K. M. Passino, A. N. Michel and P. J. Antsaklis, "Lyapunov Stability of a Class of Discrete Event Systems," P roc o f t he 1 991 A merican C ontrol C o nference , pp 2911-2916, Boston, MA, June 26-28,1991.

Proc. of the American Control Conference, Boston, 1991

LYAPUNOV STABILITY OF A CLASS OF DISCRETE EVENT SYSTEMS

Kevin M. Passino*
Dept. of Electrical Engineering
The Ohio State University
2015 Neil Ave.
Columbus, OH 43210

Anthony N. Michel* and Panos J. Antsaklis**
Dept. of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Abstract

Discrete event systems (DES) are dynamical systems which evolve in time by the occurrence of events at possibly irregular time intervals. "Logical" DES are a class of discrete time DES with equations of motion that are most often non-linear and discontinuous with respect to event occurrences. Recently, there has been much interest in studying the stability properties of logical DES and several definitions for stability, and methods for stability analysis have been proposed. Here we introduce a logical DES model and define *stability in the sense of Lyapunov* for logical DES. Then we show that a more conventional analysis of stability which employs appropriate Lyapunov functions can be used for logical DES. This standard approach has the advantage of not requiring high computational complexity (as some of the others) but the difficulty lies in specifying the Lyapunov functions. The approach is illustrated on a manufacturing system that processes batches of N different types of parts according to a priority scheme, one of Dijkstra's "self-stabilizing" distributed systems, and a load balancing problem in computer networks.

1.0 Introduction

Discrete event systems (DES) are dynamical systems which evolve in time by the occurrence of events at possibly irregular time intervals. Some examples include flexible manufacturing systems, computer networks, logic circuits, and traffic systems. "Logical" DES are a class of discrete time DES with equations of motion that are most often non-linear and discontinuous in the occurrence of the events. Recently, there has been much interest in studying the stability properties of logical DES and several definitions for stability, and methods for stability analysis have been proposed. Here we introduce a logical DES model and define *stability in the sense of Lyapunov* for logical DES. Then we show that the metric space formulation in [30] can be adapted so that a conventional analysis of stability which employs appropriate Lyapunov functions can be used for logical DES. An important advantage of the Lyapunov approach is that it does not require high computational complexity (as some of the other new approaches) but the difficulty lies in specifying the Lyapunov function. The approach is illustrated on a manufacturing system that processes batches of N different types of parts according to a priority scheme, one of Dijkstra's "self stabilizing" distributed systems [7], and a "load balancing problem" in computer networks. The full version of this paper which includes the proofs in given in [31].

* The work of K.M. Passino and A.N. Michel was supported in part by the National Science Foundation under Grant ECS88-02924. Please address all correspondence to K.M. Passino (email: passino@eagle.eng.ohio-state.edu).
** The work of P.J. Antsaklis was supported in part by the Jet Propulsion Laboratory.

K. M. Passino, A. N. Michel and P. J. Antsaklis, "Lyapunov Stability of a Class of Discrete Event Systems," P roc o f t he 1 991 A merican C ontrol C o nference , pp 2911-2916, Boston, MA, June 26-28,1991.

The foundations for the study of stability properties of logical DES lie in the areas of general stability theory (the approach used herein) and theoretical Computer Science (recent DES-theoretic research). The two (related) main areas in theoretical Computer Science that form the foundation for logical DES-theoretic stability studies are temporal logic and automata. Intuitively speaking, in a temporal logic or automata-theoretic framework a system is considered in some sense *stable* if (i) for some set of initial states the system's state is guaranteed to enter a given set and stay there forever, or (ii) for some set of initial states the system's state is guaranteed to visit a given set of states infinitely often.

In temporal logic, stability characteristics are most often represented with temporal formulas from a linear or branching time language (modal logics) and either a proof system or effective procedure is used to verify that the temporal formula is satisfied [9,15,28,11,12,18,5,22]. Stability concepts for logical DES such as finite automata have foundations in the study of, for instance, Buchi and Muller automata [4,17], and how infinite strings are accepted by such automata. This automata theoretic work in Computer Science has also been adapted for the study of stability of DES. In [20] the authors introduce a special DES model (finite automaton) and use a state-space approach to develop efficient algorithms for the study of the two types of stability described above. They also provide approaches to synthesize stabilizing controllers for DES and to study several other characteristics of logical DES (for more details see [19]). Related studies are given in [3,26,13]. Certainly, results in the Ramadge-Wonham framework [27] can be utilized for the study of types of stability of logical DES.

Certain general formulations for the study of stability are relevant to the study of stability properties of logical DES. For instance, there have been studies of stability of asynchronous iterative processes in [29]. For an introduction to general stability theory and an overview of such research see [16]. Finally, in other DES studies, there have recently been significant advances in the study of stability properties of manufacturing systems in [24,14].

In Section 2 we introduce a logical DES model and compare it to several other models to help characterize the class of systems that we can represent and study. In Section 3 we define stability in the sense of Lyapunov for DES and give necessary and sufficient conditions for stability of invariant sets of DES in a metric space. The applications are given in Section 4 and Conclusions in Section 5.

2.0 A Discrete Event System Model

We will consider stability properties of discrete event systems that can be accurately modelled with

$$G=(\mathcal{X},\mathcal{E},f_e,g,E_v) \qquad (1)$$

where $\mathcal{X}$ is the set of states, $\mathcal{E}$ is the set of *events*,

$$f_e:\mathcal{X}\rightarrow\mathcal{X} \qquad (2)$$

for $e\in\mathcal{E}$ are operators,

$$g:\mathcal{X}\rightarrow\mathbb{P}(\mathcal{E})-\{\emptyset\} \qquad (3)$$

is the *enable function*, and $E_v\subset\mathcal{E}^{\mathbb{N}}$ is the set of *valid event trajectories*. Here, for an arbitrary set Z, $\mathbb{P}(Z)$ denotes the power set of Z. We only require that $f_e(x)$ be defined when $e\in g(x)$. We associate "time" indices with the states and events so that $x_k\in\mathcal{X}$ represents the state at time $k\in\mathbb{N}$ and $e_k\in\mathcal{E}$ represents an *enabled* event at time $k\in\mathbb{N}$ if $e_k\in g(x_k)$. If at state $x_k\in\mathcal{X}$, event $e_k\in\mathcal{E}$ *occurs* at time $k\in\mathbb{N}$ (randomly, not necessarily according to any particular statistics) then the next state x_{k+1} is given by application of the operator f_{e_k}, i.e., $x_{k+1}=f_{e_k}(x_k)$. Events can only occur if they lie on valid event trajectories as we now discuss.

Any sequence $\{x_k\}\in\mathcal{X}^{\mathbb{N}}$ such that for all k, $x_{k+1}=f_{e_k}(x_k)$ where $e_k\in g(x_k)$, is a *state trajectory*. The set of all *event trajectories* denoted with $\mathbf{E}$ is composed of those sequences $\{e_k\}\in\mathcal{E}^{\mathbb{N}}$ such that there exists a state trajectory $\{x_k\}\in\mathcal{X}^{\mathbb{N}}$ where for all k, $e_k\in g(x_k)$. Hence, to each event trajectory, which specifies the order of the application of the operators f_e, there corresponds a unique state trajectory (but, in general, not vice versa). The set of *valid event trajectories* $E_v\subset\mathbf{E}$ represents the event trajectories that are physically possible in G. Hence, even if $x_k\in\mathcal{X}$ and $e_k\in g(x_k)$ it is not the case that e_k can occur unless it lies on a valid event trajectory that ends at x_{k+1}, where $x_{k+1}=f_{e_k}(x_k)$. In Section 4 we shall see that the use of E_v can facilitate the modelling of many DES and provide flexibility in the study of stability properties. Let $E_v(x_0)\subset E_v$ denote the set of all possible valid event trajectories that begin from state $x_0\in\mathcal{X}$. Below, we shall also utilize a special set of *allowed event trajectories* denoted with E_a, where $E_a\subset E_v$, and allowed event trajectories that begin at state $x_0\in\mathcal{X}$ denoted by $E_a(x_0)$.

Let E_k, for fixed $k\in\mathbb{N}$, denote an event sequence of k events that have occurred (by definition $E_0=\emptyset$). If $E_k=e_0,e_1,\cdots,e_{k-1}$ we let $E_kE\in E_v(x_0)$ denote the *concatenation* of E_k and (the infinite sequence) $E=e_k,e_{k+1},\cdots$, i.e. $E_kE=e_0,e_1,\cdots,e_{k-1},e_k,e_{k+1},\cdots$. The function $X(x_0,E_k,k)$ will be used to denote the state reached from $x_0\in\mathcal{X}$ by application of event sequence E_k such that $E_kE\in E_v(x_0)$. (By definition, $X(x_0,\emptyset,0)=x_0$ for all $x_0\in\mathcal{X}$.) For fixed x_0 and E_k, $X(x_0,E_k,k)$ shall be called a *motion*. We assume that for all $x_0\in\mathcal{X}$, if $E_kE_{k'}E\in E_v(x_0)$

$$X(X(x_0,E_k,k),E_{k'},k')=X(x_0,E_{k+k'},k+k') \qquad (4)$$

for all $k,k'\in\mathbb{N}$, such that $k'\geq k$ where $E_{k+k'}=E_kE_{k'}$. This is the standard semi-group property for dynamical systems.

Notice that any system that can be represented with the General and Extended Petri nets [25] can also be represented with our DES model (1). (To see this, choose $\mathcal{X}=\mathbb{N}^n$, where n is the number of *places*, let the *transitions* be events, g defines how transitions are *enabled*, and f_e defines what happens when transitions are *fired*). There are many models that have been compared to the General and Extended Petri net so this helps to characterize the generality of our model in relation to other logical DES models [25,21] and it shows that our logical DES model is quite general and hence our results will apply to a wide class of DES.

3.0 Necessary and Sufficient Conditions for the Stability of Invariant Sets of DESs in a Metric Space

The following adapts the formulation developed in [30] to the study of stability properties of systems represented by the logical DES model introduced above. Let $\rho:\mathcal{X}\times\mathcal{X}\rightarrow\mathbb{R}$ denote a *metric* on $\mathcal{X}$, and $\{\mathcal{X};\rho\}$ a *metric space.* Let $\mathcal{X}_z\subset\mathcal{X}$ and $\rho(x,\mathcal{X}_z)=\inf\{\rho(x,x'):x'\in\mathcal{X}_z\}$ denote the *distance* from point x to the set $\mathcal{X}_z$. By a *functional* we shall mean a mapping from an arbitrary set to $\mathbb{R}$.

<u>Definition 1</u>: The *r-neighborhood* of an arbitrary set $\mathcal{X}_z\subset\mathcal{X}$ is denoted by the set $S(\mathcal{X}_z;r)=\{x\in\mathcal{X}:0<\rho(x,\mathcal{X}_z)<r\}$ where $r>0$.

<u>Definition 2</u>: The set $\mathcal{X}_m\subset\mathcal{X}$ is called *invariant* with respect to (w.r.t) G if from $x_0\in\mathcal{X}_m$ it follows that $X(x_0,E_k,k)\in\mathcal{X}_m$ for all E_k such that $E_kE\in E_v(x_0)$ and $k\in\mathbb{N}$.

<u>Definition 3</u>: A closed invariant set $\mathcal{X}_m\subset\mathcal{X}$ of G is called *stable in the sense of Lyapunov w.r.t.* E_a if for any $\varepsilon>0$ it is possible to find a quantity $\delta>0$ such that when $\rho(x_0,\mathcal{X}_m)<\delta$ we have $\rho(X(x_0,E_k,k),\mathcal{X}_m)<\varepsilon$ for all E_k such that $E_kE\in E_a(x_0)$ and $k\in\mathbb{N}$. If furthermore $\rho(X(x_0,E_k,k),\mathcal{X}_m)\rightarrow 0$ for all E_k such that $E_kE\in E_a(x_0)$ as $k\rightarrow\infty$, then the closed invariant set $\mathcal{X}_m$ of G is called *asymptotically stable w.r.t.* E_a.

<u>Definition 4</u>: A closed invariant set $\mathcal{X}_m\subset\mathcal{X}$ of G is called *unstable in the sense of Lyapunov w.r.t.* E_a if it is not stable in the sense of Lyapunov w.r.t E_a.

<u>Definition 5</u>: If the closed invariant set $\mathcal{X}_m\subset\mathcal{X}$ of G is asymptotically stable in the sense of Lyapunov w.r.t. E_a, then the set $\mathcal{X}_a$ of all states $x_0\in\mathcal{X}$ and $x_0\notin\mathcal{X}_m$ having the property $\rho(X(x_0,E_k,k),\mathcal{X}_m)\rightarrow 0$ for all E_k such that $E_kE\in E_a(x_0)$ as $k\rightarrow\infty$ is called the *region of asymptotic stability of* $\mathcal{X}_m$ *w.r.t.* E_a.

Definition 6: The closed invariant set $\mathcal{X}_m \subset \mathcal{X}$ of G with region of asymptotic stability $\mathcal{X}_a$ w.r.t. $\mathbf{E_a}$ is called *asymptotically stable in the large w.r.t.* $\boldsymbol{E_a}$ if $\mathcal{X}_a = \mathcal{X}$.

The following Theorem provides necessary and sufficient conditions for stability of the DES defined in (1).

Theorem 1: In order for a closed invariant set $\mathcal{X}_m \subset \mathcal{X}$ of G to be stable in the sense of Lyapunov w.r.t $\mathbf{E_a}$ it is necessary and sufficient that in a sufficiently small neighborhood $S(\mathcal{X}_m;r)$ of the set $\mathcal{X}_m$ there exists a specified functional V with the following properties:

(i) For sufficiently small $c_1>0$, it is possible to find a $c_2>0$ such that $V(x)>c_2$ for $x \in S(\mathcal{X}_m;r)$ and $\rho(x,\mathcal{X}_m)>c_1$.

(ii) For any $c_4>0$ as small as desired, it is possible to find a $c_3>0$ so small that when $\rho(x,\mathcal{X}_m)<c_3$ for $x \in S(\mathcal{X}_m;r)$ we have $V(x) \leq c_4$.

(iii) $V(X(x_0,E_k,k))$ is a non-increasing function for $k \in \mathbb{N}$, for $x_0 \in S(\mathcal{X}_m;r)$, for all $k \in \mathbb{N}$, as long as $X(x_0,E_k,k) \in S(\mathcal{X}_m;r)$ for all E_k such that $E_kE \in \mathbf{E_a}(x_0)$.

Corollary 1: If the closed invariant set $\mathcal{X}_m \subset \mathcal{X}$ of G is stable in the sense of Lyapunov w.r.t $\mathbf{E_a}$ then it is stable in the sense of Lyapunov w.r.t all $\mathbf{E'_a}$ such that $\mathbf{E'_a} \subset \mathbf{E_a}$.

Theorem 2: In order for a closed invariant set $\mathcal{X}_m \subset \mathcal{X}$ of G to be asymptotically stable in the sense of Lyapunov w.r.t. $\mathbf{E_a}$ it is necessary and sufficient that in a sufficiently small neighborhood $S(\mathcal{X}_m;r)$, of the set $\mathcal{X}_m$ there exists a specified functional V having properties (i), (ii), and (iii) of Theorem 1 and furthermore $V(X(x_0,E_k,k)) \rightarrow 0$ as $k \rightarrow \infty$ for all E_k such that $E_kE \in \mathbf{E_a}(x_0)$ and for all $k \in \mathbb{N}$ as long as $X(x_0,E_k,k) \in S(\mathcal{X}_m;r)$.

Corollary 2: If the closed invariant set $\mathcal{X}_m \subset \mathcal{X}$ of G is asymptotically stable w.r.t $\mathbf{E_a}$ then it is asymptotically stable w.r.t all $\mathbf{E'_a}$ such that $\mathbf{E'_a} \subset \mathbf{E_a}$.

4.0 Discrete Event System Applications

In this Section we show how to perform conventional Lyapunov stability analysis for three types of DES applications: (i) a manufacturing system that processes batches of N different types of parts according to a priority scheme, (ii) one of Dijkstra's "self stabilizing" distributed systems, and (iii) a load balancing problem in computer networks. In each case we specify the logical DES model (1) and the invariant set $\mathcal{X}_m$, pick the metric ρ, choose the Lyapunov function V(x), then show that V(x) satisfies the appropriate

K. M. Passino, A. N. Michel and P. J. Antsaklis, "Lyapunov Stability of a Class of Discrete Event Systems," P roc o f t he 1 991 A merican C ontrol C o nference , pp 2911-2916, Boston, MA, June 26-28,1991.

properties. Detailed comparisons to similar applications found in the literature are given throughout.

4.1 Manufacturing System

The first example that we shall consider is the manufacturing system shown in Figure 1 that processes batches of N different types of parts according to a priority scheme. There are N *producers* P_i, where $1 \leq i \leq N$, of parts of different types. The producers P_i place batches of their parts in their respective *buffers* B_i, where $1 \leq i \leq N$. These buffers B_i have *safe capacity limits* of b_i where $b_i > 0$, $1 \leq i \leq N$. Let x_i, $1 \leq i \leq N$, denote the number of parts in buffer B_i. Let x_i for $N+1 \leq i \leq 2N$ denote the number of P_{i-N} type parts in the machine. The machine can safely process less than or equal to M (where $M>0$) parts of any type, at any time. As the machine finishes processing batches of P_i type parts they are placed in their respective *output bins* (P_i-bins). The producers P_i can only place batches of parts in their buffers B_i if $x_i < b_i$. Also, there is a priority scheme whereby batches of P_i type parts are only allowed to enter the machine if $x_j = 0$ for all j such that $j < i \leq N$, i.e. only if there are no parts in any buffers to the left of the B_i buffer. Next, we specify the DES model G for the manufacturing system.

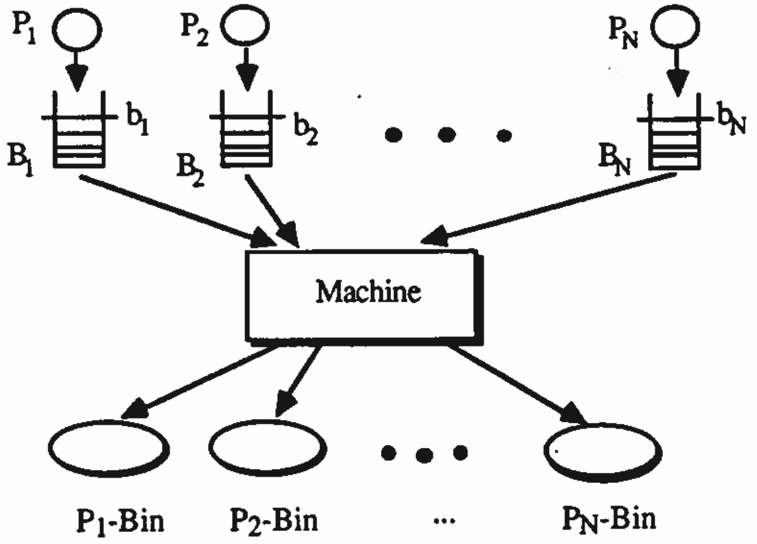

Figure 1. Manufacturing System with Priority Batch Processing

Let $\mathcal{X} = \mathbb{N}^{2N}$ and $x_k \in \mathcal{X}$, where $x_k = [x_1\ x_2 \cdots x_N\ x_{N+1}\ x_{N+2} \cdots x_{2N}]^t$ (t denotes transpose) denote the state at time k. Let the set of events $\mathcal{E}$ be composed of events e_{pi} for $1 \leq i \leq N$ (representing the case where producer P_i places a batch of α_{pi} parts in buffer B_i), events e_{ai} for $1 \leq i \leq N$ (representing the case where a batch of α_{ai} P_i parts, from buffer B_i, arrive at the machine), and events e_{di} for $N+1\le i\le 2N$ (representing the case where a batch of α_{di} P_i parts depart from the machine and are placed in their respective output bins). According to the above specifications the enable function g and event operators f_e for $e\in g(x_k)$ are defined as follows:

(i) If $x_i<b_i$ for any i, $1\le i\le N$, then $e_{pi}\in g(x_k)$ and

$f_{e_{pi}}(x_k)=[x_1\ x_2 \cdots x_i+\alpha_{pi} \cdots x_N\ x_{N+1}\ x_{N+2} \cdots x_{2N}]^t$,

where $\alpha_{pi}\in \mathbb{N}-\{0\}$, $\alpha_{pi}\le|x_i-b_i|$.

(ii) If $\sum_{j=N+1}^{2N} x_j<M$, and for some i, $1\le i\le N$, $x_i>0$, and $x_\ell=0$ for all ℓ, $\ell<i\le N$, then $e_{ai}\in g(x_k)$ and

$f_{e_{ai}}(x_k)=[x_1 \cdots x_i-\alpha_{ai} \cdots x_N\ x_{N+1}\ x_{N+2} \cdots x_{N+i}+\alpha_{ai} \cdots x_{2N}]^t$,

where $\alpha_{ai}\in \mathbb{N}-\{0\}$, $\alpha_{ai}\le x_i$, and

$$\alpha_{ai}\le\left|\sum_{j=N+1}^{2N} x_j - M\right|.$$

(iii) If $x_i>0$ for any i, $N+1\le i\le 2N$, then $e_{di}\in g(x_k)$ and

$f_{e_{di}}(x_k)=[x_1\ x_2 \cdots x_N\ x_{N+1}\ x_{N+2} \cdots x_{N+i}-\alpha_{di} \cdots x_{2N}]^t$

where $\alpha_{di}\in \mathbb{N}-\{0\}$ and $\alpha_{di}\le x_{N+i}$.

We let $E_v=E$, i.e. the set of all event trajectories is defined by g and f_e for $e\in g(x_k)$. The system operates in a standard asynchronous fashion.

This manufacturing system is a generalization of computer systems often used in the study of a simple "mutual exclusion problem" in Computer Science [25,15], and similar to several applications studied in the DES literature. For instance, if $\alpha_{pi}=\alpha_{ai}=\alpha_{di}=1$ then our manufacturing system is similar to the "Two Class Parts Processing" example in [28] (except they allow an arbitrary finite number of parts to enter their machine and consider only two producers), and the manufacturing system example in [11,12] (they also consider only two producers). For M=1, N=2, $\alpha_{pi}=\alpha_{ai}=\alpha_{di}=1$, and the same priority scheme, the system is also similar to the one used in the study of the "buffered producer/consumer problem with a shared channel" described in [25].

Let

$$\mathcal{X}_{m1}=\left\{x_k\in \mathcal{X}: x_i\le b_i\ \forall i,\ 1\le i\le N, \text{ and } \sum_{j=N+1}^{2N} x_j \le M\right\}. \tag{5}$$

It is easy to see that $\mathcal{X}_{m1}$ is invariant by letting $x_k\in \mathcal{X}_{m1}$ and showing that no matter which event occurs it is the case that the next state $x_{k+1}\in \mathcal{X}_{m1}$. The invariance of $\mathcal{X}_{m1}$ is the property of the manufacturing system that has been studied extensively in similar manufacturing system examples [28,11,12]. Also, if M=1, N=2, $\alpha_{pi}=\alpha_{ai}=\alpha_{di}=1$, and the priority scheme is removed, then the proof of the invariance of $\mathcal{X}_{m1}$ is equivalent to proving the mutual exclusion property often studied in Computer Science mentioned above.

Here, we provide a new study of the stability properties of the above manufacturing system. Intuitively this will, for instance, show that under certain conditions if the manufacturing system starts in an unsafe operating mode (too many parts in a buffer or in the machine, or both) it will eventually return to a safe operating condition. This is more carefully quantified in the following propositions.

Proposition 1: For the manufacturing system, the closed invariant set $\mathcal{X}_{m1}$ is stable in the sense of Lyapunov w.r.t $\mathbf{E}_a$, where $\mathbf{E}_a=\mathbf{E}_v$, for any neighborhood of $\mathcal{X}_{m1}$ (i.e., any $r>0$).

Proof: Let $x_k=[x_1 \cdots x_{2N}]^t$, $x_{k+1}=[x'_1 \cdots x'_{2N}]^t$, $\bar{x}=[\bar{x}_1 \cdots \bar{x}_{2N}]^t$, and $\bar{x}'=[\bar{x}'_1 \cdots \bar{x}'_{2N}]^t$. Choose

$$\rho(x_k,\mathcal{X}_{m1})=\inf\left\{\sum_{j=1}^{2N}|x_j-\bar{x}_j|:\bar{x}\in\mathcal{X}_{m1}\right\} \qquad (6)$$

and $V_1(x_k)=\rho(x_k,\mathcal{X}_{m1})$ and then it is easy to show that this V_1 satisfies the conditions of Theorem 1 for all $r>0$. The details of the proof for this Theorem are contained in [23]. ■

Proposition 2: For the manufacturing system, the closed invariant set $\mathcal{X}_{m1}$ is not asymptotically stable in the large w.r.t. $\mathbf{E}_a$, where $\mathbf{E}_a=\mathbf{E}_v$.

Let $\mathbf{E}_a'\subset\mathbf{E}_a$ denote the set of event trajectories such that each event $e\in\mathcal{E}$ occurs infinitely often on each event trajectory $E\in\mathbf{E}_a'$. If we assume for the manufacturing system that only events which lie on event trajectories in $\mathbf{E}_a'$ occur, then it is always the case that eventually each type of event (e_{pi}, e_{ai}, $1\leq i\leq N$, and e_{di}, $N+1\leq i\leq 2N$) will occur.

Proposition 3: For the manufacturing system, the closed invariant set $\mathcal{X}_{m1}$ is asymptotically stable in the large w.r.t $\mathbf{E}_a'$.

The use of the set $\mathbf{E}_a'$ for the manufacturing system imposes what is called a "fairness" constraint in Computer Science (in our example we require that each producer P_i get fair use of the machine) [8]. Such constraints are used in the study of temporal logic [5,15], the mutual exclusion problem in Computer Science, and in [29] when the author studies conditions under which the Lyapunov function can be constructed mechanically for a class of logical DES.

4.1 Dijkstra's Self-Stabilizing Distributed System

In [7] Dijkstra proposed three candidate systems as being "self-stabilizing" but provided no proof of this property. Using a linear temporal logic approach (a sound proof system) Thistle and Wonham [28] showed that with minor corrections the first of Dijkstra's systems was in fact self-stabilizing. Dijkstra's second system is studied with a branching-time temporal logic approach in [22]. Here, we use the approach of Section 3 to study the stability properties of the invariant set given by Thistle and Wonham for Dijkstra's first system.

The distributed system that Dijkstra proposes is a network of N+1 machines arranged in a "ring". He denotes machine n with m_n and each m_n has as neighbors machines $m_{(n-1)\mathrm{mod}(N+1)}$ and $m_{(n+1)\mathrm{mod}(N+1)}$. The distributed system is shown in Figure 2. The state of the machine m_n is represented with $x_n \in \mathbb{N}$, where $x_n < K$ for some K, $K \geq N$. We must define when there can be communication in the ring and what will be transferred from machine to machine. This is done by specifying G for the distributed system.

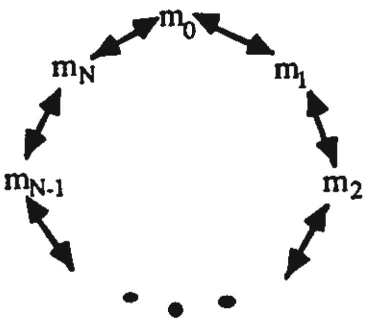

Figure 2. Distributed System

Let $\mathcal{X}=\{x \in \mathbb{N}^{N+1}: x=[x_0\ x_1 \cdots x_N]^t$, each $x_i<K$, where $K \geq N\}$ be the set of states. The set of events is given by $\mathcal{E}=\{e_i: 0 \leq i \leq N\}$ and the occurrence of e_i will indicate that a communication between machines occurs. Next we define g and f_e for all $e \in g(x)$. Assume $x_k=[x_0\ x_1 \cdots x_N]^t$ and $x_{k+1}=[x'_0\ x'_1 \cdots x'_N]^t$ and let "x:=y" denote the assignment of x to the value of y, then

(i) If $x_N=x_0$, then $e_0 \in g(x_k)$ and $f_{e_0}(x_k)=x_{k+1}$ where for i, $1 \leq i \leq N$,

$x'_i := x_i$ and $x'_0 := (x_0+1)\mathrm{mod}K$, and

(ii) If $x_i \neq x_{i-1}$ for any i, $1 \leq i \leq N$, then $e_i \in g(x_k)$ and $f_{e_i}(x_k)=x_{k+1}$

where $x'_i := x_{i-1}$ and $x'_j := x_j$ for all $j \neq i$.

With a minor correction to Dijkstra's example Thistle and Wonham provide the following invariant set $\mathcal{X}_{m2}$,

$\mathcal{X}_{m2}=\{x_k \in \mathcal{X}: x_i=x_0\ \forall i,\ 0 \leq i \leq N$, or if $\exists$ n,$0 \leq n \leq N$,$x_i=x_0\ \forall i,\ 0 \leq i \leq n$

$$\text{and } x_j=(x_0-1)\text{mod}K,\ \forall j,\ n<j\leq N\} \tag{7}$$

and explain how this is an invariant set. We choose $\mathbf{E}_v=\mathbf{E}$, and note that each event $e_i \in \mathcal{E}$ will occur infinitely often on each event trajectory in $\mathbf{E}_v$ (since for no event to be enabled it must be the case that $x_0 \neq x_N$ and $x_i=x_{i-1}$ for all i, $1\leq i\leq N$ which can never be true).

The following Proposition and its proof provide a more conventional characterization and analysis of the stability properties of Dijkstra's first system than in [28].

<u>Proposition 1</u>: For Dijkstra's distributed system the closed invariant set $\mathcal{X}_{m2}$ is asymptotically stable in the large w.r.t. $\mathbf{E}_a$, where $\mathbf{E}_a=\mathbf{E}_v$.

Proof: Let $\bar{x}=[\bar{x}_1 \cdots \bar{x}_N]^t$, $\bar{x}'=[\bar{x}'_1 \cdots \bar{x}'_N]^t$,

$$\rho(x_k,\mathcal{X}_{m2})=\inf\left\{\max\left\{|x_0-\bar{x}_0|,|x_1-\bar{x}_1|,\cdots,|x_N-\bar{x}_N|\right\}:\ \bar{x}\in\mathcal{X}_{m2}\right\}, \tag{8}$$

and $V_2(x_k)=\rho(x_k,\mathcal{X}_{m2})$. This choice results in the satisfaction of conditions (i) and (ii) of Theorem 1. For condition (iii) of Theorem 1 it is shown in [31] that for all $x_k \notin \mathcal{X}_{m2}$ and all $e \in g(x_k)$ when e occurs and $x_{k+1}=f_e(x_k)$, then $V_2(x_k)\geq V_2(x_{k+1})$. Asymptotic stability follows from the fact that the state will never return to a previous value as long as it is outside the invariant set. ■

4.3 Computer Network Load Balancing Problem

Consider a network of computers described by an undirected graph (C,A) where $C=\{1,2,\ldots,N\}$ represents a set of computers that are numbered with $i\in C$, and $A\subset C\times C$ is the set of connections between the computers. We require that if $i\in C$ then there exists $(i,j)\in A$ (or $(j,i)\in A$) for some $j\in C$ (i.e., every computer is connected to the network). Also, if $(i,j)\in A$ then $(j,i)\in A$. Each computer has a buffer which holds tasks (load) each of which can be executed by any computer in the network. Let the load of computer $i\in C$ be given by x_i; hence, $x_i\geq 0$. Each connection in the network $(i,j)\in A$ allows for computer i to pass a portion of its load to computer j and vice-versa. It also allows computer i and computer j to sense the size of the load of one another (any two computers i and j such that $(i,j)\notin A$ may not pass portions of their loads or sense the values of each others loads).

We assume that initially the distribution of the load across the computers is uneven and seek to prove properties relating to the system achieving a more even distribution of tasks so that the computers in the network are more fully utilized. For convenience we assume that the computers will not begin working on any of the tasks or receive any more to process until the load has been balanced (This assumption is easily lifted and our analysis still applies as we discuss below.). Next we specify the model G for the computer network load balancing problem.

Let $\mathcal{X}=\mathbb{N}^N$ denote the set of states and $x_k=[x_1\ x_2 \cdots x_N]^t$ and $x_{k+1}=[x'_1\ x'_2 \cdots x'_N]^t$ denote the state at time k and k+1 respectively. Let $\mathcal{E}=\{e^{ij}_\alpha\!: (i,j)\in A, \alpha\in \mathbb{N}-\{0\}\}\cup\{e_0\}$ be the set of events for G where e^{ij}_α represents the case where α amount of load of computer i is transferred to computer j and e_0 represents an null event (i.e., that no load is transfered). Let $M\in \mathbb{N}-\{0\}$ be the amount of load imbalance tolerated between any two computers i and j where $(i,j)\in A$. (The case where M=0 is discussed below.) Next we specify g and f_e for $e\in g(x_k)$:

(i) If for any $(i,j)\in A$, $|x_i-x_j|>M$ then

(a) if $x_i>x_j$, then $e^{ij}_\alpha\in g(x_k)$ and $f_e(x_k)=x_{k+1}$ where $e=e^{ij}_\alpha$, $x'_i:=x_i-\alpha$, $x'_j:=x_j+\alpha$, $x'_k:=x_k$ for all $k\neq i,j$, and $0<\alpha\leq(1/2)|x_i-x_j|$ for $\alpha\in\mathbb{N}$.

(b) if $x_j>x_i$, then $e^{ji}_\alpha\in g(x_k)$ and $f_e(x_k)=x_{k+1}$ where $e=e^{ji}_\alpha$, $x'_j:=x_j-\alpha$, $x'_i:=x_i+\alpha$, $x'_k:=x_k$ for all $k\neq i,j$, and $0<\alpha\leq(1/2)|x_i-x_j|$ for $\alpha\in\mathbb{N}$.

(ii) If $|x_i-x_j|\leq M$ for all $(i,j)\in A$ then $e_0\in g(x_k)$ and $f_{e_0}(x_k)=x_k$.

Let $E_v=E$ and $\mathcal{X}_{m3}=\{x_k\in\mathcal{X}: |x_i-x_j|\leq M \text{ for all } (i,j)\in A\}$ which is clearly invariant.

This load balancing problem is similar to the one in [1] except they require the load of the computers to be represented by a continuous variable, seek a perfect balance of tasks, allow a computer to simultaneously pass load to several neighboring computers, and allow for the possibility that a computer's information about the load of adjacent computers is outdated. They also require that the system is "partially asynchronous" so that they can achieve load balancing when the computers only have possibly outdated information about neighboring loads. Various forms of the load balancing problem have also been studied in the DES literature [2] and extensively studied in the Computer Science literature (See [1,2,6] and the references therein).

The following Proposition provides a new characterization and analysis of the Lyapunov stability of the computer network load balancing problem described above.

Proposition 5: For the computer network load balancing problem, the closed invariant set $\mathcal{X}_{m3}$ is asymptotically stable in the large w.r.t. $\mathbf{E}_a$, where $\mathbf{E}_a=\mathbf{E}_v$.

Proof: Let $\bar{x}=[\bar{x}_1 \cdots \bar{x}_N]^t$, $\bar{x}'=[\bar{x}'_1 \cdots \bar{x}'_N]^t$, and choose

$$\rho(x_k,\mathcal{X}_{m3})=\inf\{\max\{|x_1-\bar{x}_1|,\cdots,|x_N-\bar{x}_N|\}: \bar{x}\in\mathcal{X}_{m3}\} \tag{9}$$

and $V_3(x_k)=\rho(x_k,\mathcal{X}_{m3})$ so that conditions (i) and (ii) of Theorem 1 are satisfied. For condition (iii) of Theorem 1 it is shown in [31] that for all $x_k\notin\mathcal{X}_{m3}$ and all $e^{ij}_\alpha\in g(x_k)$ when e^{ij}_α occurs $V_3(x_k)\geq V_3(x_{k+1})$ and that the invariant set is asymptotically stable in the large w.r.t. $\mathbf{E}_a$. ■

If tasks enter the computer network or get processed by one of the computers $i \in C$ we let a new initial state x_0 reflect the increased or decreased load and the above stability analysis shows that the load will still eventually balance. (This characteristic was also discussed in [1].)

Suppose that we had allowed M=0 and assumed that (C,A) was also *strongly connected.* The region of asymptotic stability of the subsequent invariant set $\mathcal{X}'_{m3}=\{x_k:x_i=x_j \text{ for all } (i,j)\in A\}$ is given by

$$\mathcal{X}_a=\left\{x_k:\left(\sum_{i=1}^{N} x_i\right)\mathrm{mod}N=0\right\}.$$

That is, the region of asymptotic stability is given by the set of all initial states such that the total load can be divided evenly among the N computers $i \in C$.

5.0 Conclusions

It has been shown that it is possible to define and study Lyapunov stability of a wide class of logical DES (e.g., DES modelled with Extended Petri nets, finite automata, and others) by adapting the metric space formulation in [30]. Hence, logical DES, which have recently received much attention in the literature are amenable to conventional stability analysis via the choosing of appropriate Lyapunov functions. Other notions of stability and more recent stability analysis techniques based on methods from theoretical Computer Science (surveyed in the Introduction) are often prohibitive due to problems with computational complexity. Here, we completely avoid problems with computational complexity but instead rely on the specification of Lyapunov functions that satisfy certain properties. We have shown that it is not difficult to specify such Lyapunov functions for three types of DES applications: a manufacturing system that processes batches of N different types of parts according to a priority scheme, one of Dijkstra's self stabilizing distributed systems, and a load balancing problem in computer networks. Our characterization of stability of DES in a traditional stability-theoretic framework will, in the future, allow researchers to use the vast body of concepts from the field of Lyapunov stability theory to study properties of DES.

References

[1] Bertsekas D.P., Tsitsiklis J.N., Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, Engelwood Cliffs, NJ, 1989.

[2] Boel R.K., van Schuppen J.H., "Distributed Routing for Load Balancing", Proceedings of the IEEE, Vol. 77, No. 1, pp. 210-221, 1989.

[3] Brave Y., Heymann M., "On Stabilization of Discrete Event Processes", Proceeding of the 28th Conf. on Dec. and Control, pp. 2737-2742, Tampa, FL, Dec. 1989.

[4] Buchi J.R., "On a Decision Method in Restricted Second Order Arithmetic", Proc. of the Int. Congress on Logic, Mathematics, and Phil. of Sci., 1960, Stanford Univ. Press, Stanford, CA, 1962.

[5] Clarke E.M., Browne M.C., Emerson E.A., Sistla A.P., "Using Temporal Logic for Automatic Verification of Finite State Systems", Apt K.R., ed., Logics and Models of Concurrent Systems, Springer Verlag, NY, 1985.

[6] Cybenko G., "Dynamic Load Balancing for Distributed Memory Multi-Processors", Dept. of Computer Science Technical Report 87-1, Tufts Univ., Medford MA, 1987. (To appear in the Journal of Parallel and Distributed Computing.)

[7] Dijkstra E.W., "Self-stabilizing Systems in Spite of Distributed Control", Comm. of the ACM, Vol. 17, No. 11, pp. 643-644, Nov. 1974.

[8] Francez N., Fairness, Springer-Verlag, NY, 1986.

[9] Fusaoka A., Seki H., and Takahashi K., "A Description and Reasoning of Plant Controllers in Temporal Logic", Proc. of the 8th Int. Joint Conf. on Artificial Intelligence, pp. 405-408, August 1983.

[10] Hahn W., Stability of Motion, Springer Verlag, NY, 1967.

[11] Knight J.F., Passino K.M., "Decidability for Temporal Logic in Control Theory", Proc. of the Allerton Conf. on Communication, Control, and Computing, Univ. of Illinois, Urbana-Champaign, pp. 335-344, Oct. 1987.

[12] Knight J.F., Passino K.M., "Decidability for a Temporal Logic Used in Discrete Event System Analysis", To appear in the Int. Journal of Control 1990.

[13] Krogh B.H., "Controlled Petri Nets and Maximally Permissive Feedback Logic", Proc. of the 25th Allerton Conf., Univ. of Illinois at Urbana-Champaign, Sept. 1987.

[14] Kumar P.R., Seidman T.I., "Dynamic Instabilities and Stabilization Methods in Distributed Real-Time Scheduling of Manufacturing Systems", IEEE Trans. on Automatic Control, Vol. 35, No. 3, March 1990.

[15] Manna Z., Pnueli A., "Verification of Concurrent Programs: a Temporal Proof System", Dept. of Computer Science, Stanford Univ., Report No. STAN-CS-83-967, 1983.

[16]. Michel A.N., Miller R.K., Qualitative Analysis of Large Scale Dynamical Systems, Academic Press, NY, 1977.

[17] Muller D.E., "Infinite Sequences and Finite Machines", Switching Circuit Theory and Logic Design, Proc. Fourth Annual Symp., IEEE, Chicago, IL, 1963.

[18] Ostroff J. S., Real-Time Computer Control of Discrete Systems Modelled by Extended State Machines: A Temporal Logic Approach, PhD Dissertation, Report No. 8618, Dept. of Elect. Eng., Univ. of Toronto, Jan. 1987.

[19] Ozveren C.M., Analysis and Control of Discrete Event Dynamic Systems: A State Space Approach, Ph.D. Dissertation, MIT, LIDS Report LIDS-TH-1907, Aug. 1989.

[20] Ozveren C.M., Willsky A.S., Antsaklis P.J., "Stability and Stabilizability of Discrete Event Dynamic Systems", MIT, LIDS Report LIDS-P-1853, Feb. 1989 (To appear in the Journal of the ACM).

[21] Passino K.M., "The Input/Output Petri Net for Discrete Event System Modelling", Control Systems Technical Note #69, Dept. of Electrical Eng., Univ. of Notre Dame, June 1989.

[22] Passino K.M., Antsaklis P.J., "Branching Time Temporal Logic for Discrete Event System Analysis", Proc. of the 26th Allerton Conf. on Communication, Control, and Computing, pp. 1160-1169, Univ. of Illinois at Urbana-Champaign, Sept. 1988.

[23] Passino K.M., Michel A.N., Antsaklis P.J., "Stability Analysis of Discrete Event Systems", Proc. of the 28th Allerton Conf. on Communication, Control, and Computing, pp. 487-496, Univ. of Illinois at Urbana-Champaign, Oct. 1990.

[24] Perkins J.R., Kumar P.R., "Stable, Distributed, Real-Time Scheduling of Flexible Manufacturing/Assembly/Disassembly Systems", IEEE Trans. on Automatic Control, Vol. 34, No. 2, Feb. 1989.

[25] Peterson J.L., Petri Net Theory and the Modeling of Systems, Prentice Hall, NJ, 1981.

[26] Ramadge P.J., "Some Tractable Supervisory Control Problems for Discrete-Event Systems Modeled by Buchi Automata", IEEE Transactions on Automatic Control, Vol. 34, No. 1, pp. 10-19, Jan. 1989.

[27] Ramadge P.J., Wonham W.M., "Supervisory Control of a Class of Discrete Event Processes", SIAM J. Control and Optimization, Vol. 25, pp. 206-230, Jan. 1987.

[28] Thistle J. G., Wonham W. M., "Control Problems in a Temporal Logic Framework", International Journal Control, Vol. 44, No. 4, pp. 943-976, 1986.

[29] Tsitsiklis J.N., "On the Stability of Asychronous Iterative Processes", Mathematical Systems Theory, Vol. 20, pp. 137-153, 1987.

[30] Zubov V.I., Methods of A.M. Lyapunov and Their Application, P. Noordhoff Ltd., The Netherlands, 1964.

[31] Passino K.M., Michel A.N., Antsaklis P.J., "Lyapunov Stability of a Class of Discrete Event Systems", Submitted for journal publication, 1990.