Abstract—Dissipativity is a useful tool for analyzing and synthesizing stable feedback systems. Briefly, the property of dissipativity is that a system only stores and dissipates energy, with respect to a given energy storage function, and does not generate its own energy. This property carries a strong reliance on a notion of time that prohibits its direct application to discrete event systems (DES). The current paper takes the concept of dissipativity and applies it to DES by replacing the notion of time with an event-based definition. Two notions of dissipativity are defined, one for finite automata and the other for a general class of DES. For each notion, properties of dissipative DES are shown that connect with existing notions of stability for DES as would be expected from the classical definition of dissipativity. Examples are provided to illustrate the methods covered in the paper.

I. INTRODUCTION

The notion of dissipativity is a powerful tool for analysis of dynamical systems [1], [2]. It provides methods of assessing a large range of dynamical behaviors by simply varying the energy supply rate in the dissipative inequality. These behaviors typically involve the input-output behavior of a system, but can also include the behavior of internal state, see e.g. [3], [4]. The theory provides stability results when considering two continuous time or two discrete time systems in feedback, however this traditional notion does not directly apply when one of the systems is a discrete event system (DES) as is the case when computers are used for control. This problem brings up the question of whether dissipativity can be applied to DES in a meaningful way. The ability to capture properties of DES by specifying a dissipative supply rate appears to be a promising approach.

DES are an important class of systems that have discrete states and laws dictating transitions between these states. Important classes of DES with a finite number of states include finite state machines and finite automata [5]. Petri nets are an important class of DES that allow for infinite states [6], [7]. Other classes of DES that operate over infinite time horizons include Büchi automata [8], [9] and Muller automata [10]. Some existing work focuses instead on more general models of DES that allow for infinite states or infinite executions [11]. It is important to note that DES are very different from discrete-time linear or nonlinear systems. While discrete-time systems follow dynamic motion models based on a notion of time, DES follow logical transitions that occur without a dependence on time. As such, the existing notions of dissipativity for discrete-time systems are not applicable to DES. DES are often controlled using an approach referred to as supervisory control where state transitions are allowed or disallowed based on additional considerations. There are several results for supervisory control of DES modeled using finite automata [12], [13] or Petri nets [7], [14], [15].

This paper covers two original definitions of dissipativity for DES. The first definition applies to finite automata based on discrete state transition behavior. The second is defined for a more general class of DES that includes both DES with infinite states and DES with infinite executions. The goal is to capture the benefits of traditional dissipativity, such as stability results and an intuitive energy-based property, and expand it to this important class of logical systems modeled as DES. Key results are presented to make such connections and utilize the intuitive nature of dissipativity. Parts of this work are contained within [16], but are previously unpublished.

The remainder of the paper is organized as follows. Section II introduces the finite automata model and the more general DES model. Section III provides a new definition of dissipativity for finite automata including definitions of energy storage and energy supplied. Section IV presents two results that show implications of this notion of dissipativity. Some additional examples are given in Section V on dissipativity for finite automata. The paper goes on to introduce a new definition of dissipativity for general DES in Section VI with a connection to Lyapunov stability for DES. The paper closes with concluding remarks provided in Section VII.

II. BACKGROUND ON DISCRETE EVENT SYSTEMS

Discrete event systems (DES) are systems that have discrete states and a set of events that cause transitions between these states. Common examples of DES are finite state machines, finite automata [5], and Petri nets [6], [7]. The discrete states can be numbered or labeled without the label necessarily implying a physical quantity. While a dynamical system having the value 2 typically has a physical interpretation, e.g. position or velocity, in DES the state value of 2 is often just a numbered state. Additionally, the state of a dynamical system is real-valued and evolves according to a vector field. The state of a DES is discrete valued. The system will stay in a given state for an unspecified amount of time and will transition to the next state instantaneously. In this sense, time is not relevant for DES. This section will introduce some background material for finite automata and then for a more general class of DES.
A. Background on Finite Automata

This section focuses on DES modeled using the finite automata model [5]. This model captures DES with a finite number of states that halt execution after a finite number of events have occurred.

**Definition 1.** A finite automaton is defined by the five-tuple: 
\[ A = \{Q, \Sigma, \alpha, q_0, F\} \]
where
- \(Q\) is the set of discrete states,
- \(\Sigma\) is the set of discrete events,
- \(\alpha : Q \times \Sigma \to Q\) is the set of possible state transitions,
- \(Q_0 \subset Q\) is the set of initial states, and
- \(F\) is the set of final states.

The state of the automaton is only allowed to take on one value in \(Q\) at a time. The automaton must begin in a state in \(Q_0\) and end in an element of the final state set \(F\). The occurrence of an event \(\epsilon \in \Sigma\) causes a transition to a state in \(Q\). The set of transitions \(\alpha\) enumerates all possible transitions from the current state to the next state. For example, the state may transition from state \(q\) to state \(q'\) caused by event \(\epsilon \in \Sigma\) if \(q' \in \alpha(q, \epsilon)\). The next state \(q'\) may take on any value in \(Q\) including the current state \(q\). In the case that the automaton is deterministic, the transition relation will specify exactly one next state per event given the current state, i.e. \(q' = \alpha(q, \epsilon)\). Assuming this is true for all states, and the initial set \(Q_0\) contains exactly one member, the automaton is deterministic; otherwise it is nondeterministic. For the remainder of this paper, we will assume the finite automata of interest are deterministic. This assumption is made since it is always possible to generate an equivalent deterministic finite automaton from a non-deterministic automaton [5].

It should be noted that there is an equivalence between finite-automata and regular languages. Since regular languages are not the focus of this paper, the interested reader is directed to [5], [17] for more detail. Some notions related to regular languages that will be used are notions of an alphabet or word.

**Definition 2.** The event set \(\Sigma\) of a finite automaton is also known as an alphabet in a formal language.

**Definition 3.** A word \(u : \mathbb{Z} \to \Sigma\) indicates the sequence of events from alphabet \(\Sigma\) that occurred. The elements are ordered and each event must be allowed to occur at the given position based on the current state of the automata.

The notion of a word can be used to track the execution history of discrete events that have occurred. We add an index to the word \(u\) to indicate specific events. For example, \(u(1)\) indicates the first event, \(\epsilon_1\), and \(u(k)\) indicates the occurrence of the \(k\text{th}\) event, \(\epsilon_k\). It is important to note that \(k\) is only for ordering and has no dependence on physical time as it would for discrete time dynamical systems. Sometimes it is necessary to differentiate between a word that is accepted, i.e. ends in a state in \(F\), or not accepted. The history of states visited can be similarly tracked and indicated.

**Definition 4.** A run \(r : \mathbb{Z} \to Q\) indicates the sequence of states from \(Q\) that occurred. Each state in the run must be allowed at the given position when considering the previous state and the allowable transitions in \(\alpha\).

The definition of finite automata assumes that the process halts execution after a finite number of transitions. For many controlled DES, it is not desirable to stop execution at an arbitrary point. Instead, the process should continue indefinitely, typically until a human operator stops the machine. Systems like these are better described by Büchi automata [9]. However, we will consider finite automata without losing much generality since a finite automaton may restart after successfully completing a process.

One main focus on control of DES has been on the notion of supervisory control. Rather than a controller that forces state transitions to occur, a supervisor only disallows transitions that may be unsafe or undesirable. This allows for the DES to continue operation with the fewest restrictions in place while still guaranteeing desirable operation. Such a supervisory controller is referred to as maximally permissive [7]. There are several results for supervisory control of DES whether modeled using finite automata [12], [13] or Petri nets [7], [14], [15]. The structure of a supervisory control scheme can be illustrated in Fig. 1. The supervisor tracks the event sequence, \(u \subset \Sigma^*\), of the plant automaton \(A\) and generates a set of allowable subsets that is in the power series \(2^{\Sigma}\). The default is for the supervisor to not restrict the allowable events, i.e. \(f(\cdot) = \Sigma\).

![Fig. 1. Example of a finite automata with supervisor \(f\).](image)

B. Background on General DES

While finite automata make up an important class of DES, the set of all DES is much larger. This subsection covers a general definition of DES and a notion of Lyapunov stability [11].

**Definition 5.** A general discrete event system (DES) is defined by the five-tuple:
\[ D = \{X, \Sigma, f, g, E_v\} \]
where
- \(X\) is the set of discrete states,
- \(\Sigma\) is the set of discrete events,
- \(f : X \times \Sigma \to 2^X\) is the set of possible state transitions,
- \(g : X \to 2^\Sigma\) is the set of enabled events for a state \(x \in X\), and
- \(E_v \subset \Sigma^* \cup \Sigma^\omega\) is the set of valid event sequences.
Note that $2^\Sigma$ is the power set of $\Sigma$, the set of all subsets of $\Sigma$. The set of discrete states $X$ may be finite or infinite. For a given state $x \in X$, $g(x)$ indicates which events are enabled, i.e. allowed to occur. For an event $e \in g(x)$, the set of next states is $f(x, e) \subseteq 2^X$. For a deterministic DES, for all $x \in X$ and $e \in g(x)$, the set $f(x, e)$ has exactly one element. This state can be denoted $x'$ and the transition can be written, $x \xrightarrow{e} x'$. In addition to the possibly infinite state set $X$, the length of valid event sequences may be finite or infinite. The set $E_v$ captures event sequences that start in a valid initial state and follow transitions of a DES given by the functions $f$ and $g$. Not all sequences in $E_v$ are accepted by the DES.

**Definition 6.** The set of accepted event sequences $E_a \subseteq E_v$ is the set of event sequences accepted by a given DES $D$.

For finite automata, accepted words are ones that terminate in an accepting state. For Büchi automata, accepted strings are those that are in $X$ for all $k > 0$ ($x_0, x_k \in X$) for every $k > 0$ such that $E_k \in E_v(x_0)$ where $E$ is an infinite event sequence.

The notion of Lyapunov stability can be defined with respect to an invariant set $X_e$ using a metric $\rho$.

**Definition 9.** The set $X_i \subseteq X$ is invariant with respect to DES $D$ if $\forall x_0 \in X_i$, then for all $k > 0$ ($x_0, x_k \in X_i$) for all $E_k \in E_v(x_0)$ where $E$ is an infinite event sequence.

The theorem of Lyapunov stability can be defined with respect to an invariant set $X_e$ using a metric $\rho$.

**Definition 10.** Consider a DES $D$ with a closed invariant set $X_e \subseteq X$. $D$ is stable in the sense of Lyapunov if for all $\epsilon > 0$, there exists a $\delta > 0$ such that $\rho(X(x_0, E_k), X_e) < \epsilon$ for all $E_k$ such that $E_k \in E_v(x_0)$ for infinite $E$. If, in addition, $\rho(X(x_0, E_k), X_e) \rightarrow 0$ as $k$ goes to infinity, then $D$ is asymptotically stable.

**III. DISSIPATIVITY FOR FINITE AUTOMATA**

There are some concepts that must be defined in order to apply dissipativity to DES. The system must have a notion of internally stored energy, traditionally captured by an energy storage function $V$. There must also be a measure of external energy supply given by an energy supply rate $\omega$. This is based on a model of a system that has an input $u \in U \subset \mathbb{R}^m$ and output $y \in Y \subset \mathbb{R}^p$. While dissipativity does not require a notion of internal state, this is useful to consider if asymptotic stability is of interest. This internal state can be noted $x \in X \subseteq \mathbb{R}^n$ with an equilibrium $x_{eq} \in X$. When dissipativity is assessed around this equilibrium, stability of the state may be considered. The internally stored energy is a function of state ($V : X \rightarrow \mathbb{R}^+$) while the energy supply rate may be a function of input, output, and state ($\omega : U \times Y \times X \rightarrow \mathbb{R}$).

Before dissipativity is defined, the notion of equilibrium should be clarified. For a dynamical system, the equilibrium is a state or a compact set of states having the property that the unfixed state remains at the equilibrium for all time. For DES it is typically not desirable to stay in a single state for all time. This may be an indication of a fault. Instead, it is often desirable to stay within a specified set of discrete states or continue along a specified trajectory of states indefinitely. For a DES, rather than defining the equilibrium as a single state it can be defined with respect to an invariant set of states or to a limit cycle. The equilibrium can be defined arbitrarily as long as it satisfies desired behaviors for the specific application. The equilibrium set of a finite automaton will be denoted $Q_e$, where $Q_e \subseteq Q$.

For dissipativity, the finite state space $Q$ of an automaton can replace the internal state space $X \subseteq \mathbb{R}^n$ of a dynamical system. An energy storage function is required to capture the notion of internally stored energy.

**Definition 11.** An energy storage function $V : Q \rightarrow \mathbb{R}^+$ for a finite automaton $A = \{Q, \Sigma, \alpha, q_0, F\}$ captures a notion of internal energy storage. As a notion of energy, the storage function must be non-negative, i.e. $V(q) \geq 0$, $\forall q$. Additionally, it is assumed that:

- $V(q) = 0$ for all $q \in Q_e$ and
- $V(q) > 0$, $\forall q \in Q \setminus Q_e$.

As the storage function $V$ is defined on a domain of discrete states, it can be defined by a table rather than a true function.
Similarly, there are no continuity conditions on the discrete energy for DES. The only limitation is that the energy of all reachable states is finite. This notion of energy storage function for DES is similar to previous work on Lyapunov stability for DES [11].

The occurrence of an event that transitions the system from a state in \( Q_e \) to a state outside \( Q_e \) must have an increase of internal energy. Likewise, a transition to the equilibrium set must have a decrease of energy. While a system stays within its equilibrium set, the change in internally stored energy is zero. The rate energy is supplied to the system is a function of the event and the current state, \( \omega : \Sigma \times Q \rightarrow \mathbb{R} \). This energy supply rate can be defined for a particular system in order to characterize the system’s behavior or to synthesize a supervisory controller.

**Definition 12.** A finite automaton \( A = (Q, \Sigma, \alpha, Q_0, F) \) is dissipative with respect to supply rate \( \omega(u, q) \) if there exists a non-negative energy storage function \( V(V(q) \geq 0, \forall q \in Q) \) such that the following inequality holds \( \forall K \geq 0, \):

\[
\sum_{k=0}^{K-1} \omega(u(k), q(k)) \geq V(q(K)) - V(q(0)).
\]

This definition is a statement that the energy added over any time period (\( \sum \omega(u(k), q(k)) \)) always bounds the energy stored over that time period (\( V(q(K)) - V(q(0)) \)) which keeps the general notion of dissipativity in tact from traditional dynamical systems.

This concept can be illustrated with an example of a finite automaton (Fig. 2) with states \( Q = \{0, 1, 2, 3, 4\} \) and transitions \( \Sigma = \{a, b, c\} \). The initial set \( Q_0 \) is indicated by the arrow that does not originate from another state. The set of final states \( F \) is indicated by double circles. The transitions in \( \alpha \) are indicated by edges of the graph. For example, the transition from state 0 to state 1 happens when the automata is in state 0 and event \( a \) occurs.

![Fig. 2. Example of a finite automata with five states, alphabet \{a, b, c\}, and equilibrium \( Q_e = \{0, 1, 2\} \).](image)

The equilibrium cycle is made up of states \( \{0, 1, 2\} \) which coincides with the final state set marked with double circles. The energy of those states is zero while the energy in the other states is larger than zero. A possible storage function can be visualized in Fig. 3 where \( V_3 > V_4 > 0 \).

![Fig. 3. Example of a storage function for the given DES.](image)

This example is dissipative is the supply rate \( \omega(c, \cdot) \) is chosen as follows.

\[
\begin{align*}
\omega(a, 0) &= \omega(b, 1) = \omega(c, 2) = 0 \\
\omega(c, 0) &\geq V(3) \\
\omega(c, 3) &\geq V(4) - V(3) \\
\omega(c, 4) &\geq V(3) - V(4) \\
\omega(a, 4) &\geq -V(4)
\end{align*}
\]

If instead we choose, for example, \( \omega(c, 0) < V(3) \) then this example is not dissipative unless a supervisory control scheme enforces that event \( c \) cannot occur while the system is in state 0. This illustrates that the dissipative supply rate can be chosen to characterize the existing system behavior or the desired system behavior.

**IV. RESULTS FOR DISSIPATIVITY OF FINITE AUTOMATA**

A definition of dissipativity for DES is only useful if certain results can be shown from the property. For continuously-varying systems, being dissipative with respect to a specific supply rate may imply passivity, stability, or stability of interconnections. For finite automata, dissipativity may imply properties that hold for all runs of the DES or properties that hold when DES are combined. One notion of stability for DES is that staying in an invariant set, i.e. all executions that enter the set stay there, implies that the DES is stable. It will be assumed that when dissipativity is applied to invariant sets that the storage function does not increase as the system transitions towards the equilibrium set. This can be assumed without loss of generality and will be more formally explored in Section VI. Briefly, the notion is that storage functions should generally decrease as states move towards the equilibrium and increase as it moves away, which is intuitive when compared to traditional dissipativity.

**Theorem 1.** Consider a dissipative finite automaton with a desired equilibrium set \( Q_e \). This finite automaton has energy storage function \( V(q) \) and a supply rate \( \omega(q, e) \). The equilibrium of the finite automaton is an invariant set if \( \omega(q, e) = 0 \) for all \( q \in Q_e \) and all \( e \in \Sigma \).

*Proof.* For all runs of \( A \) that reach a state \( q \in Q_e \), the energy change of future transitions must be zero by \( \omega(q, e) = 0, \forall q \in Q_e \). By the definition of \( V \), \( V(q) = 0, \forall q \in Q_e \) and \( V(q) > 0, \forall q \in (Q \setminus Q_e) \). With zero energy change, the run may not leave \( Q_e \).

An example of this result is given in the next section.
While set invariance is an important notion of stability for DES, another one is staying within a limit cycle. This notion captures DES that successfully complete a sequence of events whenever starting the cycle. This requires an additional assumption.

**Assumption 1.** There exists an upper bound \( N \) such that, starting at arbitrary point \( i \), when \( q(k) \in \{ Q \backslash Q_e \} \) for all \( k \in (i, i + N) \) then

\[
\sum_{k=i}^{i+N} \omega(u(k), q(k)) < 0.
\]  

(3)

**Theorem 2.** Consider a finite automaton that is dissipative with respect to a limit cycle \( Q_e \). The automaton has energy storage function \( V(q) \) and energy supply rate \( \omega(q, e) \). The limit cycle of the finite automaton is stable if \( \omega(q, e) = 0 \) for all \( q \in Q_e \) and all \( e \in \Sigma \) and Assumption 1 holds.

Proof: Using the result of Theorem 1, once the state is in \( Q_e \), it will stay there for all time. Using Assumption 1 guarantees that some progress is made towards the limit cycle every \( N \) steps. This avoids the case of cycling through a level set outside \( Q_e \) indefinitely.

Note that the bound \( N \) may be used to guarantee a convergence rate to the limit cycle depending on the magnitude of initially stored energy, \( V(q_0) \). This result may be applied for actual behavior of an automaton or desired behavior using a supervisory control scheme.

V. EXAMPLES OF DISSIPATIVITY FOR FINITE AUTOMATA

Some examples are provided to demonstrate the dissipativity results for stability for finite automata. The first example demonstrates the invariant set result for the finite automata given in Fig. 4.

**Example 1.** This example has state set \( Q = \{ 0, 1, 2, 3 \} \), event set \( \Sigma = \{ a, b \} \), and equilibrium set \( Q_e = \{ 2, 3 \} \). A storage function can be defined where \( V(0) = V_0 \), \( V(1) = V_1 \), \( V(2) = V(3) = 0 \), and \( V_0 > V_1 > 0 \). A supply rate can be defined with \( \omega(0, a) = V_1 - V_0 \), \( \omega(1, b) = -V_1 \), and \( \omega = 0 \) for the other five transitions. It is quick to verify that \( \omega(q, e) = 0 \) for all \( q \in Q_e \) so the set \( Q_e \) is invariant. This can be verified by noting that when the finite automata reaches state 2, there isn’t a transition out of \( Q_e \) so this set is invariant.

![Fig. 4. This example shows a finite automata with equilibrium set \( \{ 2, 3 \} \).](image)

The application of this result is trivial for small finite automata as there are more direct methods of determining set invariance. However, it provides an automated method of verifying that a set is invariant for arbitrarily large finite automata. As the computational resources required to find a storage function and energy supply rate are relatively low, this may be a promising approach for automatically determining set invariance.

The following example shows how supervisory control can be used with dissipativity to ensure the limit cycle result.

**Example 2.** This example has state set \( Q = \{ 0, 1, 2, 3, 4, 5 \} \), event set \( \Sigma = \{ a, b, c \} \), final state set \( F = \{ 3 \} \), and equilibrium set \( Q_e = \{ 1, 2, 3 \} \). A storage function can be defined where \( V(0) = V_0 \), \( V(1) = V(2) = V(3) = 0 \), \( V(4) = V_4 > 0 \), and \( V(5) = V_5 > 0 \). A supply rate can be defined with \( \omega(q, a) = 0 \) for all \( q \), \( \omega(0, b) = -V_0 \), \( \omega(1, b) = V_4 \), \( \omega(3, b) = V_5 \), \( \omega(1, c) = V_6 \), and \( \omega(1, b) = V_4 - V_0 \). The dissipative rate \( \omega \) does not show invariance or a limit cycle for \( Q_e \) since \( \omega > 0 \) for some \( q \in Q_e \). A supervisor can be used to force the automaton to stay in the limit cycle \( Q_e \). This can be done by designing a supervisor to force \( \omega(q, e) = 0 \) for all \( q \in Q_e \). The transitions that violate this condition are \( \omega(1, b), \omega(3, b), \text{ and } \omega(1, c) \). When in the state 1 or 3 the supervisor must prohibit those transitions.

![Fig. 5. This example demonstrates how the concept of dissipativity can be used to verify limit cycles.](image)

VI. DISSIPATIVITY FOR GENERAL DES

The notion of dissipativity can be similarly defined for a general class of DES that allows for either infinite number of states, e.g. Petri nets, or infinite executions, e.g. Büchi [8] or Muller automata [10]. The background material on using a metric on a discrete state space will be applied to define a notion of dissipativity for this general class of DES.

**Definition 13.** Consider a discrete state space \( X \) with invariant set \( X_e \). A storage function \( V \) over \( X \) is a non-negative function \( (V(x) \geq 0 \text{ for all } x \in X) \) such that

- \( V(x) = 0 \text{ for } x \in X_e \) and
- \( V(x) > 0 \text{ for } x \in X \backslash X_e \).

\( V \) captures a notion of distance between a state in \( X \) and \( X_e \). A metric \( \rho \) defining a distance between two arbitrary states \( x_1, x_2 \in X \) is given by \( \rho(x_1, x_2) = |V(x_1) - V(x_2)| \). With this notion of energy storage, dissipativity can be defined with respect to an energy supply rate \( \omega_D(x, e) \). The notation, \( \forall x \xrightarrow{\omega_D} x' \), will be used to denote: \( \forall x \in X, \forall e \in g(x), \text{ and } \forall x' \in f(x, e) \).
Definition 14. Consider a DES \( D = \{ \mathcal{X}, \Sigma, f, g, E_v \} \) with invariant set \( \mathcal{X}_e \). D is dissipative with respect to energy supply rate \( \omega_D(x, e) \) if, \( \forall x \xrightarrow{e} x' \),

\[
V(x') \leq V(x) + \omega_D(x, e).
\] (4)

As with finite automata, both actual and desired DES behaviors may be captured with this notion of dissipativity. The following theorem connects this definition of dissipativity to Lyapunov stability for DES covered previously.

Theorem 3. Consider a dissipative DES \( D \) with invariant set \( \mathcal{X}_e \). \( D \) is Lyapunov stable with respect to \( \mathcal{X}_e \) if \( \omega_D(x, e) \leq 0 \), for all \( e \in E_v \).

Proof. Using the definition of dissipativity along with the condition that \( \omega_D(x, e) \leq 0 \), gives \( V(x') \leq V(x) \) which shows Lyapunov stability.

An additional condition is needed for asymptotic stability that captures the minimum number of transitions to move from the current state to an invariant set \( \mathcal{X}_e \). This notion is represented by function \( V_m(x) \) that can be defined constructively according to the following algorithm.

1) For all states \( x_e \in \mathcal{X}_e \), set \( V_m(x_e) = 0 \).
2) Initialize \( i = 1 \) and define \( \mathcal{X}_\text{unmarked} = \mathcal{X} - \mathcal{X}_e \).
3) For all \( x \in \mathcal{X}_\text{unmarked}, \) if \( \exists e \in g(x) \) s.t. \( x \in f(x, e) \) for some state \( x_e \in \mathcal{X}_e \), set \( V_m(x_l) = i \).
4) Remove these states from \( \mathcal{X}_\text{unmarked} \) and set \( i = i + 1 \).
5) Repeat (3) and (4) until all states are marked, i.e. \( \mathcal{X}_\text{unmarked} = \{ \emptyset \} \).

Assumption 2. A DES \( D \) with invariant set \( \mathcal{X}_e \) has a Lyapunov function \( V(x) \). This storage function decreases towards the invariant set \( \mathcal{X}_e \) if, \( \forall x \xrightarrow{e} x', V(x') < V(x) \) whenever \( V_m(x') < V_m(x) \) and \( V(x') = V(x) \) whenever \( V_m(x') = V_m(x) \).

Informally, this assumption captures the concept that the storage function \( V(x) \) decrease as transitions are approaching the invariant set \( \mathcal{X}_e \). An example of a storage function that satisfies the assumption is \( V_m(x) \) itself.

Theorem 4. Consider a dissipative DES \( D \) with invariant set \( \mathcal{X}_e \) such that \( \omega_D(x, e) \leq 0 \), for all \( e \in E_v \). If, in addition, Assumptions 1 and 2 hold then \( D \) is asymptotically stable.

Proof. From the previous theorem it is clear that \( D \) is Lyapunov stable. Assumption 1 along with Assumption 2 guarantees that executions of the DES do not get stuck in level sets outside of \( Q_e \).

The following example shows dissipativity for a DES modeled as a Büchi automaton and connects dissipativity to Lyapunov stability.

Example 3. Consider a Büchi automaton given by state set \( \mathcal{X} = \{ 0, 1, 2, 3 \} \), event set \( \Sigma = \{ a, b \} \), marked state set \( M = \{ 3 \} \), and invariant set \( \mathcal{X}_e = \{ 2, 3 \} \). A storage function can be defined where \( V(0) = V_0 \geq V_1, V(1) = V_2 \geq 0, \) and \( V(2) = V(3) = 0 \). A supply rate can be defined with \( \omega(0, a) = V_1 - V_0, \omega(0, b) = -V_0, \omega(1, a) = -V_1, \) and \( \omega(\cdot, \cdot) = 0 \) for all other transitions. This model can be shown to fit the general DES framework and have an energy storage function that meets Assumption 2. Since \( \omega(x, e) \leq 0 \) for all states, and \( \omega(x, e) = 0 \forall x \in \mathcal{X}_e \), this automaton is Lyapunov stable.

Fig. 6. This Büchi automaton is shown to be dissipative and Lyapunov stable.

VII. CONCLUSIONS

This paper presented two original notions of dissipativity, one for finite automata and one for general DES. These notions are defined based on events rather than time. This allows for energy-based analysis of DES with some promising results including stability and convergence of trajectories to invariant sets. Examples were provided to illustrate these notions and results. Future work in this direction would be to extend the definitions presented in this paper to timed automata. This would allow for a more unified definition of dissipativity that would allow for analysis of feedback interconnections of DES, dynamical systems, and hybrid systems.

REFERENCES