Supervisory Control of Real-time Discrete Event Systems using Lattice Theory

Darren D. Cofer, Vijay K. Garg

Abstract—**The behavior of timed DES can be described by sequences of event occurrence times. These sequences can be ordered to form a lattice. Since logical (untimed) DES behaviors described by regular languages also form a lattice, questions of controllability for timed DES may be treated in much the same manner as they are for untimed systems. In this paper we establish conditions for the controllability of timed DES performance specifications which are expressed as inequations on the lattice of sequences. These specifications may take the form of sets of acceptable event occurrence times, maximum or minimum occurrence times, or limits on the separation times between events. Optimal behaviors are found as extremal solutions to these inequations using fixed point results for lattices.**

Keywords: Discrete event systems, supervisory control, max-algebra, lattices.

I. Introduction

Discrete event systems (DES) are characterized by a collection of *events*, such as the completion of a job in a manufacturing process or the arrival of a message in a communication network. The system state changes only at time instants corresponding to the occurrence of one of the defined events. At the logical level of abstraction, the behavior of a DES is described by the sequences of events that it performs. The actual or desired behavior of a logical DES is specified as a set of event sequences known as a *language*. Languages may be ordered by set inclusion to form a *lattice*.

However, if time constraints are of explicit concern in the system dynamics and its performance specification, its behavior can be characterized by sequences of occurrence times for each event. These time sequences may be ordered to form a lattice. Thus, there is an underlying algebraic similarity between logical and timed models for DES that we may be able to exploit. In this paper we show how lattice techniques developed for the studying the control of logical DES behaviors can also be applied to timed DES.

Some aspects of a DES, such as processing times or machine failures, will normally be fixed and beyond control. Others, such as starting a manufacturing process or broadcasting a message, may be controllable by a human operator or an automated controller. The problem in controller design is to determine when these controllable events should or should not occur to achieve some performance goal or to ensure that certain properties always hold for the system.

A well-developed theoretical framework has been established by Ramadge and Wonham for studying the control of logical DES behavior [17]. In this framework event sequences are normally assumed to be generated by a state machine or some other automaton. Certain events are designated as being controllable and may be disabled by a *supervisor* to restrict the system to some specified behavior, given as a set of desirable event sequences. For a specification to be controllable it must satisfy a particular invariance property which is expressed as a set function inequality, or *inequation*, on the lattice of languages. An optimal supervisor is computed as an extremal solution of the lattice inequation which characterizes the invariance property [13].

One approach to supervisory control of timed DES is presented in [3]. In this model the timing features of timed transition models used in [16] are added to the control structure of [17]. Controllable events may be forced as well as disabled by a supervisor. The untimed state machine which models the system is extended by adding a clock tick event and augmenting the state space to include a timer for each activity in the system. The status of the timers may then influence state transitions. However, the addition of a global clock greatly increases the number of transitions in the system. While DES are normally characterized by state updates which are event-driven and occur at irregular intervals, this model updates the state at every clock tick as in a discrete–time dynamical system.

Another approach to controlling timed DES is given in [11]. In that work a timed DES is modelled as a generalized semi–Markov process (GSMP) which consists of a state space, a set of events, a list of active events in each state, and a state transition function defining the effect of each event on the system state. In addition, a (usually stochastic) time duration is specified for the time from the activation of events until their occurrence. Input parameters control the event durations. The authors establish structural conditions for GSMPs which cause the system performance to be a monotone function of the inputs.

Our work here extends the supervisory control ideas of [17] to timed DES but by using an entirely different system model from [3]. Suppose that certain events are controllable by means of delaying their execution, rather than simply prohibiting them. Desirable behavior is specified by a range of acceptable execution times for events. By using a *max-algebra* model for timed DES [2], [8], [9] we demonstrate how this control problem may be viewed from the Ramadge–Wonham perspective.

The max-algebra approach allows the dynamics of cer-

This work was supported in part by NSF Grant CCR-9110605, Army Grant N00039-91-C-0082, a TRW faculty assistantship award, General Motors Fellowship, and an IBM grant.

D. Cofer is with Honeywell, Inc., Honeywell Technology Center, Minneapolis, MN 55418 USA

V. Garg is with the Deptartment of Electrical and Computer Engineering, University of Texas at Austin, TX 78712-1084 USA.

tain timed DES to be modelled by a system of linear equations. This is accomplished using a non–traditional algebraic structure known as a *dioid* or *semiring* in which the conventional ring operations of addition and multiplication in $\mathcal{R}$ are typically replaced by maximization and addition, respectively. Work in this area was initiated by Cuninghame–Green in [10] from the perspective of operations research. In the early 1980's this work was discovered by Cohen, Dubois, Quadrat, and Viot and utilized in their study of discrete event manufacturing processes [8]. This eventually led to the formation of the "Max Plus" working group of researchers. Their study of DES using related algebraic structures continues to the present and is well–documented in [2].

Controllable behavior for timed DES in the max-algebra framework may be defined by an invariance condition which is quantified by a lattice inequation. A set of desired behaviors (event schedules) must satisfy this inequation in order for it be realizable by any supervisor which is restricted to delaying only the controllable events. This formulation allows us to find optimal supervisors by computing extremal solutions to the inequation using fixed point results for lattices.

Previous work on max-algebra models of timed DES has focused on performance analysis rather than control. The control problems which have been addressed in [2] deal with questions of stability (maintaining finite sojourn times in the system) or resource placement to achieve the earliest output times.

In [11] the control objective is minimization of a cost function associated with holding times for states. They show that systems similar to the ones we consider (based on $(\max, +)$ recursions) satisfy a monotonicity condition on the number or occurrences of each type of event in the system. This condition guarantees that optimal control policies for the system will be monotone functions of the number of events that have occurred.

In contrast with both [2] and [11], the control problem we address here has more in common with "forbidden state" problems in logical DES. Given a set of desirable behaviors we examine the existence and computation of optimal supervisors for a system. Depending on the performance objective, "optimal" may mean either minimally restricting the system so as to remain within the desired region or limiting the system to an operating region which guarantees that at least all desirable behaviors can be achieved.

In Section 2 we review the max-algebra model used in this paper and show the similarity between logical DES modelled by finite state machines and timed DES modelled by timed event graphs. Using this similarity, Section 3 defines supervisory control for timed DES. In Section 4 we present the lattice theory results for computing extremal solutions to inequations and apply these results to the control of timed DES.

II. Timed Event Graphs

DES which are subject to time synchronization constraints can be modelled by automata known as *timed event graphs.* A timed event graph is a Petri net in which a time delay is associated with each place and with forks and joins permitted only at its transitions (see figure 1). Each transition $t_i \in T$ in the graph corresponds to an event in the system. Tokens entering a place are made available after incurring the specified delay. Transitions fire as soon as tokens are available in all their predecessor places.

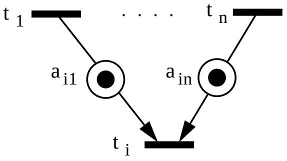

Fig. 1. Timed event graph.

In [2] and [9] these systems are modelled using an algebraic structure called a *dioid.* Also called an *idempotent semiring*, a dioid has two binary operations normally denoted $\oplus$ and $\otimes$ and differs most notably from a ring in that there is no inverse with respect to the sum ($\oplus$). To describe a timed event graph, $\oplus$ is usually defined to be either maximization or minimization and $\otimes$ is defined as addition.

The dynamic behavior of such systems can also be studied using an algebraic structure called a *semimodule* (or *moduloid*). While a module is a ring acting on a commutative group, a semimodule is essentially a dioid acting on a monoid. The principal difference between a module and a semimodule is that the existence of inverses with respect to the sum is replaced by the idempotency property. More specifically, a semimodule over a dioid D is a commutative idempotent monoid $\mathcal{X}$ together with an operation called *scalar multiplication* mapping $D \times \mathcal{X} \to \mathcal{X}$. Scalar multiplication distributes over the sum operation $\oplus$ in both D and $\mathcal{X}$, is associative with the product in D, and has as its identity the identity element of D. We let ε denote the null element for $\oplus$ in $\mathcal{X}$. D and $\mathcal{X}$ are *complete* if they are closed for arbitrary $\oplus$ operations.

A partial order is induced in both D and $\mathcal{X}$ by the relation

$$x \leq y \Leftrightarrow x \oplus y = y. \tag{1}$$

Scalar multiplication is isotone with respect to this order in $\mathcal{X}$. Properties of semimodules are discussed in [19] and [2].

Definition 1: A function $f : \mathcal{X} \to \mathcal{X}$ is called *lower-semicontinuous (l.s.c.)* [2] if for any collection of elements $X \subseteq \mathcal{X}$

$$f(\bigoplus_{x \in X} x) = \bigoplus_{x \in X} f(x).$$

Note that the identity function $(0(x) = x)$ and the null function $(\varepsilon(x) = \varepsilon)$ are both l.s.c. Also note that any l.s.c. function maps ε to itself (simply take the index set X in the definition to be empty).

Theorem 1: Let $(\mathcal{X}, \oplus)$ be an idempotent commutative monoid and let F be the set of l.s.c. functions on $\mathcal{X}$. Let $\oplus$ and $\otimes$ be binary operators on F defined by

$$(f \oplus g)(x) \quad = \quad f(x) \oplus g(x)$$

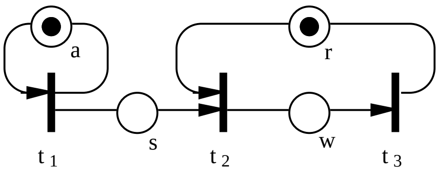

Fig. 2. Timed event graph for manufacturing process.

$$(f \otimes g)(x) \quad = \quad f(g(x)).$$

Then $(F, \oplus, \otimes)$ is a dioid. If $\mathcal{X}$ is complete, so is F. Furthermore, if scalar multiplication is defined to be the action of functions in F on $\mathcal{X}$, then $\mathcal{X}$ is a semimodule over F [6].

Proof summary: F is a dioid since the required properties of $\oplus$ in F are inherited from $(\mathcal{X}, \oplus)$, while the properties of $\otimes$ follow from its definition as functional composition. The $\otimes$ operator distributes over $\oplus$ because of lower–semicontinuity. Similarly, the required properties of scalar multiplication follow from the definitions of $\oplus$ and $\otimes$. □

Consider a timed event graph such as Figure 1. Using the semimodule structure the behavior of a timed event graph with n events is governed by the equation

$$x = Ax \oplus v \tag{2}$$

where x is the sequence of firing time vectors for events, v is a sequence of earliest allowable firing time vectors, and A is an $n \times n$ matrix of delay functions at places. The least solution of (2) is $x = A^* v$ where

$$\begin{aligned} A^* &= \bigoplus_{i \geq 0} A^i \\ A^0 &= I \equiv \begin{bmatrix} \varepsilon & & 0 \\ & \ddots & \\ 0 & & \varepsilon \end{bmatrix}. \end{aligned}$$

The set of vector sequences in $(\mathcal{R} \cup \{\pm\infty\})^n$ forms an idempotent commutative monoid under pointwise maximization which we denote by $\mathcal{S}^n$. Note that this monoid is complete. For systems with constant delay times, the delay functions are of the form $a\gamma^m$, where a represents unary addition of some constant and γ is the index backshift function $(\gamma x(k) = x(k-1))$. Functions of this form are easily shown to be l.s.c. This structure can also be used to model systems more general than timed event graphs with constant delays, including some untimed DES and time-varying systems [6].

Example 1: To illustrate this approach, consider the timed event graph of Figure 2 which represents a manufacturing process. Upon arrival, a part is set up (s) in a machine queue and then worked (w) in order of arrival. Each of the subprocesses s and w takes essentially constant time. However, the part interarrival times a may vary due to the workfloor schedule. The machine reset time r is normally constant except for periodic replacement of the cutting head. Letting x_i denote the occurrence times of event t_i, the process is described by

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a\gamma & \varepsilon & \varepsilon \\ s & \varepsilon & r\gamma \\ \varepsilon & w & \varepsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \oplus \begin{bmatrix} \bar{0} \\ \bar{0} \\ \bar{0} \end{bmatrix}.$$

The overbar notation indicates a constant sequence, so for this example no events may occur before time 0. If we are interested in the part completion times given by x_3 this is computed using $x = A^* v$. Since

$$A^* = \begin{bmatrix} (a\gamma)^* & \varepsilon & \varepsilon \\ (r\gamma w)^* s(a\gamma)^* & (r\gamma w)^* & (r\gamma w)^* r\gamma \\ (wr\gamma)^* ws(a\gamma)^* & (wr\gamma)^* w & (wr\gamma)^* \end{bmatrix}$$

and $v_1 = v_2 = v_3 = \bar{0}$ we have

$$\begin{aligned} x_3 &= (wr\gamma)^*(ws(a\gamma)^* \oplus w \oplus 0)(\bar{0}) \\ &= (wr\gamma)^* ws(a\gamma)^*(\bar{0}). \end{aligned} \tag{3}$$

Terms like $(a\gamma)^*$ are used to denote an expression of the form

$$0 \oplus a\gamma \oplus (a\gamma)^2 \oplus \ldots$$

Once the delay functions are specified, the part completion times are computed from (3). In particular, let $s(x) = x + 1$, $w(x) = x + 4$, and

$$\begin{aligned} a(x(k)) &= \begin{cases} x(k) + 5 & k \text{ odd} \\ x(x) + 7 & k \text{ even} \end{cases} \\ r(x(k)) &= \begin{cases} x(k) + 4 & k \bmod 5 = 0 \\ x(k) + 1 & \text{otherwise} \end{cases} \end{aligned}$$

For the $(a\gamma)^*$ term we find

$$\begin{aligned} (a\gamma)^*(\bar{0}) &= \bar{0} \oplus a\gamma\bar{0} \oplus (a\gamma)^2(\bar{0}) \oplus (a\gamma)^3(\bar{0}) \ldots \\ &= \{0, 0, 0, 0, \ldots\} \oplus \\ &\quad\ \{\varepsilon, 5, 7, 5, 7, \ldots\} \oplus \\ &\quad\ \{\varepsilon, \varepsilon, 12, 12, 12, \ldots\} \oplus \\ &\quad\ \{\varepsilon, \varepsilon, \varepsilon, 17, 19, \ldots\} \oplus \ldots \\ &= \{0, 5, 12, 17, 24, \ldots\} \end{aligned}$$

Completing the computation, we obtain

$$x_3 = \{5, 10, 17, 22, 29, 37, 42, 47, 58 \ldots\}.$$

□

Timed event graphs are structurally very similar to *finite state machines* which are often used to model logical DES. Both are directed graphs with labelled edges. While in a timed event graph the nodes correspond to events and the edges to process delays, the nodes in a finite state machine represent the system states and the edges correspond to events. This is illustrated by comparing Figures 2 and 3. The structures are dual in the sense that a timed event graph models synchronization and concurrency but not nondeterministic choice while a finite state machine models nondeterministic choice but not synchronization or concurrency. (We note that timed event graphs which include deterministic choices are discussed in [2] and [5].)

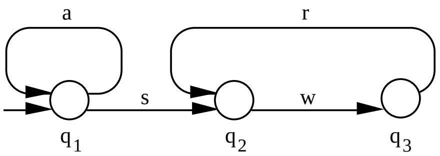

Fig. 3. Finite state machine with same structure as Figure 2

Because of the structural similarities between these automata there is an algebraic similarity as well. If we consider sets of event sequences with the operation $\oplus$ defined as set union, the sequences accepted by a finite state machine can be described in the semimodule framework used for timed event graphs. The functions which are assigned to the edges in this case are right concatenations of event labels.

The system of Figure 3 is therefore governed by the equation

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a & \varepsilon & \varepsilon \\ s & \varepsilon & r \\ \varepsilon & w & \varepsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ \emptyset \\ \emptyset \end{bmatrix}.$$

which is of the form $x = Ax \oplus v$ as before. Here ε is the constant function which maps to the empty set and 1 is the zero–length sequence. Its purpose here is to indicate the initial state of the system (q_1). The solution for state q_3 is

$$q_3 = a^* sw(rw)^*$$

which is a regular expression over the event set of the system. Furthermore, except for omission of the γ functions and the reversed order (due to the convention of composing functions from the left) it is the same as the solution (3) for the timed event graph of Figure 2.

It is this algebraic similarity which suggests that control of timed event graphs may be studied using techniques developed for untimed DES.

III. Supervisory Control of Timed DES

To motivate our approach to controlling timed DES, we briefly recall the framework for controlling logical or untimed DES behaviors. In a logical DES model with event set Σ, the *language* $L \subseteq \Sigma^*$ of the system is the set of all event sequences it can generate. Events are classified as either *controllable*, meaning that their occurrence may be prevented, or *uncontrollable*. Control is accomplished by dynamically disabling certain of the controllable events to avoid undesirable behaviors (event sequences not in the specified language). Observe that such a controller or *supervisor* may restrict system behavior, but not introduce any new behaviors.

Since uncontrollable events may not be disabled, not all behaviors are realizable. Language K is said to be *controllable* with respect to L and the uncontrollable events Σ_u if

$$pr(K).\Sigma_u \cap L \subseteq pr(K) \tag{4}$$

where $pr(K)$ is the *prefix closure* of K.

For a supervisor to be able to restrict the system to a desired language, uncontrollable actions must not result in sequences which lie outside of that language. Furthermore, we must permit the system to execute those behaviors in $pr(K)$ which lead to the acceptable region, even though they may not lie in that region themselves. Therefore, these potentially acceptable behaviors as well as the region of acceptable behavior should be invariant or closed under uncontrollable actions. Since uncontrollable events in the system act by concatenation this invariance property yields the definition of controllability stated in (4). If the desired behavior does not satisfy the controllability condition, a supervisor may be constructed to achieve the supremal sublanguage which is controllable.

Now return to the max-algebra model of a timed event graph governed by (2). Suppose that some events $T_c \subseteq T$ are designated as controllable, meaning that their transitions may be delayed from firing until some arbitrary later time. This is similar in its mechanics to the controlled Petri net concept introduced in [12] for untimed DES. The delayed enabling times $u_i(k)$ for the controllable events are to be provided by a supervisor. Let u represent the sequence of transition enabling times provided by the supervisor, with $u_i(k) = \varepsilon$ for t_i uncontrollable. Then the supervised system is described by $x = Ax \oplus v \oplus u$.

To compute the effect of uncontrollable events, let I_c denote the matrix having the identity function on diagonal elements i for which $t_i \in T_c$ and ε elsewhere. Then for any desired sequence $y \in Y$ the supervisor provides firing times $u = I_c y$ which results in $x = A^*(I_c y \oplus v)$. Since the desired behavior must be invariant under uncontrollable actions, we have the following definition of controllability.

Definition 2: A set of sequences $Y \subseteq \mathcal{S}$ is *controllable* with respect to A, v, and T_c if

$$A^*(I_c Y \oplus v) \subseteq Y. \tag{5}$$

Intuitively, this means that enabling controllable events at any time allowed by the specification set Y must result in behavior within Y for all events. Notice that, as in the untimed model, no new behavior is introduced by the supervisor. System operation can never be accelerated — events can only be delayed.

As an example suppose then that we wish to slow the system down as much as possible without causing any event to occur later than some sequence of execution times y. Such a specification could be used to prevent buffer overflows, ensure the availability of sufficient processing time to accomplish a task, or to synchronize events in independent systems. This type of specification is described by

$$Y = \{x \in \mathcal{S} | x \leq y\} \tag{6}$$

where $y \in \mathcal{S}$ is a fixed sequence. For the remainder of this section we will consider acceptable behaviors of this form.

Theorem 2: If Y specifies sequences no later than y as in (6) then Y is controllable if and only if

$$A^*(I_c y \oplus v) \leq y. \tag{7}$$

Proof: If Y is controllable then (5) holds. Thus

$$\begin{aligned} y\in Y &\Rightarrow A^*(I_c y\oplus v)\in Y\\ &\Rightarrow A^*(I_c y\oplus v)\le y. \end{aligned}$$

Conversely, suppose $A^*(I_c y\oplus v)\le y$. Then $\forall x\in Y$

$$\begin{aligned} x\in Y &\Rightarrow x\le y\\ &\Rightarrow A^*(I_c x\oplus v)\le A^*(I_c y\oplus v)\\ &\Rightarrow A^*(I_c x\oplus v)\le y\\ &\Rightarrow A^*(I_c x\oplus v)\in Y. \end{aligned}$$

Therefore $A^*(I_c Y\oplus v)\subseteq Y$. □

Example 2: For the system in Figure 2 with t_2 controllable, suppose we are given as a specification the set Y of sequences less than or equal to

$$y=\begin{bmatrix}(7\gamma)^*(\bar{0})\\ s(7\gamma)^*(\bar{0})\\ ws(7\gamma)^*(\bar{0})\end{bmatrix}=\left\{\begin{bmatrix}0\\1\\5\end{bmatrix},\begin{bmatrix}7\\8\\12\end{bmatrix},\begin{bmatrix}14\\15\\19\end{bmatrix},\ldots\right\}.$$

This specification seeks to remove variations due to the interarrival delays a. To determine if Y is controllable, we compute

$$\begin{aligned} A^*I_c y\oplus A^*v &=\\ \begin{bmatrix}\varepsilon\\ (r\gamma w)^*s(7\gamma)^*(\bar{0})\\ (wr\gamma)^*ws(7\gamma)^*(\bar{0})\end{bmatrix} &\oplus \begin{bmatrix}(a\gamma)^*(\bar{0})\\ (r\gamma w)^*s(a\gamma)^*(\bar{0})\\ (wr\gamma)^*ws(a\gamma)^*(\bar{0})\end{bmatrix}\\ &= \begin{bmatrix}(a\gamma)^*(\bar{0})\\ (r\gamma w)^*s(7\gamma)^*(\bar{0})\\ (wr\gamma)^*ws(7\gamma)^*(\bar{0})\end{bmatrix}, \end{aligned}$$

making use of the fact that $(7\gamma)^*\ge(a\gamma)^*$. Examining the third component we find that

$$\begin{aligned} (wr\gamma)^*ws(7\gamma)^*(\bar{0}) &= \{5,12,19,26,33,41,47,\ldots\}\\ > y_3 &= \{5,12,19,26,33,40,47,\ldots\} \end{aligned}$$

so Y is not controllable. □

When a specification is found to be uncontrollable, a less restrictive controllable specification can always be generated from it.

Corollary 1: If $y\ge v$ then A^*y is a controllable sequence.

Proof: By the construction of I_c we have $I_cA^*y\le A^*y$ and

$$\begin{aligned} &\Rightarrow A^*I_cA^*y\le A^*y\\ &\Rightarrow A^*I_cA^*y\oplus A^*v\le A^*y\oplus A^*v\\ &\Rightarrow A^*(I_c(A^*y)\oplus v)\le A^*y \end{aligned}$$

□

Example 3: Consider y from Example 2. The relaxed specification is

$$\begin{aligned} A^*y &= \begin{bmatrix}(a\gamma)^*(7\gamma)^*(\bar{0})\\ (r\gamma w)^*(s(a\gamma)^*\oplus s\oplus r\gamma ws)(7\gamma)^*(\bar{0})\\ (wr\gamma)^*(ws(a\gamma)^*\oplus ws)(7\gamma)^*(\bar{0})\end{bmatrix}\\ &= \begin{bmatrix}(7\gamma)^*(\bar{0})\\ (r\gamma w)^*s(7\gamma)^*(\bar{0})\\ (wr\gamma)^*ws(7\gamma)^*(\bar{0})\end{bmatrix} \end{aligned}$$

where we have used the fact that $(r\gamma w)^*(0\oplus r\gamma w)=(r\gamma w)^*$. Checking condition (7) we find

$$\begin{aligned} A^*I_c(A^*y)\oplus A^*v &=\\ \begin{bmatrix}\varepsilon\\ (r\gamma w)^*s(7\gamma)^*(\bar{0})\\ w(r\gamma w)^*s(7\gamma)^*(\bar{0})\end{bmatrix} &\oplus \begin{bmatrix}(a\gamma)^*(\bar{0})\\ (r\gamma w)^*s(a\gamma)^*(\bar{0})\\ (wr\gamma)^*ws(a\gamma)^*(\bar{0})\end{bmatrix}\\ &= \begin{bmatrix}(7\gamma)^*(\bar{0})\\ (r\gamma w)^*s(7\gamma)^*(\bar{0})\\ w(r\gamma w)^*s(7\gamma)^*(\bar{0})\end{bmatrix} \end{aligned}$$

so A^*y is controllable. □

It is more likely that a given specification cannot be relaxed. In this case we must find the least restrictive set of controllable behaviors which meets the specification. Depending on the situation this may mean finding the largest set which is contained in the specified behavior or the smallest set which contains the specified behavior. Determining these extremal optimal behaviors is the subject of the next section.

IV. Extremal Behaviors of Timed DES

The controllability of a set of behaviors for a timed DES is specified by an inequation (5) on the lattice of time sequences. In algebraic terms this is the same situation we find for specifying the controllability of logical DES behaviors. We demonstrate next that for timed DES where the desired behavior is not controllable the extremal controllable behaviors can be found using fixed point results for lattices.

A. Lattice Theory for Logical DES

Let $\mathcal{X}$ be an idempotent commutative monoid as considered in Section II. The operation $\oplus$ induces a partial order on $\mathcal{X}$ defined by (1). For any pair in $\mathcal{X}$ the least upper bound with respect to this order or $\sup\{x,y\}$ is given by $x\oplus y$. Since $\mathcal{X}$ is complete, we may define $\sup X$ for any set $X\subseteq\mathcal{X}$ (not just pairs) by

$$\sup X=\bigoplus_{x\in\mathcal{X}} x.$$

We can also induce a greatest lower bound $\wedge$ from $\oplus$ by taking

$$\inf X=\sup\{z\in\mathcal{X}\mid z\le x\ \forall x\in X\}.$$

Since $\varepsilon\le x$ for all $x\in\mathcal{X}$, $\inf X$ is well defined for arbitrary $X\subseteq\mathcal{X}$.

The partially ordered set $\mathcal{X}$ is a *complete lattice* if $\sup X$ and $\inf X$ are in $\mathcal{X}$ for any $X\subseteq\mathcal{X}$. Therefore a complete idempotent commutative monoid $\mathcal{X}$ is also a complete lattice. We will use the operators $\sqcup$ and $\sqcap$ to denote sup and inf respectively in a general lattice. Some complete lattices which will be of interest are:

- $(\mathcal{S},\le)$, the lattice of sequences with the order induced by $\oplus$ (pointwise maximization).
- $(2^{\mathcal{S}},\subseteq)$, the power set of sequences with the order of set containment. In this lattice $\sqcup$ and $\sqcap$ are $\cup$ and $\cap$, respectively.

- $(2^{\Sigma^*}, \subseteq)$, the set of all languages over the event set Σ with the order of set containment.

We next recall some useful properties of functions over lattices. A function $f : \mathcal{X} \to \mathcal{X}$ is *idempotent* if

$$\forall x \in \mathcal{X} : f(x) = f(f(x)).$$

It is *monotone* if

$$\forall x, y \in \mathcal{X} : x \leq y \Rightarrow f(x) \leq f(y).$$

For $\mathcal{X}$ a complete lattice, a function is *disjunctive* if

$$\forall X \subseteq \mathcal{X} : f(\sup X) = \sup_{x \in X} \{f(x)\}.$$

Note that for the lattice $(\mathcal{S}, \leq)$ this is equivalent to the l.s.c. property. A function is *conjunctive* if

$$\forall X \subseteq \mathcal{X} : f(\inf X) = \inf_{x \in X} \{f(x)\}.$$

Disjunctive and conjunctive functions can both be shown to be monotone as well.

Example 4: As an example of a disjunctive function consider any function $f : \mathcal{S} \to \mathcal{S}$ on the lattice of sequences. The function may be extended to the power set lattice $(2^{\mathcal{S}}, \subseteq)$ by taking

$$\hat{f}(X) = \bigcup_{x \in X} \{f(x)\}.$$

Clearly any function $\hat{f} : 2^{\mathcal{S}} \to 2^{\mathcal{S}}$ defined in this way is disjunctive. □

Definition 3: Consider a complete lattice $(\mathcal{X}, \leq)$ and a function $f : \mathcal{X} \to \mathcal{X}$. The *disjunctive closure* of f, denoted $f^{\sqcup}$, is defined by

$$f^{\sqcup}(x) = \bigsqcup_{i \geq 0} f^i(x).$$

It is easy to see that the disjunctive closure of a function is idempotent and disjunctive. For the lattice $(\mathcal{S}, \leq)$ the least upper bound operation is $\oplus$ so $f^{\sqcup}(x) = f^*(x)$. For the lattice $(2^{\mathcal{S}}, \subseteq)$ the least upper bound operation is $\cup$ so $f^{\sqcup}(x) = \cup_{i \geq 0} f^i(x)$.

Definition 4: Consider a complete lattice $(\mathcal{X}, \leq)$ and a function $f : \mathcal{X} \to \mathcal{X}$. If f is disjunctive then its *dual*, denoted $f^{\perp}$, is defined by

$$f^{\perp}(y) = \sup\{x \in \mathcal{X} | f(x) \leq y\}.$$

If f is conjunctive then its *co-dual*, denoted $f^{\top}$, is defined by

$$f^{\top}(y) = \inf\{x \in \mathcal{X} | y \leq f(x)\}.$$

The conditions of disjunctivity and conjunctivity are necessary in the definition to guarantee the existence of the dual and co-dual [13].

Remark: On the lattice of sequences $(\mathcal{S}, \leq)$ the dual is equivalent to the *residual* defined in [2]. The residual of a function $f : X \to Y$ is a function $f^{\sharp} : Y \to X$ which map points $y \in Y$ to the greatest subsolution of the equation $f(x) = y$.

Example 5: Consider the index backshift function $\gamma : \mathcal{S} \to \mathcal{S}$ defined by

$$\gamma(x(k)) = \begin{cases} x(k-1) & : k \geq 1 \\ \varepsilon & : k = 0 \end{cases} .$$

Let γ^{-1} be defined by $\gamma^{-1}(x(k)) = x(k+1)$. Since $\forall y \in \mathcal{S} : \gamma(\gamma^{-1}(y)) = y$ and, furthermore, $\gamma(x) = y \Leftrightarrow x = \gamma^{-1}(y)$ we must have

$$\gamma^{\perp}(y) = \sup\{x \in \mathcal{S} | \gamma(x) \leq y\} = \gamma^{-1}.$$

Now consider a time-varying delay as in Example 1 where $a(x(k)) = x(k) + a_k$. In this case $a^{\perp}x(k) = x(k) - a_k$. □

It can be shown that the dual of a function has the following properties.

1. If f_1 and f_2 are disjunctive functions then $(f_1 f_2)^{\perp} = f_2^{\perp} f_1^{\perp}$ [13].
2. If f is idempotent then $f^{\perp}$ is also idempotent (follows from 1.).
3. If f and g are disjunctive then $(f \sqcup g)^{\perp} = f^{\perp} \sqcap g^{\perp}$ [2].

As discussed the previous section the behavior of a DES is often specified by a system of *inequations* over the underlying lattice of the form

$$\{f_i(x) \leq g_i(x)\}_{i \leq n}. \qquad (8)$$

It is shown in [13] and [14] that extremal solutions of (8) can be reduced to extremal fixed–point computation of certain induced functions. The relevant results are summarized in the following two theorems.

Theorem 3: Given the system of inequations (8) over a complete lattice $(\mathcal{X}, \leq)$, let

$$Y = \{y \in \mathcal{X} | \forall i \leq n : f_i(y) \leq g_i(y)\}$$

be the set of all solutions of the system of inequations. Consider the sets of all fixed points of functions h_1 and h_2 defined by

$$h_1(y) = \inf\{f_i^{\perp}(g_i(y))\}, Y_1 = \{y \in \mathcal{X} | h_1(y) = y\}$$

$$h_2(y) = \sup\{g_i^{\top}(f_i(y))\}, Y_2 = \{y \in \mathcal{X} | h_2(y) = y\}.$$

1. If f_i is disjunctive and g_i is monotone $\forall i \leq n$, then $\sup Y \in Y$, $\sup Y_1 \in Y_1$, and $\sup Y = \sup Y_1$.
2. If f_i is monotone and g_i is conjunctive $\forall i \leq n$, then $\inf Y \in Y$, $\inf Y_2 \in Y_2$, and $\inf Y = \inf Y_2$.

Proof summary: Under the stated conditions, the induced functions h_1 and h_2 are monotone. On a complete lattice, monotonicity guarantees the existence of supremal and infimal fixed points [18]. When f and g satisfy these conditions, extremal solutions of (8) exist and correspond to the extremal fixed-points of h_1 and h_2. □

The next theorem uses these functions to compute extremal solutions to (8).

Theorem 4: Consider the system of inequations (8) over a complete lattice $(\mathcal{X}, \leq)$ and the set Y of all solutions of the system.

1. Let f_i be disjunctive and g_i be monotone. Consider the following iterative computation:
   - $y_0 := \sup \mathcal{X}$
   - $y_{k+1} := h_1(y_k)$

   If $y_{m+1} = y_m$ for some $m \in \mathcal{N}$ then $y_m = \sup Y$.
2. Let f_i be monotone and g_i be conjunctive. Consider the following iterative computation:
   - $y_0 := \inf \mathcal{X}$
   - $y_{k+1} := h_2(y_k)$

   If $y_{m+1} = y_m$ for some $m \in \mathcal{N}$ then $y_m = \inf Y$.

Proof summary: For part 1, the stopping condition $y_{m+1} = y_m$ is sufficient to show that $y_m \in Y$. If $z \in Y$ is another solution of the system of inequations, then an inductive argument is used to show that $z \leq y_k \; \forall k \geq 0$. Therefore, $y_m = \sup Y$. Part 2 follows analogously. □

In the controllability condition for logical DES (4) on the lattice $(2^{\Sigma^*}, \subseteq)$ the prefix closure function is disjunctive and monotone and the concatenation function is disjunctive. With some additional modifications Theorems 3 and 4 demonstrate the existence and computation of the supremal controllable sublanguage of any specified language. Formulas for other extremal behaviors, such as those reported in [4], can be derived in a similar manner. Details of this approach may be found in [13]. Next we show that these results can be applied analogously to timed DES using the controllability condition presented in section 3.

B. Timed DES Behavior Specified as a Set

Suppose we are given a set $Y \subseteq \mathcal{S}$ of acceptable sequences. We wish to find the least subset or the greatest superset of Y such that enabling the controllable events at times given by a sequence in the extremal set results in an actual behavior which lies in the extremal set. That is, we seek extremal sets of sequences which are invariant under uncontrollable actions.

In this case, the underlying lattice is $(2^{\mathcal{S}}, \subseteq)$ and the definition of controllability is given by inequation (5). It is of the form

$$f(X) \leq g(X) \tag{9}$$

where $f(\cdot) = A^*(I_c(\cdot) \oplus v)$ and $g(\cdot)$ is the identity function. The identity function is monotone and since f is actually a function on $\mathcal{S}$ extended to $2^{\mathcal{S}}$ we know by Example 4 that it is disjunctive. Furthermore, f is also monotone and the identity function is conjunctive. Therefore Theorem 3 guarantees the existence of both extremal solutions and we make use of Theorem 4 to find these solutions.

Note that Theorem 4 yields the supremal and infimal solutions to (8) over the entire lattice, which in this case are $\mathcal{S}$ and $\emptyset$ respectively. What we really want is the supremal solution of (5) which is contained in a fixed set Y, which is the desired behavior. To find the supremal controllable subset of Y we must find the supremal solution to the pair of inequations

$$\begin{aligned} A^*(I_c X \oplus v) &\leq X \\ X &\leq Y. \end{aligned} \tag{10}$$

The identity function and $f(\cdot) = A^*(I_c(\cdot) \oplus v)$ are disjunctive and the constant function Y is monotone. Therefore, we meet the conditions of the first part of Theorem 3, with

$$h_1(X) = Y \sqcap f^{\perp}(X).$$

Similarly, to find the infimal solution containing Y let the second inequation in (10) be $Y \leq x$. Since the constant function Y is monotone and the identity function is conjunctive we can use the second part of Theorem 3 with

$$h_2(X) = Y \sqcup f(X)$$

where we use the fact that the co-dual of the identity is itself.

Summarizing, we have the following result.

Theorem 5: Given $G = (T, A)$ with controllable events T_c and a set of acceptable behaviors $Y \in \mathcal{S}$, the supremal controllable subset of Y and the infimal controllable superset of Y both exist.

Both of these solutions are computable by the iterative method of Theorem 4. For the special case where the function f is disjunctive and also idempotent the iterative computation reduces to a single step.

Theorem 6: Consider a complete lattice $(\mathcal{X}, \leq)$, a disjunctive and idempotent function f on $\mathcal{X}$, and a fixed $\hat{x} \in \mathcal{X}$.

1. The supremal solution less than $\hat{x}$ of $f(x) \leq x$ is $\hat{x} \sqcap f^{\perp}(\hat{x})$.
2. The infimal solution greater than $\hat{x}$ of $f(x) \leq x$ is $\hat{x} \sqcup f(\hat{x})$.

Proof: For the first part, since f is disjunctive and the identity is monotone it follows from Theorem 3 than the supremal solution less than $\hat{x}$ exists. The iterative computation yields

$$\begin{aligned} y_1 &= \hat{x} \sqcap f^{\perp}(\hat{x}) \\ y_2 &= \hat{x} \sqcap f^{\perp}(\hat{x} \sqcap f^{\perp}(\hat{x})) \\ &= \hat{x} \sqcap f^{\perp}(\hat{x}) \sqcap f^{\perp}(f^{\perp}(\hat{x})) \\ &= \hat{x} \sqcap f^{\perp}(\hat{x}) \\ &= y_1. \end{aligned}$$

The second part follows analogously. □

We have already seen that the function on the left-hand side of the controllability condition (5) is disjunctive. It is also idempotent, allowing us to compute solutions using Theorem 6.

Lemma 1: The function $f : \mathcal{S} \to \mathcal{S}$ defined by $A^*(I_c(\cdot) \oplus v)$ is idempotent.

Proof: We first show that the function $A^* I_c$ is idempotent.

$$\begin{aligned} A^* \geq I &\;\Rightarrow\; A^* I_c A^* \geq A^* I_c \\ &\;\Rightarrow\; A^* I_c A^* I_c \geq A^* I_c \\ I_c \leq I &\;\Rightarrow\; A^* I_c A^* \leq A^* A^* = A^* \\ &\;\Rightarrow\; A^* I_c A^* I_c \leq A^* I_c \end{aligned}$$

Using this fact we have

$$f^2(x) = A^*(I_c(A^*(I_c x \oplus v)) \oplus v)$$

$$\begin{aligned} &= A^*I_cA^*I_cx \oplus A^*I_cA^*v \oplus A^*v \\ &= A^*I_cx \oplus (A^*I_cA^* \oplus A^*)v \\ &= A^*I_cx \oplus A^*(I_c \oplus I)A^*v \\ &= A^*I_cx \oplus A^*v \\ &= f(x). \end{aligned}$$

□

If f is disjunctive but not idempotent, it can be replaced by its disjunctive closure $f^{\sqcup}$. We need the following lemma which is simply an application of [2, Theorem 4.70] to the present situation.

Lemma 2: For a complete lattice $(\mathcal{X}, \leq)$ and a disjunctive function $f : \mathcal{X} \to \mathcal{X}$

$$f(x) \leq x \Leftrightarrow f^{\sqcup}(x) \leq x.$$

Since the disjunctive closure is idempotent and preserves disjunctivity the next corollary is immediate.

Corollary 2: Consider a complete lattice $(\mathcal{X}, \leq)$, a disjunctive function f on $\mathcal{X}$, and a fixed $\hat{x} \in \mathcal{X}$.

1. The supremal solution less than $\hat{x}$ of $f(x) \leq x$ is $\hat{x} \sqcap (f^{\sqcup})^{\perp}(\hat{x})$.
2. The infimal solution greater than $\hat{x}$ of $f(x) \leq x$ is $\hat{x} \sqcup f^{\sqcup}(\hat{x})$.

Example 6: Consider the manufacturing process of Example 2 with the specification set

$$Y = \left\{ \begin{bmatrix} (7\gamma)^*(\bar{0}) \\ s(7\gamma)^*(\bar{0}) \\ ws(7\gamma)^*(\bar{0}) \end{bmatrix}, \begin{bmatrix} (8\gamma)^*(\bar{0}) \\ s(8\gamma)^*(\bar{0}) \\ ws(8\gamma)^*(\bar{0}) \end{bmatrix} \right\} = \{y_1, y_2\}.$$

Since $f(\cdot) = A^*(I_c(\cdot) \oplus v)$ is idempotent we can use Theorem 6 to find the supremal controllable subset of Y. In Example 2 we saw that $f(y_1) \notin Y$, but $f(y_2) = y_2$. Therefore,

$$\begin{aligned} Y \cap f^{\perp}(Y) &= Y \cap \sup\{X \subseteq \mathcal{X} | f(X) \subseteq Y\} \\ &= \{x \in Y | f(x) \in \{y_1, y_2\}\} \\ &= y_2. \end{aligned}$$

Similarly, the infimal controllable superset of Y is given by

$$\begin{aligned} Y \cup f(Y) &= \{y_1, y_2\} \cup \{f(y_1), f(y_2)\} \\ &= \{y_1, y_2, f(y_1)\} \end{aligned}$$

□

C. Behavior Specified as an Extremal Sequence

For general $Y \subseteq \mathcal{S}$ computation of $f^{\sqcup}$ or the iterative set computations of Theorem 4 may become unwieldy. If the desired behavior can be expressed in terms of upper and/or lower bounds in $\mathcal{S}$ such that

$$Y = \{x \in \mathcal{S} | y_1 \leq x \leq y_2\}$$

then an alternative approach based on the lattice $(\mathcal{S}, \leq)$ may be appropriate.

For a set of the form $Y = \{x \in \mathcal{S} | x \leq y\}$ we wish to find the greatest sequence z less than a fixed y such that enabling the controllable events at times less than or equal to z results in actual behavior less than or equal to z. We have already seen from Theorem 2 that in this case the controllability condition is of the form

$$f(x) = A^*(I_cx \oplus v) \leq x.$$

The function f is almost l.s.c (which is equivalent to disjunctivity on the lattice $(\mathcal{S}, \leq)$) but not quite since $f(\varepsilon) \neq \varepsilon$. However, if we require $y \geq A^*v$ then

$$\sup\{x \in \mathcal{S} | A^*(I_c(x) \oplus v) \leq y\} = \sup\{x \in \mathcal{S} | A^*I_c(x) \leq y\}$$

so we can let $f(x) = A^*I_cx$. Now we can apply Corollary 2 directly to find the supremal controllable sequence less than the given y.

Example 7: For the system of Example 2 we find the supremal controllable specification sequence $x \leq y$. Using Theorem 6, this is given by

$$y \wedge f^{\perp}(y) = y \wedge \sup\{x \in \mathcal{S} | A^*I_c(x) \leq y\}.$$

Since only t_2 is controllable the first and third components of x can be made arbitrarily large and so are set equal to $+\infty$. Using the definition of A^* we are left with the task of finding $((r\gamma w)^*)^{\perp}(y_2)$. Using the properties of the dual and Example 5 first observe that

$$\begin{aligned} (0 \oplus r\gamma w)^{\perp}(y_2) &= 0^{\perp}y_2 \wedge (r\gamma w)^{\perp}y_2 \\ &= y_2 \wedge w^{\perp}\gamma^{\perp}r^{\perp}y_2 \\ &= y_2 \wedge \{3, 10, 17, 24, 28, 38, \ldots\} \\ &= \{1, 8, 15, 22, 28, 36, \ldots\} \\ &\equiv x_2. \end{aligned}$$

Also note that

$$0 \oplus r\gamma w \leq (r\gamma w)^* \Rightarrow ((r\gamma w)^*)^{\perp}(y_2) \leq (0 \oplus r\gamma w)^{\perp}(y_2).$$

Since $(0 \oplus r\gamma w)x_2 = (r\gamma w)^*x_2$ we must have $((r\gamma w)^*)^{\perp}(y_2) = x_2$. Finally, the supremal controllable specification sequence less than y is given by

$$y \wedge \begin{bmatrix} \bar{\infty} \\ x_2 \\ \bar{\infty} \end{bmatrix} = \begin{bmatrix} y_1 \\ x_2 \\ y_3 \end{bmatrix}.$$

□

Remark: This formulation of the control problem is similar to the latest start date problem considered in Section 5.6 of [2]. There, the supremal solution to an inequation of the form $A^*Bx \leq y$ is sought. In our notation this is given by $x = (A^*B)^{\perp}(y)$, which agrees with [2]. Note that the latest start date problem is not concerned with specifying the behavior of the uncontrollable events nor with invariance under uncontrollable actions, accounting for the difference in formulation.

On the other hand, for a set of the form $Y = \{x \in \mathcal{S} | x \geq y\}$ an argument along the same lines as Theorem 2 yields

$$A^*(I_cx \oplus v) \geq x \tag{11}$$

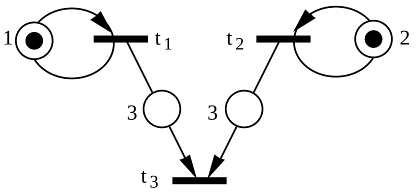

Fig. 4. Timed event graph with no infimal controllable specification.

as the condition for controllability. This is an inequation of the form $f(x) \leq g(x)$ with f being the identity function and $g(\cdot) = A^*(I_c(\cdot) \oplus v)$. While f is monotone, g in this case is by no means necessarily conjunctive. Thus we cannot show the existence of the infimal controllable specification greater than the given sequence y using Theorem 3. In fact the following example shows that this sequence may not exist at all.

Example 8: Consider the system of Figure 4 governed by

$$x = \begin{bmatrix} 1\gamma & \varepsilon & \varepsilon \\ \varepsilon & 2\gamma & \varepsilon \\ 3 & 3 & \varepsilon \end{bmatrix} x \oplus \begin{bmatrix} \bar{0} \\ \bar{0} \\ \bar{0} \end{bmatrix}.$$

For t_1 and t_2 controllable and

$$\hat{x} = \left\{ \begin{bmatrix} \varepsilon \\ \varepsilon \\ 3 \end{bmatrix}, \begin{bmatrix} \varepsilon \\ \varepsilon \\ 6 \end{bmatrix}, \begin{bmatrix} \varepsilon \\ \varepsilon \\ 9 \end{bmatrix}, \ldots \right\}$$

observe that both

$$x_1 = \left\{ \begin{bmatrix} 0 \\ \varepsilon \\ 3 \end{bmatrix}, \begin{bmatrix} 3 \\ \varepsilon \\ 6 \end{bmatrix}, \begin{bmatrix} 6 \\ \varepsilon \\ 9 \end{bmatrix}, \ldots \right\}$$

and

$$x_2 = \left\{ \begin{bmatrix} \varepsilon \\ 0 \\ 3 \end{bmatrix}, \begin{bmatrix} \varepsilon \\ 3 \\ 6 \end{bmatrix}, \begin{bmatrix} \varepsilon \\ 6 \\ 9 \end{bmatrix}, \ldots \right\}$$

are controllable sequences (in the sense of (11)) greater than $\hat{x}$. However, $x_1 \wedge x_2$ is uncontrollable since $A^* I_c(x_1 \wedge x_2) < \hat{x}$. Therefore there is no infimal controllable specification sequence greater that $\hat{x}$. □

Remark: If there is only one controllable event in the system, this is normally sufficient to guarantee that the function $A^*(I_c(\cdot) \oplus v)$ is conjunctive. This is because the delay functions are often also *upper–semicontinuous*, meaning that they distribute over $\wedge$ as well as $\oplus$. When this is the case, the infimal controllable specification sequence exists and can be found in accordance with Theorem 6.

For those cases in which it is possible to find both the supremal controllable z_2 less than y_2 and the infimal controllable z_1 greater than y_1, then the supremal controllable subset of $Y = \{x \in \mathcal{S} | y_1 \leq x \leq y_2\}$ is given by $Z = \{x \in \mathcal{S} | z_1 \leq x \leq z_2\}$. This is true because any sequence in Z must yield a behavior in the intersection of the two extremal controllable sets.

If the existence of the infimal controllable specification cannot be guaranteed one practical method is to find the supremal controllable specification less than y_2 and check if the resulting behavior is greater than y_1. This suboptimal approach yields one controllable behavior (if any exist) which meets the specification. This method is demonstrated in [7].

D. Behavior Specified as a Single Sequence

Controllability constraints on the lattice $(\mathcal{S}, \leq)$ examined thus far allow us only to specify bounds on the occurrence times of events. Thus enabling the controllable events at times within the specified range guarantees that all events will occur at times within that range. Suppose instead we wish to find a single extremal controllable sequence as opposed to a range or set of sequences invariant under uncontrollable events. That is, we seek a sequence such that enabling the controllable events at the designated times causes all events to actually occur at their designated times, rather that merely occurring within the specified set.

Using the results of section 3 the controllability of a single sequence is specified by requiring that $A^*(I_c x \oplus v) = x$. If there is an upper limit $\hat{x}$ on acceptable behavior, the optimal behavior is the supremal controllable sequence less than $\hat{x}$. This sequence is the supremal solution to the set of inequations

$$\begin{aligned} A^*(I_c x \oplus v) &\leq x \\ x &\leq A^*(I_c x \oplus v) \\ x &\leq \hat{x}. \end{aligned}$$

As before, if we require that $\hat{x} \geq A^* v$ then the first inequation may be replaced by

$$A^* I_c x \leq x.$$

Now all the left-hand side functions are disjunctive (l.s.c.) and all the right-hand side functions are monotone. Thus, we can apply Theorem 4 with

$$h_1(y) = \hat{x} \wedge A^*(I_c y \oplus v) \wedge (A^* I_c)^\perp(y)$$

to find the desired behavior. Since the function $A^*(I_c(\cdot) \oplus v)$ is not necessarily conjunctive, we will not consider infimal behavior here.

Example 9: Consider an upper limit such that $\hat{x} = A^* \hat{x}$. Then

$$\begin{aligned} y_1 &= h_1(\sup \mathcal{S}) = \hat{x} \\ y_2 &= h_1(\hat{x}) = \hat{x} \wedge A^*(I_c \hat{x} \oplus v) \wedge (A^* I_c)^\perp(\hat{x}). \end{aligned}$$

Our restrictions on $\hat{x}$ imply that

$$\hat{x} \geq A^* I_c \hat{x}$$

and therefore

$$(A^* I_c)^\perp(\hat{x}) \geq \hat{x}.$$

We also have

$$A^*(I_c \hat{x} \oplus v) \leq \hat{x}$$

and so $y_2 = A^*(I_c \hat{x} \oplus v)$. Furthermore, because of the idempotency of the function $A^*(I_c(\cdot) \oplus v)$ it follows that

$$y_3 = h_1(A^*(I_c \hat{x} \oplus v)) = A^*(I_c \hat{x} \oplus v)$$

which shows that this is the optimal behavior. □

There are many cases in which it is also important to consider the separation times between events. In [15] bounds on event separation times are determined for acyclic graphs while in [1] this is extended to cyclic timed event graphs (there called *process graphs*). Our interest is to determine whether a specified limit on event separation times can be achieved given the control available in the system.

The ability to guarantee minimal separation times between certain events could be useful for ensuring sufficient time to perform an in-process inspection in a manufacturing system or to provide sufficient time between arrivals and departures of connecting trains or airplanes. The ability to guarantee maximal separation times between certain events could be useful to avoid timeouts between communicating processors.

Let D be a matrix of delay functions which specifies separation times between events. A solution of the inequation

$$Dx \leq x \tag{12}$$

guarantees separation times of at least D. If the delay functions in D are l.s.c. then D is disjunctive and inequation (12) meets the conditions of the first part of Theorem 3. Thus for the supremal controllable sequence less than some $\hat{x}$ which satisfies the minimal separation time requirement of (12) we have

$$h_1(y) = \hat{x} \wedge D^{\perp}(y) \wedge A^*(I_c y \oplus v) \wedge (A^* I_c)^{\perp}(y).$$

Therefore this supremal behavior exists and is computed by iterating h_1.

A solution of the inequation

$$x \leq Dx \tag{13}$$

guarantees separation times of no more than D. Since D is also monotone, inequation (13) also meets the conditions of the first part of Theorem 3. In this case, the supremal controllable sequence less than $\hat{x}$ which satisfies the maximal separation time requirement of (13) gives

$$h_1(y) = \hat{x} \wedge Dy \wedge A^*(I_c y \oplus v) \wedge (A^* I_c)^{\perp}(y).$$

Therefore this supremal behavior also exists and is computed by iterating h_1.

Example 10: For the manufacturing process of Figure 2 suppose now that t_1 is controllable instead of t_2. This means that part arrivals from the rest of the factory may be inhibited. We find the supremal controllable sequence $x \leq \hat{x}$ where

$$\hat{x} = \begin{bmatrix} 8(8\gamma)^*(\bar{0}) \\ 8(8\gamma)^*(\bar{0}) \\ 8(8\gamma)^*(\bar{0}) \end{bmatrix}$$

and such that the separation between $x_3(k-1)$ and $x_1(k)$ does not exceed 3 for all k. This requirement ensures that the process does not sit idle for longer than 3 time units. This is specified by the inequation

$$x \leq Dx = \begin{bmatrix} \varepsilon & \varepsilon & 3\gamma \\ \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & 0 \end{bmatrix} x \oplus w$$

where w is a constant sequence with $w_1(0) = +\infty$ and equal to 0 elsewhere. This modification is necessary to indicate that there is no separation constraint on $x_1(0)$, the first firing of t_1.

Now applying the iteration scheme of Theorem 4 we find

$$\begin{aligned} y_0 &= \sup \mathcal{S} \\ y_1 &= \hat{x} \\ y_2 &= \begin{bmatrix} 3(8\gamma)^*(\bar{0}) \\ 8(8\gamma)^*(\bar{0}) \\ 8(8\gamma)^*(\bar{0}) \end{bmatrix} \\ y_3 &= \begin{bmatrix} 3(8\gamma)^*(\bar{0}) \\ 4(8\gamma)^*(\bar{0}) \\ 8(8\gamma)^*(\bar{0}) \end{bmatrix} \\ y_4 &= y_3. \end{aligned}$$

Therefore this is the supremal controllable behavior satisfying the required conditions. □

V. Conclusion

For timed DES the objective of supervisory control is to impose delays on controllable events to modify system behavior to meet some specified performance goal. Using a max-algebra representation it is possible to compute the uncontrolled behavior of a timed event graph, define a specification for the desired behavior, and determine whether the specification can be realized by any supervisor given the set of controllable events. The behavioral constraints take the form of inequations over a complete lattice. These constraints may specify minimal or maximal separation times between events as well as bounds on their absolute occurrence times. We have shown that extremal solutions to these inequations can be found using lattice theoretic methods developed for studying logical (untimed) DES behaviors.

References

[1] T. Amon, H. Hulgaard, S. M. Burns, G. Borriello, "An algorithm for exact bounds on the time separation of events in concurrent systems," in *IEEE International Conference on Computer Design*, October 1993.

[2] F. Baccelli, G. Cohen, G. J. Olsder, J. P. Quadrat, *Synchronization and Linearity: An Algebra for Discrete Event Systems*, Wiley, New York, 1992.

[3] B. A. Brandin, W. M. Wonham, "Supervisory Control of Timed Discrete-Event Systems," *IEEE Transactions on Automatic Control*, vol. 39, no. 2, pp. 329–342, 1994.

[4] R. D. Brandt, V. K. Garg, R. Kumar, F. Lin, S. I. Marcus, W. M. Wonham, "Formulas for calculating supremal and normal sublanguages," *Systems and Control Letters*, 15(8):111–117, 1990.

[5] D. D. Cofer, V. K. Garg, "A timed model for the control of discrete event systems involving decisions in the max-plus algebra," in *Proceedings of the 31st IEEE Conference on Decision and Control*, Tuscon, AZ, pp. 3363-3368, 1992.

[6] D. D. Cofer, V. K. Garg, "A Generalized Max-algebra Model for Timed and Untimed DES," in *Proceedings of the 1993 American Control Conference*, San Francisco, CA, pp. 2288–2292, June 1993.

[7] D. D. Cofer, V. K. Garg, "A Max-algebra Solution to the Supervisory Control Problem for Real-Time Discrete Event Systems," in *Lecture Notes in Control and Information Sciences 199: 11th International Conference on Analysis and Optimization of Systems*, G. Cohen, J. P. Quadrat, eds., Springer–Verlag, 1994.

[8] G. Cohen, D. Dubois, J. P. Quadrat, M. Viot, "A linear-system-theoretic view of discrete-event processes and its use for per-

formance evaluation in manufacturing," *IEEE Transactions on Automatic Control*, vol. AC-30, no. 3, pp. 210-220, March 1985.

[9] G. Cohen. P. Moller, J. P. Quadrat, M. Viot, "Algebraic tools for the performance evaluation of DES," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 39–58, 1989.

[10] R. A. Cuninghame–Green, *Minimax Algebra*, Springer–Verlag, Berlin, 1979.

[11] P. Glasserman, D. D. Yao, *Monotone Structure in Discrete-event Systems*, Wiley, New York, 1994.

[12] B. H. Krogh, "Controlled Petri Nets and Maximally Permissive Feedback Logic,"*Proc. 25th Allerton Conference on Communication, Control, and Computation*, Urbana, IL, 1987.

[13] R. Kumar, V. K. Garg, "Extremal Solutions of Inequations over Lattices with Applications to Supervisory Control," in *Proceedings of the 33rd IEEE Conference on Decision and Control*, Orlando, FL, pp. 3636–3641, December 1994.

[14] R. Kumar, V. K. Garg, *Modeling and Control of Logical Discrete Event Systems*, Kluwer, Boston, 1995.

[15] K. McMillan, D. L. Dill, "Algorithms for interface timing verification," in *1992 IEEE International Conference on Computer Design*, October 1992.

[16] J. S. Ostroff, "Deciding properties of timed transition models," *IEEE Transactions on Parallel and Distributed Systems*, vol. 1, no. 2, pp. 170–183, 1990.

[17] P. J. Ramadge, W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[18] A. Tarski, "A Lattice-theoretical Fixpoint Theorem and its Applications," *Pacific Journal of Mathematics*, vol. 5, pp. 285–309, 1955.

[19] E. Wagneur, "Moduloids and pseudomodules: 1. Dimension Theory," *Discrete Mathematics*, vol. 98, 57–73, 1991.

Darren D. Cofer received the B.S. degree in electrical engineering from Rice University in 1985 and the M.S. and Ph.D. degrees in electrical and computer engineering from The University of Texas at Austin in 1992 and 1995, respectively.

From 1985 to 1990 he served as a Naval officer at the Nuclear Propulsion Directorate in Washington, DC directing the development, production, and testing of nuclear power plant components for U.S. Navy ships. He is currently a principal research scientist at the Honeywell Technology Center in Minneapolis, MN. His research interests include the modelling, analysis, and control of discrete event systems and real-time embedded software.

Vijay K. Garg received his Bachelor of Technology degree in computer engineering from the Indian Institute of Technology, Kanpur in 1984 and the M.S. and Ph.D. degree in electrical engineering and computer science from the University of California at Berkeley in 1985 and 1988, respectively.

He is currently an associate professor in the Department of Electrical and Computer Engineering at the University of Texas, Austin where he holds the General Motors Centennial Fellowship. His research interests are in the areas of distributed systems and supervisory control of discrete event systems. He has authored or co-authored more than seventy research articles in these areas. He is the author of the book *Principles of Distributed Systems* and a co-author of the book *Modeling and Control of Logical Discrete Event Systems*, both published by Kluwer Academic Publishers. He is a recipient of TRW Faculty Assistantship Award.