least ℓ_σ occurrences of *tick*. A remote event may stay enabled indefinitely without occurring. A prospective event, on the other hand, becomes *imminent* if it remains enabled for u_σ units of time; it then must occur (or be disabled) before the next *tick* of the clock.

For brevity, we shall write $\Sigma_{\mathbf{G}}(s) = \Sigma_{L(\mathbf{G})}(s)$ for the set of *eligible* events in $\mathbf{G}$ upon the occurrence of string s. Equivalently, $\Sigma_{\mathbf{G}}(q)$ denotes the set of eligible events in state $q \in Q$.

Remark 1: Recall from Brandin and Wonham (1994) that, for the sake of physical plausibility, a TDES $\mathbf{G}$ must be *activity-loop-free* (ALF), that is,

$$\forall q \in Q,\ s \in \Sigma_{\text{act}}^+ \Rightarrow \delta(q, s) \neq q\,.$$

Violating such a condition would imply allowing for the unrealistic possibility that a sequence of events is repeated indefinitely without any occurrence of *tick*, i. e., within a unit time interval. Furthermore, by definition, if $\Sigma_{\mathbf{G}}(s) \cap \Sigma_{\text{act}} = \emptyset$ for some $s \in L(\mathbf{G})$, then $tick \in \Sigma_{\mathbf{G}}(s)$. This, combined with the ALF condition and with the fact that Q is a finite set, means that the clock of the system never stops, which can be formally stated as

$$\forall s \in L(\mathbf{G}), \exists u \in \Sigma^* \mid \Sigma_{\mathbf{G}}(su) = \{tick\}\,.$$

Remark 2: As highlighted in Lin and Wonham (1995), if no *tick* is possible in $\mathbf{G}$ after a string s, then there must be at least one prospective event eligible (in fact, *imminent*) after s. According to Remark 1, there can be at most a *finite* string of events following s before the *tick* is enabled. To summarize, one can conclude that

$$\forall s \in L(\mathbf{G}), tick \notin \Sigma_{\mathbf{G}}(s) \Rightarrow \exists x \in \Sigma_{\text{spe}}^+ \,\big|\, sx \in L(\mathbf{G}) \ \&\ tick \in \Sigma_{\mathbf{G}}(sx)\,.$$

It will be useful to bear in mind that a TDES $\mathbf{G}$ is constructed from an activity generator $\mathbf{G}_{\text{act}}$ (with alphabet Σ_{act}) and the aforementioned time bounds on events — see Brandin and Wonham (1994) for details. So, $\mathbf{G}$ can be graphically represented by both its *activity transition graph* (ATG), namely the transition graph of $\mathbf{G}_{\text{act}}$, and its *timed transition graph* (TTG), which is the ordinary transition graph of $\mathbf{G}$ (with the explicit display of *tick*).

2.3 Composition of TDES

We first review the definition of *synchronous composition* of automata. The *accessible part* of an automaton $\mathbf{A} = (Q, \Sigma, \delta, q_0, Q_m)$, denoted $Ac(\mathbf{A})$, is obtained by eliminating all of its unreachable states, i. e., by replacing Q with $Q_{ac} = \{q \in Q \mid \exists s \in \Sigma^*,\ \delta(q_0, s) = q\}$. For any two automata $\mathbf{A}_1 = (Q_1, \Sigma_1, \delta_1, q_{1_0}, Q_{1\text{m}})$ and $\mathbf{A}_2 = (Q_2, \Sigma_2, \delta_2, q_{2_0}, Q_{2\text{m}})$, we have

$$\mathbf{A}_1 || \mathbf{A}_2 = Ac\big(Q_1 \times Q_2, \Sigma_1 \cup \Sigma_2, \delta_{12}, (q_{1_0}, q_{2_0}), Q_{1\text{m}} \times Q_{2\text{m}}\big),$$

with δ_{12} defined as follows. Given $q = (q_1, q_2) \in Q_1 \times Q_2$ and $\sigma \in \Sigma_1 \cup \Sigma_2$, $\delta_{12}(q, \sigma)$ is equal to

$$\begin{cases} \big(\delta_1(q_1, \sigma), \delta_2(q_2, \sigma)\big) & \text{if } \sigma \in \Sigma_{\mathbf{G}_1}(q_1) \cap \Sigma_{\mathbf{G}_2}(q_2)\,; \\ \big(\delta_1(q_1, \sigma), q_2\big) & \text{if } \sigma \in \Sigma_{\mathbf{G}_1}(q_1) - \Sigma_2\,; \\ \big(q_1, \delta_2(q_2, \sigma)\big) & \text{if } \sigma \in \Sigma_{\mathbf{G}_2}(q_2) - \Sigma_1\,; \\ \text{undefined} & \text{otherwise}\,. \end{cases}$$

This way, one has $L(\mathbf{A}_1 \,||\, \mathbf{A}_2) = L(\mathbf{A}_1) \,||\, L(\mathbf{A}_2)$ and $L_m(\mathbf{A}_1 \,||\, \mathbf{A}_2) = L_m(\mathbf{A}_1) \,||\, L_m(\mathbf{A}_2)$. Please notice that, here and throughout the paper, the symbol $||$ is used to denote both the synchronous product of languages and the synchronous composition of automata, the distinction being clear from the context.

Secondly, let us recall from Brandin and Wonham (1994) the composition operation *comp*, defined as follows. Given two TDES $\mathbf{G}_1$ and $\mathbf{G}_2$, $\mathbf{G} = comp(\mathbf{G}_1, \mathbf{G}_2)$ is a TDES so that $\mathbf{G}_{\text{act}} = \mathbf{G}_{1_{\text{act}}} \,||\, \mathbf{G}_{2_{\text{act}}}$. The time bounds on the events of $\mathbf{G}$ are determined by the following rule: if $\sigma \in (\Sigma_{1_{\text{act}}} - \Sigma_{2_{\text{act}}})$ or $\sigma \in (\Sigma_{2_{\text{act}}} - \Sigma_{1_{\text{act}}})$, ℓ_σ and u_σ remain unchanged; otherwise, if $\sigma \in \Sigma_{1_{\text{act}}} \cap \Sigma_{2_{\text{act}}}$, then $\ell_\sigma = \max\{\ell_{1_\sigma}, \ell_{2_\sigma}\}$ and $u_\sigma = \min\{u_{1_\sigma}, u_{2_\sigma}\}$, provided $\ell_\sigma \leq u_\sigma$. In case the latter rule is violated for any σ, the composition is considered undefined.