Invariance Under Scaling of Time Bounds in Timed Discrete-Event Systems

Sean E. Bourdon, W.M. Wonham
Systems Control Group
Dept. of Electrical and Computer Engineering
University of Toronto
Toronto, ON M5S 3G4
bourdon,wonham@control.toronto.edu

M. Lawford
Dept. of Computing and Software
McMaster University
Hamilton, ON L8S 4L8
lawford@mcmaster.ca

Abstract

A family of affine mappings on the time bounds for timed discrete-event systems is introduced. It is shown that unless these affine maps are in fact linear, then the timed activity transition graph (TATG) of an arbitrary timed discrete-event system (TDES) may not be preserved under the scaling operation. Moreover, it is shown that when the scaling is linear, the TATG is always preserved under scaling. We examine some applications of the result, including state space reduction and make connections to suboptimal supervisory controller synthesis and dense time scaling. Finally, we briefly discuss topics for future study including extensions to model-checking and efficient representation of TDES.

1 Introduction

Several paradigms have been proposed in order to introduce greater modeling flexibility and realism to the Ramadge-Wonham framework of discrete-event systems. Among these is the timed discrete-event systems (TDES) framework of Brandin and Wonham [6] in which the passage of time (in the global sense) is marked by a special event, *tick*. In this theory, time is uniformly quantized and the quantum used represents the elapsed time between successive *ticks* of an integer valued clock. The framework is meant to realistically model sampled systems. Dense time systems, while sometimes mathematically satisfying, are necessarily idealized.

When modeling a real-time system in the TDES framework for the purposes of implementing supervisory control, a faster *tick* rate lends itself to increased accuracy. However, in any physical system, there are limitations on the *tick* rate that may be used for the model. These constraints on the *tick* rate generally arise from two sources. The first of these is hardware limitations. For example, regardless of the technology used, our sampling rate is always bounded. The second constraint on the *tick* rate stems from the well-known state explosion problem. Namely, the size of the state space for a TDES model can grow rapidly as the *tick* rate increases. Hence, a TDES with a high *tick* density may not be very amenable to computation.

One way in which a model's suitability can be judged is via its timed activity transition graph (TATG). A given TDES **G** can be represented graphically using a timed transition graph, which displays all possible sequences of events that can occur within the definition of $\mathbf{G}$, including the *tick* transitions. In order to obtain the TATG from the TTG, we merely need to suppress all *tick* transitions. Thus, the TATG incorporates the constraints imposed on the event sequences of $\mathbf{G}$ by their time bounds.

In Section 3 of this paper, we focus on a specific family of transformations of TDES in which the time bounds for the activities in the original TDES $\mathbf{G}$ are affinely scaled to produce a new TDES $\mathbf{G}'$. We first consider the case where the *tick* rate is increased in the scaled system. It will be shown that when the time bounds are linearly scaled the TATGs for $\mathbf{G}$ and $\mathbf{G}'$ will be the same. Moreover, we shall prove that when the scaling is affine but not linear, the TATG for $\mathbf{G}'$ may not necessarily coincide with that of $\mathbf{G}$. These first results can assist in determining how a model should change with the introduction of faster components or better processors into the system. We then consider the effect of scaling a system so that *ticks* become slower and discuss application to model reduction. It is also shown that scaling commutes with composition of TDES, so that scaling can be used in a compositional model reduction scheme of the type described in [12] and [11].

Next, we add to the set of results from Section 3 and discuss connections to related work. The results obtained here allow us to relate our work to that of [8] on suboptimal supervisory control.

In Section 4, we examine a few interesting properties that arise in the limit as *ticks* become dense in the real line. Specifically, it is shown that the original definition of TDES breaks down, motivating the need for minor semantic changes.

The final section of the paper summarizes the results presented in the two preceding sections and discusses potential future topics of study.

2 Timed Discrete-Event Systems

As in [6], we model a TDES $\mathbf{G}$ in two steps. The first is to construct an activity transition structure $\mathbf{G_{act}} := (A, \Sigma, \xi, a_o)$, which is simply a finite deterministic automaton [13]. In this definition, A is the set of activities, $a_o \in A$ is the initial activity, and Σ represents the set of system events. The partial function $\xi : A \times \Sigma \to A$ is called the activity transition function and we write $\xi(a, \sigma)!$ whenever $\xi(a, \sigma)$ is defined.

The second step in constructing a TDES consists of introducing time into the ATG. In a TDES, time is measured via the *tick* of a discrete global clock, relative to which the enablement of each of the events in Σ is defined. Specifically, to each event $\sigma \in \Sigma$ we assign upper and lower time bounds $u_\sigma \in \mathbb{N}_o \cup \{\infty\}$ and $l_\sigma \in \mathbb{N}_o$, respectively, which satisfy $l_\sigma \leq u_\sigma$. Here, $\mathbb{N}_o := \mathbb{N} \cup \{0\}$, where $\mathbb{N}$ denotes the set of positive integers. An event $\sigma \in \Sigma$ must remain continuously enabled for at least l_σ, but no more than u_σ, *ticks* of the clock in order to occur (provided its occurrence is not preempted by the occurrence of another competing event). In the sequel, we refer to $\Sigma_{tim} := \{(\sigma, l_\sigma, u_\sigma) : \sigma \in \Sigma\}$ as the set of timed events of $\mathbf{G}$.

Following [6], we give the TDES $\mathbf{G}$ an automaton structure. For this, we need some further notation. Let $\Sigma_{spe} := \{\sigma \in \Sigma : u_\sigma < \infty\}$ and $\Sigma_{rem} := \{\sigma \in \Sigma : u_\sigma = \infty\}$. We say that an event σ is prospective (remote) if $\sigma \in \Sigma_{spe}$ ($\sigma \in \Sigma_{rem}$). It is clear from the above definitions that $\Sigma = \Sigma_{spe} \dot\cup \Sigma_{rem}$ and that a remote event has no deadline attached to it. Next, we introduce the special event *tick* to denote the passage of one unit of time in our global clock, and write $\Sigma_t := \Sigma \cup \{tick\}$. Finally, we define the *timer interval*, T_σ, for an event σ via

$$T_\sigma := \begin{cases} [0, u_\sigma], & \sigma \in \Sigma_{spe} \\ [0, l_\sigma], & \sigma \in \Sigma_{rem} \end{cases},$$

where $[j, k]$ denotes the set of integers i satisfying $j \le i \le k$.

We are now able to formally write $\mathbf{G} := (Q, \Sigma_t, \delta, q_o)$. The state set in this case is defined to be $Q = A \times \prod_{\sigma \in \Sigma} T_\sigma$. Similarly, we have that the initial state $q_0 := (a_o, \{t_{\sigma 0} : \sigma \in \Sigma\})$, where $t_{\sigma 0} := \max(T_\sigma)$. Now, suppose $q = (a, \{t_\sigma : \sigma \in \Sigma\})$. We have that $\delta(q, \sigma)!$ if and only if

- $\sigma = tick$ and $(\forall \tau \in \Sigma_{spe})\ \xi(a, \tau)!$ implies $t_\tau > 0$; or
- $\sigma \in \Sigma_{spe}$, $\xi(a, \sigma)!$, and $0 \le t_\sigma \le u_\sigma - l_\sigma$; or
- $\sigma \in \Sigma_{rem}$, $\xi(a, \sigma)!$, and $t_\sigma = 0$

When one of the above conditions is satisfied, we have that $q' = \delta(q, \sigma) := (a', \{t'_\sigma : \sigma \in \Sigma\})$ where either

- $\sigma = tick$ implies $a' := a$ and $t'_\tau := \begin{cases} \max(0, t_\tau - 1), & \xi(a, \tau)! \\ t_{\tau 0}, & \text{otherwise} \end{cases}$
- $\sigma \in \Sigma$ implies $a' := \xi(a, \sigma)$ and $t'_\tau := \begin{cases} t_\tau, & \xi(a', \tau)! \text{ and } \tau \ne \sigma \\ t_{\tau 0}, & \tau = \sigma \text{ or } \neg(\xi(a', \tau)!) \end{cases}$

Now, given an event set, or alphabet, Σ_t we define the set Σ_t^* to be the set of all finite strings of elements in Σ_t, including the empty string ε. It is straightforward to extend the definition for the transition function δ so that $\delta : Q \times \Sigma_t^* \to Q$. Specifically, given $q \in Q$, $\sigma \in \Sigma_t$, and $s \in \Sigma_t^*$, we set $\delta(q, s\sigma) := \delta(\delta(q, s), \sigma)$. Of course, the definition of ξ can similarly be extended. A language over Σ_t is simply a subset of Σ_t^*. The language consisting of strings generated by a TDES $\mathbf{G}$ is called the *closed behaviour* of $\mathbf{G}$ and is denoted by $L(\mathbf{G})$. Using the above definitions, we can write $L(\mathbf{G}) = \{s \in \Sigma_t^* : \delta(q_o, s)!\}$. We also define the projection operator $P_t : \Sigma_t^* \to \Sigma^*$ inductively via

$$P_t(\varepsilon) := \varepsilon$$
$$P_t(\sigma) := \begin{cases} \sigma, & \sigma \in \Sigma \\ \varepsilon, & \sigma = tick \end{cases}$$
$$P_t(s\sigma) := P_t(s) P_t(\sigma).$$

Thus, P_t simply erases all occurrences of the *tick* event in any string $s \in \Sigma_t^*$.

Since a TDES $\mathbf{G}$ has an automaton structure, it can be uniquely represented using a labeled digraph. Such a representation will be referred to as a timed transition graph (TTG) in the sequel. Similarly, we call the graphical representation of $\mathbf{G_{act}}$ the activity transition graph (ATG) of $\mathbf{G}$. A third useful representation of $\mathbf{G}$ is called the timed activity transition graph (TATG) and is the graphical representation of the untimed language $P_t(L(\mathbf{G}))$. Throughout the paper, we assume that all automata and graphical representations are deterministic in their transition structures and minimal state. This ensures that there is a one-to-one correspondence between the automaton, language, and graphical characterizations of $\mathbf{G}$. We denote by $\mathbf{G_{tact}}$ the automaton with the property that $L(\mathbf{G_{tact}}) = P_t(L(\mathbf{G}))$. These definitions in hand, we are now ready to introduce the notion of scaling timed discrete-event systems.

3 Invariance of TATGs Under Scaling of TDES

The goal of this section is to introduce a three parameter family of scaling functions on TDES. The scaling functions are affine transformations on the time bounds of the events in Σ. We choose to work with affine rather than linear scaling functions mainly for flexibility. While linear scaling functions allow for expansion and contraction of time intervals, affine scaling functions provide additional capacity for resizing intervals and also permit translations along the real-time axis. Moreover, when applying inverse scaling to TDES for the purposes of aggregation, affine scaling functions allow for rounding of time bounds and hence approximation methods. This in turn makes scaling applicable to a wider class of systems. We begin by showing that only under linear scaling of time bounds can we guarantee *a priori* the invariance of the TATG of an arbitrary TDES $\mathbf{G}$ under scaling.

3.1 Scaling Timed Discrete-Event Systems

Consider the TDES $\mathbf{G} = (Q, \Sigma_t, \delta, q_o)$ with $\Sigma_t = \Sigma \cup \{tick\}$ and whose timed events belong to the set $\Sigma_{tim} = \{(\sigma, l_\sigma, u_\sigma) : \sigma \in \Sigma\}$.

Definition 1 *Suppose a, b_1, and b_2 are integers. Then $\mathbf{G}' =: S_{a,b_1,b_2}(\mathbf{G})$ is an affine scaling of $\mathbf{G}$ if $\mathbf{G}'_{\mathbf{act}} = \mathbf{G}_{\mathbf{act}}$ and $\Sigma'_{tim} = \{(\sigma, l'_\sigma, u'_\sigma) : \sigma \in \Sigma\}$ where*

$$l'_\sigma := al_\sigma + b_1$$

$$u'_\sigma := \begin{cases} au_\sigma + b_2, & \sigma \in \Sigma_{spe} \\ \infty, & \sigma \in \Sigma_{rem} \end{cases}$$

Moreover, the affine scaling is well-defined if the following conditions are satisfied:

1) $a > 0$

2) $0 \leq l'_\sigma \leq u'_\sigma$, *for every* $\sigma \in \Sigma$

3) *for every* $q \in Q$ *and every string* $s \in \Sigma^* - \{\varepsilon\}$, $\delta(q, s) \neq q$

The first two conditions say that an affine scaling is well-defined if the scaling factor $a > 0$ and (l'_σ, u'_σ) forms an admissible set of time bounds for each event $\sigma \in \Sigma$, given the choice of offsets b_1 and b_2. The third condition is included for technical reasons. It prevents "non-Zeno" behaviour in which an infinite number of transitions are executed in a finite time interval. This condition can only be violated if $\mathbf{G}_{\mathbf{act}}$ contains a loop whose events *all* have lower time bounds that are mapped to 0 in $\mathbf{G}'$. For future reference, we denote the set of all TDES defined over the alphabet Σ by $\mathbb{G}(\Sigma)$, from which we define the subsets $\mathbb{S}_{a,b_1,b_2}$ by

$$\mathbb{S}_{a,b_1,b_2} := \{\mathbf{G} \in \mathbb{G}(\Sigma) : S_{a,b_1,b_2}(\mathbf{G}) \text{ is well defined}\}.$$

Note that in this notation, we have $\mathbb{S}_{a,0,0} = \mathbb{G}(\Sigma)$ for any integer $a > 0$.

Theorem 2 below constitutes the main result of the paper. It states that the only well-defined affine scalings which preserve the TATG of an arbitrary TDES $\mathbf{G}$ are those which are *linear* – namely, those for which the offsets b_1 and b_2 are identically zero. Moreover, the result states that the TATG will remain the same, regardless of the scaling factor used. Thus, from an event-sequence perspective, refinement of the original model provides no additional insight. This consequence is somewhat reminiscent of the famous Shannon Sampling Theorem for continuous time systems.

Theorem 2 *Given integers $a > 0$, b_1 and b_2, let $\mathbf{G}' := S_{a,b_1,b_2}(\mathbf{G})$ for each $\mathbf{G} \in \mathbb{S}_{a,b_1,b_2}$. Then $L(\mathbf{G}'_{\mathbf{tact}}) = L(\mathbf{G}_{\mathbf{tact}})$ for every TDES $\mathbf{G} \in \mathbb{S}_{a,b_1,b_2}$ if and only if $b_1 = b_2 = 0$.*

The proof of the result is much too long to present here. We merely sketch the main ideas employed and refer the interested reader to [5] for details. Necessity of the condition $b_1 = b_2 = 0$ is easily shown using examples. However, proving the sufficiency of this condition is significantly more challenging. Using the definition of an arbitrary TDES $\mathbf{G}$, it can be shown that if $s = s_0\, tick\, s_1\, tick \ldots \in L(\mathbf{G})$ then $s' = s_0\, tick^n\, s_1\, tick^n \ldots \in L(\mathbf{G}')$, where $\mathbf{G}' := S_n(\mathbf{G})$ and $S_n := S_{n,0,0}$. Since $P_t(s) = P_t(s')$, we have that $L(\mathbf{G}_{\mathbf{tact}}) \subseteq L(\mathbf{G}'_{\mathbf{tact}})$. The reverse subset inclusion is similarly proved using the definition of TDES although it is more difficult to deal with *tick* transitions in this direction.

While an analogous result to that of Theorem 2 is obvious for dense time systems such as the timed automata of Alur and Dill [2], it is somewhat counterintuitive in the TDES framework due to the discretization of time. Consider as an example the TDES $\mathbf{G}$ and $\mathbf{G}'$ whose ATG $\mathbf{G}_{\mathbf{act}}$ is shown in Figure 1.

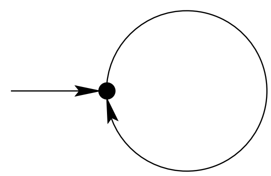

Figure 1: ATG $\mathbf{G}_{\mathbf{act}}$ for the example

Assume for $\mathbf{G}$, we have $\Sigma_{tim} := \{(\alpha, 1, 1)\}$ while $\Sigma'_{tim} := \{(\alpha, 2, 2)\}$. Suppose the *tick* frequency is one *tick* per second for $\mathbf{G}$ while $\mathbf{G}'$ operates with a *tick* rate of two *ticks* per second. The first occurrence of α must take place between the first and second *ticks* in $\mathbf{G}$. Namely, the first occurrence of α must take place in the real-time interval $(1, 2)$. Similarly, we find that the second occurrence of α takes place in the interval $(2, 3)$. Hence, the time between successive α transitions can be arbitrarily small in $\mathbf{G}$. Compare this to the TDES $\mathbf{G}'$ in which the first occurrence of α takes place between the second and third *tick*. Thus, this first occurrence of *tick* takes place in the real-time interval $(1, 1.5)$. It is easy to see that the second occurrence of α must lie in the real time interval $(2, 2.5)$ in $\mathbf{G}'$. So, we find that successive occurrences of α are subject to a relaxation time of at least 0.5 second in $\mathbf{G}'$. This can be seen in Figure 2 below. Note that despite this difference Theorem 2 says that $\mathbf{G}$ and $\mathbf{G}'$ share the same TATG. This phenomenon can lead to difficulties when composing subsystems operating with different granularities of time [5].

Before extending the result in different directions and discussing applications, we present an equivalent characterization of the sufficient condition appearing in Theorem 2. To this end, we make two final definitions before introducing Lemma 3. First, we define the function $tickcount : \Sigma_t^* \to \mathbb{N}_o$ which simply counts the number of occurrences of *tick* in a given string $s \in \Sigma_t^*$. Formally, for any $\sigma \in \Sigma_t$ and $s \in \Sigma_t^*$, we have

$$tickcount(\varepsilon) = 0$$

$$tickcount(\sigma) = \begin{cases} 0, & \sigma \in \Sigma \\ 1, & \sigma = tick \end{cases}$$

$$tickcount(s\sigma) = tickcount(s) + tickcount(\sigma).$$

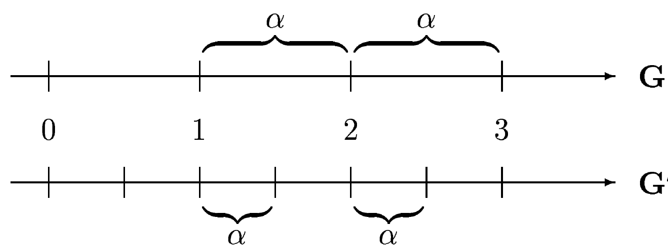

Figure 2: Real time axis

Next, for each $n \geq 2$, we define a new projection operator $P_n : \Sigma_t^* \to \Sigma_t^*$ inductively as follows:

$$P_n(\varepsilon) = \varepsilon$$

$$P_n(\sigma) = \begin{cases} \sigma, & \sigma \neq tick \\ \varepsilon, & \sigma = tick \end{cases}$$

$$P_n(s\sigma) = \begin{cases} P_n(s), & \sigma = tick \text{ and } tickcount(s\sigma) \not\equiv 0 \pmod n \\ P_n(s)\sigma, & \sigma \neq tick, \text{ or } \sigma = tick \text{ and } tickcount(s\sigma) \equiv 0 \pmod n), \end{cases}$$

where in the above $\sigma \in \Sigma_t$ and $s \in \Sigma_t^*$. Thus, given a string $s \in \Sigma_t^*$, the mapping P_n will keep every n^{th} occurrence of *tick* while deleting all other instances. We are now in a position to state our next result.

Lemma 3 *For every $n \in \mathbb{N}$, $L(\mathbf{G}) = P_n(L(S_n(\mathbf{G})))$.*

Notice that the sufficiency of the condition $b_1 = b_2 = 0$ in Theorem 2 immediately follows by application of P_t to both sides of the equality in Lemma 3. That $P_t(L(\mathbf{G})) = P_t(L(S_n(\mathbf{G})))$ implies $L(\mathbf{G}) = P_n(L(S_n(\mathbf{G})))$ can easily be derived using the results found in [5].

3.2 Inverse Scaling and State Space Reduction

The scaling result presented in the last section is useful in integrating faster components into a real-time system. However, inverse scaling, in which the *tick* rate experiences an n-fold *decrease*, has more applicability from a practical point of view. Below, we will show that a TDES can be inversely scaled provided the (finite) time bounds for each activity are divisible by a common integer n. In such a case, the scaled system $\mathbf{G}'$ satisfies the property $L(\mathbf{G}'_{\mathbf{tact}}) = L(\mathbf{G}_{\mathbf{tact}})$. We now make this discussion more precise.

Once again, we suppose the activity set for the TDES $\mathbf{G}$ is Σ and that the corresponding timed events are $\Sigma_{tim} = \{(\sigma, l_\sigma, u_\sigma) : \sigma \in \Sigma\}$. Now consider the mapping S^{-1}_{a,b_1,b_2}. If $\mathbf{G}' := S^{-1}_{a,b_1,b_2}(\mathbf{G})$, then it is clear that $\mathbf{G}'_{\mathbf{act}} = \mathbf{G}_{\mathbf{act}}$ and $\Sigma'_{tim} = \{(\sigma, l'_\sigma, u'_\sigma\} : \sigma \in \Sigma\}$, where

$$l'_\sigma := \frac{1}{a}(l_\sigma - b_1)$$

$$u'_\sigma := \begin{cases} \frac{1}{a}(u_\sigma - b_2), & \sigma \in \Sigma_{spe} \\ \infty, & \sigma \in \Sigma_{rem} \end{cases}$$

Hence, the mapping S^{-1}_{a,b_1,b_2} also affinely scales the time bounds of the activities in $\mathbf{G}$. In this case, we will say that $\mathbf{G}'$ is well-defined if $0 \le l'_\sigma \le u'_\sigma$, $l'_\sigma \in \mathbb{N}_o$ for every $\sigma \in \Sigma$, and $u'_\sigma \in \mathbb{N}_o$ for each $\sigma \in \Sigma_{spe}$. By analogy with our earlier work, we define the set $\mathbb{S}^{-1}_{a,b_1,b_2}$ to consist of all TDES $\mathbf{G} \in \mathbb{G}(\Sigma)$ such that $S^{-1}_{a,b_1,b_2}(\mathbf{G})$ is well-defined. Using these definitions, we have the following corollary of Theorem 2.

Corollary 4 *Suppose $a > 0$, b_1 and b_2 are integers. Then $L(\mathbf{G}'_{\mathbf{tact}}) = L(\mathbf{G}_{\mathbf{tact}})$ for every TDES $\mathbf{G} \in \mathbb{S}^{-1}_{a,b_1,b_2}$ if and only if $b_1 = b_2 = 0$, where for each $\mathbf{G} \in \mathbb{S}^{-1}_{a,b_1,b_2}$, $\mathbf{G}' := S^{-1}_{a,b_1,b_2}(\mathbf{G})$.*

Proof: It is obvious that $\mathbf{G} \in \mathbb{S}^{-1}_{a,b_1,b_2}$ if and only if $\mathbf{G}' \in \mathbb{S}_{a,b_1,b_2}$. The result follows by a direct application of Theorem 2. ■

The result is not unexpected. It says that if the time bounds are all integer multiples of a common integer $n > 1$, then our system is to some extent overspecified. Although the original system contains more accurate timing information which may be used elsewhere, the scaled system is identical with respect to its possible event sequences.

That this result is also quite useful from a practical point of view should now be obvious. Recall that typically the TATG for a given timed discrete-event system is obtained by first constructing its TTG and then projecting out all occurrences of *tick*. Now, the state size of a TTG grows quite rapidly with an increase in *tick* frequency. (It is easy to construct an example similar to that shown in Figure 1 for which the number of states and transitions both grow as $\mathcal{O}(n^r)$, where n is the *tick* density and r is the number of self-looped events.) Using Corollary 4, we can dramatically reduce the state size of the TTG for certain TDES and hence the computational effort required in obtaining its TATG.

Composition is also a source of combinatorial explosion of TDES. We now show that scaling for the purposes of state space reduction can be applied to composed systems, before or after composition, without changing the resulting system provided all subsystems can be scaled by a common factor $n \in \mathbb{N}$. The result can also be combined with the compositional model reduction techniques described in [12, 11].

Lemma 5 *Suppose $\mathbf{G_1}, \mathbf{G_2} \in \mathbb{S}^{-1}_n$, for some integer $n \in \mathbb{N}$. Then $\mathbf{G_1} \parallel \mathbf{G_2} \in \mathbb{S}^{-1}_n$ and*

$$S^{-1}_n(\mathbf{G_1}) \parallel S^{-1}_n(\mathbf{G_2}) = S^{-1}_n(\mathbf{G_1} \parallel \mathbf{G_2}).$$

It should be clear that inverse scaling can only be applied in special cases. In [5], we investigate other variants of the scaling functions introduced here. Specifically, we are able to derive language containment results by reintroducing nonzero offsets b_1 and b_2. This naturally leads to approximate scaling methods. For example, in [8], the authors use the scaling function $\llcorner S^{-1}_n \urcorner$ modeled after S^{-1}_n but with $l'_\sigma := \left\lfloor \frac{1}{n} l_\sigma \right\rfloor$, and $u'_\sigma := \left\lceil \frac{1}{n} u_\sigma \right\rceil$ (i.e., the floor and ceiling respectively) whenever $u_\sigma < \infty$. The function $\llcorner S^{-1}_n \urcorner$ is used in order to synthesize suboptimal supervisory controllers for TDES.

4 Dense Time Limit

We now begin our study of the limiting behaviour of affine scalings as *ticks* become dense in the real time axis. In this section, we merely sketch the details of our findings. The interested reader is referred to [5] for additional details.

Suppose we are given a TDES $\mathbf{G}$ and that for each $n \in \mathbb{N}$, $\mathbf{G_n} := S_n(\mathbf{G})$. Our specific goal is to investigate whether the limit

$$\mathbf{G}' := \lim_{n\to\infty} \mathbf{G_n},$$

is meaningful. If the system $\mathbf{G}'$ can in fact be defined, then we would also like to study some of its properties. It is clear that $\mathbf{G}'$ need not be in $\mathbb{G}(\Sigma)$, solely by considering the time bounds for its timed events. Notice that if $l_\sigma > 0$ then $l'_\sigma = \infty$ in the limit, which violates the definition of TDES.

Instantaneity and Simultaneity of Events

For the purposes of the discussion below, we assume that the system $\mathbf{G}'$ as defined above exists, ignoring the above mentioned problem regarding the new definition of time bounds. We now give further evidence that such a system is not a TDES over Σ. A second problem, more subtle than the one discussed in the above paragraph, exists with $\mathbf{G}'$. Consider once again the TDES $\mathbf{G}$ whose ATG is that of Figure 1. If the activity α has time bounds $(1, 1)$, then the first occurrence of α must lie between the first and the second *tick* of the global clock. If *ticks* occur at a rate of one per unit of time in $\mathbf{G}$, then we see that the first α occurs at some time $t \in (1, 2)$. Now scale $\mathbf{G}$ by a factor of n to obtain the TDES $\mathbf{G_n}$. Under this transformation, the timed event $(\alpha, 1, 1)$ maps to (α, n, n) and so α must first occur between the n^{th} and the $(n+1)^{st}$ occurrence of *tick* in $\mathbf{G_n}$. It is not difficult to see that given any $n \in \mathbb{N}$, if the *tick* frequency increases to n *ticks* per unit time in $\mathbf{G_n}$, then the first occurrence of α in $\mathbf{G_n}$ must be at some time $t \in (1, 1 + \frac{1}{n})$. In the limit as n becomes large, this interval is empty and hence $\mathbf{G}'$ is ill-defined.

The problem we encountered above results from the interleaving semantics of TDES in which no two events are allowed to occur simultaneously. Since $tick \in \Sigma_t$, it follows that no activity $\sigma \in \Sigma$ may occur at the precise moment in time that the digital clock is updated. One solution is to declare *tick* a "special" event whose occurrence can coincide with that of any other event $\sigma \in \Sigma$. The interpretation is that *tick* is simply an external signal which denotes the passage of one unit of time. By redefining TDES in this way, the occurrence of *tick* will always precede the occurrence of σ in a string $s \in L(\mathbf{G})$ in which *tick* and $\sigma \in \Sigma$ have occurred simultaneously in the real world. Two things should be clear at this point. First, by redefining TDES in this fashion, we have not enlarged the behaviour of any given TDES $\mathbf{G}$. Second, we note that splitting hairs in this manner is rooted in the mathematical description of the model rather than physical reality since measuring time with infinite precision is impossible to accomplish. Nonetheless, we make this definition and examine its consequences. To this end, consider the example of the previous paragraph once again. In this new framework, the first occurrence of the event α in $\mathbf{G_n}$ lies in the interval $[1, 1+\frac{1}{n})$. In the limit, the interval collapses to the single point $\{1\}$ and is thus well-defined. However, note that in $\mathbf{G}'$, α has become an instantaneous event, a feature which is reserved only for *tick* in the original TDES definition.

It should also be clear that two instantaneous (non-*tick*) events can be made to occur simultaneously with this redefinition and that $\mathbf{G}'$, regardless of how it is defined, is not a TDES in the conventional sense. This motivates the introduction of a new representation for TDES which is invariant under scaling. It is believed that such a new representation will provide further insight into how the limiting system $\mathbf{G}'$ should be defined and help elucidate more of its interesting features.

TDES and Discrete Systems

Before concluding this section, we will examine one last property of TDES that can be gleaned by applying the limit as *ticks* become dense in the real line. The simultaneity encountered above is a salient feature of synchronous (viz. clock driven) systems which use interleaving semantics [4, 3, 7, 9]. We will now show that given a TDES $\mathbf{G}$, we can define a corresponding synchronous system $\mathbf{H}$ for which $L(\mathbf{G_{tact}}) = P_t(L(\mathbf{H}))$, providing some justification for the remark in [1] that "discrete-time model[s] can be viewed as a special case [of fictitious clock models] where the events happen only in lock-step with the *ticks*".

Recall the discussion following the statement of Theorem 2. It should be clear that strings of the form $s_0\, tick^n\, s_1\, tick^n s_2 \ldots (tick)^n s_k$ are sufficient to generate the TATG of $\mathbf{G_n}$. We also now know that all events in the string s_0 must occur in the time interval $[0, \frac{1}{n})$, all the events in s_1 occur in $[1, 1+\frac{1}{n})$, and so on. In the limit as n gets large, we see that all events in the string s_j must occur exactly at time $t = j$. It is now clear how one would define a synchronous transition system whose untimed language is precisely that of our limiting system. The end result is similar to the digitization of timed state sequences described in [10]. It should also be clear that creating a TDES whose untimed language is precisely that of an arbitrary synchronous system is just as easily accomplished.

5 Conclusions and Future Work

The paper introduces a three parameter family of scaling functions on timed discrete-event systems. It is shown that linearity of the scaling function is an *a priori* necessary and sufficient condition to ensure invariance of the TATG. We described inverse scaling and its role in model reduction and showed that the operation of scaling commutes with composition.

Current work in the area is focused on extending the methods of this paper to scale real-time temporal logics involving *tick* transitions. It is hoped that the cost of model-checking some TDES can be significantly reduced in this fashion. Further research will be directed toward the study of multirate TDES in which different subsystems have different *tick* frequencies and toward the introduction of a scaling invariant representation of TDES.

References

[1] R. Alur, C. Courcoubetis, and D. Dill. Model-checking in dense real-time. *Information and Computation*, **104**(1), pages 2–34, 1993.

[2] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, **126**, pages 183–235, 1994.

[3] A. Benveniste, P. Le Guernic, and C. Jacquemot. Synchronous programming with events and relations: The SIGNAL language and its semantics. *Science of Computer Programming*, **16**(2), pages 103–149, 1991.

[4] G. Berry and L. Cosserat. The ESTEREL synchronous programming language and its mathematical semantics. In *CMU Seminar on Concurrency*, LNCS 197, pages 389–448, 1985.

[5] S.E. Bourdon. Design and verification of real-time control systems. PhD thesis, Systems Control Group, Department of Electrical and Computer Engineering, University of Toronto, *In Preparation.*

[6] B.A. Brandin and W.M. Wonham. The supervisory control of timed DES. *IEEE Transactions on Automatic Control*, **39**(2), pages 329–342, 1994.

[7] P. Caspi, D. Pilaud, N. Halbwachs, and J. Plaice. LUSTRE: A declarative language for programming synchronous systems. In *Proceedings of the 14th Annual Symposium on Principles of Programming Languages*, pages 178–188, 1987.

[8] P. Gohari and W.M. Wonham. Reduced supervisors for timed discrete-event systems. In R. Boel and G. Stremersch, editors, *Discrete Event Systems: Analysis and Control*, SECS 569, pages 119–130, 2000.

[9] D. Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, **8**, pages 231–274, 1987.

[10] T.A. Henzinger, Z. Manna, and A. Pnueli. What good are digital clocks? In *Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP 1992)*, LNCS 623, pages 545–558, 1992.

[11] M. Lawford. *Model Reduction of Discrete Real-Time Systems.* PhD thesis, Systems Control Group, Department of Electrical and Computer Engineering, University of Toronto, 1997. Also available as Systems Control Group Report No. 9702.

[12] M. Lawford, J.S. Ostroff, and W.M. Wonham. Model reduction of modules for state-event temporal logics. In *Formal Description Techniques IX: Theory, Application, and Tools*, pages 263–278, 1996.

[13] P. Ramadge and W.M. Wonham. Supervisory control of a class of discrete-event processes. *SIAM Journal on Control and Optimization*, **25**(1), pages 206–230, 1987.