INT. J. CONTROL, 2003, VOL. 76, NO. 18, 1800–1808

Taylor & Francis
Taylor & Francis Group

Signed real measure of regular languages for discrete-event automata

ASOK RAY†* and SHASHI PHOHA‡

This paper presents the concept and formulation of a signed real measure of regular languages for analysis of discrete-event supervisory control systems. The measure is constructed based upon the principles of language theory and real analysis for quantitative evaluation and comparison of the controlled behaviour for discrete-event automata. The marked (i.e. accepted) states of finite-state automata are classified in different categories such that the event strings terminating at good and bad marked states have positive and negative measures, respectively. In this setting, a controlled language attempts to disable as many bad strings as possible and as few good strings as possible. Different supervisors may achieve this goal in different ways and generate a partially ordered set of controlled languages. The language measure creates a total ordering on the performance of the controlled languages, which provides a precise quantitative comparison of the controlled plant behaviour under different supervisors. Total variation of the language measure serves as a metric for the space of sublanguages of the regular language.

1. Introduction

An important paradigm for discrete event supervisory (DES) control was originally proposed by Ramadge and Wonham (1987) and subsequently extended by other researchers (for example, see the October 2000 issue of Part B of *IEEE Transactions on Systems, Man, and Cybernetics*). The Supervisory Control Theory (SCT) partitions the discrete-event behaviour of a physical plant into legal and illegal categories. The legal behaviour of the plant is modelled by a deterministic finite-state automaton, abbreviated as DFSA in the sequel. The DFSA model is equivalent to a regular language. Then, SCT synthesizes a DES controller as another language that guarantees restricted legal behaviour of the controlled plant based on the desired specifications. Instead of continuous numerical data, DES controllers process event strings to disable certain controllable events in the physical plant. The algorithms for DES control synthesis have evolved based on the automata theory and formal languages in the discipline of Computer Science.

The controlled behaviour of a given DFSA, also referred to as the plant, under different supervisors could vary, as they are designed based on different control specifications. As such the respective controlled sublanguages of the automaton form a partially ordered set that is not necessarily totally ordered. Since the literature on DES control does not apparently provide a language measure, it may not be possible to quantitatively evaluate the performance of a DES controller. Therefore, it is necessary to formulate a mathematically rigorous concept of language measure(s) to quantify performance of individual supervisors such that the measures of controlled plant behaviour, described by a partially ordered set of controlled sublanguages, can be structured to form a totally ordered set. From this perspective, the goal of the paper is to construct a signed real measure that can be assigned to any sublanguage of the uncontrolled regular language of the plant to achieve the following objective (Ray and Phoha 2002, Wang and Ray 2002):

> Given that the relation $\subseteq$ induces a partial ordering on a set of controlled sublanguages $\{L_k\}$ of a regular plant language $\mathcal{L}$, the signed real measure μ induces a total ordering $\leq$ on $\{\mu(L_k)\}$. That is, the range of the set function μ is totally ordered while its domain could be only partially ordered.

2. Concept of the language measure

Let $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$ be a trim (i.e. accessible and co-accessible) DFSA that represents the discrete-event dynamics of a physical plant (Ramadge and Wonham 1987), where $Q = \{q_1, q_2, \ldots, q_n\}$ is the set of states with q_i being the initial state; $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ is the alphabet of events; $Q_m \subseteq Q$ is the (non-empty) set of marked (i.e. accepted) states; δ: $Q \times \Sigma \to Q$ is the (possibly partial) function of state transitions and δ^*: $Q \times \Sigma^* \to Q$ is an extension of δ. The (countable) set Σ^* is the Kleene closure of Σ, i.e. the set of all (finite-length) strings made of the events belonging to Σ including the empty string ε that is viewed as the identity element of the monoid Σ^* under the operation of string concatenation, i.e. $\varepsilon s = s = s\varepsilon \ \forall s \in \Sigma^*$.

Since δ is allowed to be a partial function, the regular language $\mathcal{L}(\mathcal{G}_i)$ generated by the DFSA $\mathcal{G}_i$ is

Received 1 December 2002. Revised and accepted 13 October 2003.

* Author for correspondence. e-mail: axr2@psu.edu
† Mechanical Engineering Department, The Pennsylvania State University, University Park, PA 16802, USA.
‡ Applied Research Laboratory, The Pennsylvania State University, University Park, PA 16802, USA.

International Journal of Control ISSN 0020–7179 print/ISSN 1366–5820 online
http://www.tandf.co.uk/journals
DOI: 10.1080/00207170310001635392

given as: $\mathcal{L}(\mathcal{G}_i) \subseteq \Sigma^*$, and $\mathcal{L}(\mathcal{G}_i) = \Sigma^*$ iff δ: $Q \times \Sigma \to Q$ is a total function. Therefore, if δ is a partial function, the set of states can be augmented with an additional non-marked dead-lock state q_{n+1}, called the dump state, such that the partial function δ can be extended to a total function δ_{ext}: $(Q \cup \{q_{n+1}\}) \times \Sigma \to (Q \cup \{q_{n+1}\})$.

Definition 1: A σ-algebra $\mathcal{M}$ of a language $\mathcal{L} \subseteq \Sigma^*$ is a collection of subsets of $\mathcal{L}$ which satisfies the following three conditions:

(i) $\mathcal{L} \in \mathcal{M}$;
(ii) If $L \in \mathcal{M}$, then $(\mathcal{L} - L) \in \mathcal{M}$;
(iii) $\bigcup_{k=1}^{\infty} L_k \in \mathcal{M}$ if $L_k \in \mathcal{M}$ $\forall k$.

Definition 2: Let $\mathcal{M}$ be a σ-algebra. An at most countable collection $\{L_k\}$ of members of $\mathcal{M}$ is a partition of a member $L \in \mathcal{M}$ if $L = \bigcup_k L_k$ and $L_i \cap L_j = \emptyset$ $\forall i = j$.

Definition 3: Given a σ-algebra $\mathcal{M}$ of a language $\mathcal{L}$, the set function μ: $\mathcal{M} \to \Re \equiv (-\infty, \infty)$, is called a signed real measure if the following two conditions are satisfied:

(i) $\mu(\emptyset) = 0$;
(ii) $\mu\left(\bigcup_{k=1}^{\infty} L_k\right) = \sum_{k=1}^{\infty} \mu(L_k)$ for every partition $\{L_k\}$ of any member $L \in \mathcal{M}$.

Note that, unlike a positive measure (e.g. the Lebesgue measure), μ is finite (but not necessarily bounded) such that the series in part (ii) of Definition 3 converges absolutely in $\Re$. The result is independent of any permutation of the terms under union.

Definition 4: Total variation measure $|\mu|$ on a σ-algebra $\mathcal{M}$ is defined as $|\mu|(L) = \sup \sum_k |\mu(L_k)|$ $\forall L \subseteq \mathcal{M}$ where the supremum is taken over all partitions $\{L_k\}$ of L.

Definition 5: Relative to the signed real measure μ, a sublanguage $L \in \mathcal{M}$ is defined to be:

(i) null, denoted as $L = 0$, if $\mu(L \cap J) = 0$ $\forall J \in \mathcal{M}$;
(ii) positive, denoted as $L > 0$, if $\mu(L \cap J) > 0$ $\forall J \in \mathcal{M}$;
(iii) negative, denoted as $L < 0$, if $\mu(L \cap J) < 0$ $\forall J \in \mathcal{M}$.

Proposition 1: *Total variation measure $|\mu|$ of any regular language $\mathcal{L}$ is non-negative and finite, i.e. $|\mu|(\mathcal{L}) \in [0, \infty)$. Hence, $|\mu|(L) \in [0, \infty)$ $\forall L \in \mathcal{M}$.*

Proof of Proposition 1: The proof follows from standard theorems on complex measures (Rudin 1988). □

Proposition 2: *Every sublanguage $L \in \mathcal{M}$ can be partitioned as: $L = L^0 \cup L^+ \cup L^-$ where mutually exclusive sublanguages L^0, L^+ and L^- are null, positive, and negative, respectively, relative to a signed real measure μ.*

Proof of Proposition 2: The proof is based on the Hahn decomposition theorem (Rudin 1988). □

3. Formulation of the language measure

For a given DFSA $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$, we now construct a σ-algebra $\mathcal{M}$ as the power set $2^{\mathcal{L}(\mathcal{G}_i)}$ of the regular language $\mathcal{L}(\mathcal{G}_i)$.

Proposition 3: *Total variation measure $|\mu|$ on the σ-algebra $2^{\mathcal{L}(\mathcal{G}_i)}$ is*

$$|\mu|(K) = \sum_{s \in K} |\mu(\{s\})| \quad \forall K \subseteq \mathcal{L}(\mathcal{G}_i).$$

Proof of Proposition 3: The proof follows from Definition 4 based on the facts that $\mathcal{L}(\mathcal{G}_i) \subseteq \Sigma^*$ is at most countable and that every singleton set of a legal string belongs to $2^{\mathcal{L}(\mathcal{G}_i)}$. □

The marked (i.e. accepted) language $\mathcal{L}_m(\mathcal{G}_i)$ of a trim DFSA $\mathcal{G}_i$ has the properties $\emptyset \subset \mathcal{L}_m(\mathcal{G}_i) \subseteq \mathcal{L}(\mathcal{G}_i)$ and $\mathcal{L}_m(\mathcal{G}_i) = \mathcal{L}(\mathcal{G}_i)$ iff $Q_m = Q$. Let the marked states be designated as $Q_m \equiv \{q_{m_1}, q_{m_2}, \ldots, q_{m_\ell}\} \subseteq Q$ where $q_{m_k} = q_j$ for some $j \in \{1, 2, \ldots, n\}$.

Definition 6: For a state $q \in Q$ of a given DFSA $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$, the regular language $L(q_i, q)$ is defined to be the set of all strings that terminate at q starting from the initial state q_i. Equivalently, $L(q, q_i)$ is the sublanguage of all legal event strings terminating at q starting from q_i.

The Myhill–Nerode theorem is now applied to construct the following state-based partitions (Martin 1997, Hopcroft *et al.* 2001)

$$\mathcal{L}(\mathcal{G}_i) = \bigcup_{q \in Q} L(q_i, q) \quad \text{and} \quad \mathcal{L}_m(\mathcal{G}_i) = \bigcup_{q \in Q_m} L(q_i, q)$$

where the sublanguage $L(q_i, q_k)$ of all (legal) event strings starting at the initial state q_i is uniquely labelled by the terminal state q_k $\forall k \in \{1, 2, \ldots, n\}$.

In order to obtain a quantitative measure of the marked language $\mathcal{L}_m(\mathcal{G}_i)$, the set of marked states is partitioned as: $Q_m = Q_m^+ \cup Q_m^-$ and $Q_m^+ \cap Q_m^- = \emptyset$. The positive set Q_m^+ contains *good* marked states that we desire to reach, and the negative set Q_m^- contains *bad* marked states that we want to avoid, although it may not always be possible to completely avoid the bad states while attempting to reach the good states. In general, the marked language $\mathcal{L}_m(\mathcal{G}_i)$ consists of both good and bad event strings that, starting from the initial stage q_i, respectively lead to Q_m^+ and Q_m^-. Any event string belonging to the language $\mathcal{L}(\mathcal{G}_i) - \mathcal{L}_m(\mathcal{G}_i)$ leads to one of the non-marked states belonging to $(Q - Q_m)$ and does not contain any one of the good or bad strings.

The objective is to construct a performance measure of sublanguages of a regular language for discrete-event control and to define quantitative metrics of the controlled (i.e. supervised) plant performance. To this end, the following definitions are introduced to construct a signed real measure of sublanguages of the regular language. This measure is not restricted to regular sublanguages of the original regular language based on which the measure is constructed.

In view of Definition 5, we proceed to construct a signed real measure $\mu : 2^{\mathcal{L}(\mathcal{G}_i)} \to \Re \equiv (-\infty, \infty)$ to allow state-based decomposition of $\mathcal{L}(\mathcal{G}_i)$ into null, positive, and negative sublanguages such that:

(i) $\mu(L(q_i, q)) = 0 \ \forall q \notin Q_m$, i.e. a (legal) event string starting at the initial state q_i and terminating on any non-marked state has zero measure;

(ii) partitioning of Q_m into Q_m^+ and Q_m^- yields the properties $\mu(L(q_i, q)) > 0 \quad \forall q \in Q_m^+$ and $\mu(L(q, q_i)) < 0 \ \forall q \in Q_m^-$, which is in agreement with Proposition 2 in the sense that $L^0 = \bigcup_{q \notin Qm} L(q_i, q)$, $L_m^+ = \bigcup_{q \in Q_m^+} L(q_i, q)$ and $L_m^- = \cup_{q \in Q_m^-} L(q_i, q)$.

Partitioning the marked language $\mathcal{L}_m(\mathcal{G}_i)$ into a positive language L_m^+ and a negative language L_m^- is equivalent to partitioning Q_m into the positive set Q_m^+ and the negative set Q_m^-. Each state belonging to Q_m^+ is characterized by a positive weight and each state belonging to Q_m^- by a negative weight. These weights are chosen by the designer based on the perception of each marked state's role in the system performance.

Definition 7: The characteristic function χ: $Q \to [-1, 1]$ that assigns a signed real weight to event strings of $\mathcal{L}(\mathcal{G}_i)$ based on their terminal states is defined as

$$\chi(q) \in \begin{cases} [-1, 0) & \text{if } q \in Q_m^- \\ \{0\} & \text{if } q \notin Q_m \\ (0, 1] & \text{if } q \in Q_m^+ \end{cases}$$

For any accessible DFSA $\mathcal{G}_i$, the sublanguage $L(q_i, q_k)$ is a non-empty language $\forall k \in \{1, 2, \ldots, n\}$. In that case, the implication of the characteristic function is that a string belonging to a sublanguage $L(q_i, q_k)$, which is labelled by the terminal state q_k, has a zero measure if q_k is not a marked state, a positive measure if q_k is a good marked state and a negative measure if q_k is a bad marked state.

We now introduce the cost of event strings belonging to $\mathcal{L}(\mathcal{G}_i)$. The cost assignment procedure is conceptually similar to that for state-based conditional probability to events of a string. Since the consecutive events in a string may not be statistically independent, it is necessary to find the joint probability mass functions of arbitrarily large order. This makes the probability space of Σ^* ever expanding as there is no finite upper bound on the length of strings in Σ^*. This problem is circumvented by using the state transition function δ of the DFSA $\mathcal{G}_i$.

Definition 8: The event cost generated at a DFSA state is defined as $\tilde{\pi} : \Sigma^* \times Q \to [0, 1)$ such that $\forall q_j \in Q$, $\forall \sigma$, $\sigma_k \in \Sigma$, $\forall s \in \Sigma^*$:

- $\tilde{\pi}[\sigma_k \mid q_j] \equiv \tilde{\pi}_{jk} \in [0, 1)$: $\quad \sum_{j=1}^{m} \tilde{\pi}_{jk} < 1$;
- $\tilde{\pi}[\sigma \mid q_j] = 0$ if $\delta(q_j, \sigma)$ is undefined; $\tilde{\pi}[\varepsilon \mid q_k] = 1$:
- $\tilde{\pi}[\sigma s \mid q_j] = \tilde{\pi}[\sigma \mid q_j]\tilde{\pi}[s \mid \delta(q_j, \sigma)]$.

The event cost function $\tilde{\pi}$ for an event string $s \in L(q_k, q_i)$ starting from the initial state q_i and terminating at q_k is obtained as the product of respective costs conditioned on the state from which the events are generated. This is conceptually similar to having a product of conditional probabilities. For example, if $s = \sigma_j\sigma_k\sigma_\ell$ for a DFSA $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$, then $\tilde{\pi}(s \mid q_i) = \tilde{\pi}_{ij}\tilde{\pi}_{ak}\tilde{\pi}_{b\ell}$ where the state transition function δ defines the (Markov) states $q_a = \delta(q_i, \sigma_j)$ and $q_b = \delta(q_a, \sigma_k)$.

Definition 9: The signed measure μ of every singleton set member of $2^{\mathcal{L}(\mathcal{G}_i)}$ is defined as $\mu(\{s\}) \equiv \tilde{\pi}(s \mid q_i)\chi(q)$, where $s \in L(q_i, q)$.

Definition 10: Given a DFSA $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$, the cost v of a sublanguage $K \subseteq \mathcal{L}(\mathcal{G}_i)$ is defined as the sum of the event cost $\tilde{\pi}$ of individual strings belonging to K

$$v(K) = \sum_{s \in K} \tilde{\pi}(s \mid q_i)$$

and the signed measure of $K \subseteq \mathcal{L}(\mathcal{G}_i)$ is defined as the sum of the signed measures of equivalence classes of K as

$$\mu(K) = \sum_j v(L(q_i, q_j) \cap K)\chi(q_j)$$

Definitions 7, 8 and 10 imply that, for an event string s belonging to an accessible language $\mathcal{L}(\mathcal{G}_i)$

$$\mu(\{s\}) \begin{cases} = 0 & \text{if } s \in L(q_i, q) \text{ for } q \notin Q_m \\ > 0 & \text{if } s \in L(q_i, q) \text{ for } q \in Q_m^+ \\ < 0 & \text{if } s \in L(q_i, q) \text{ for } q \in Q_m^- \end{cases}$$

Therefore, the signed measure μ can be assigned to each event string belonging to $s \in \mathcal{L}(\mathcal{G}_i)$ that is partitioned by the sublanguages $L(q_k, q_i)$, $k \in \{1, 2, \ldots, n\}$ in terms of the non-negative cost $\tilde{\pi}$ and the signed characteristic function χ.

In view of Proposition 3, Definition 10 assigns a total variation measure $|\mu|$ to each event string $s \in \mathcal{L}(\mathcal{G}_i)$ and hence to every sublanguage $K \subseteq \mathcal{L}(\mathcal{G}_i)$ as

$$|\mu|(K) = \sum_j v(L(q_i, q_j) \cap K)\chi(q_j)$$

4. Convergence of the language measure

The previous section formulated the real signed measure μ based on Definition 10. This section establishes convergence of the measure μ in view of Proposition 1 and Proposition 3 by showing that $|\mu|(L(q_k, q_i)) < \infty$ $\forall q_k, q_i \in Q$, which is equivalent to $\mu(\mathcal{L}_m(\mathcal{G}_i)) = \mu(\mathcal{L}(\mathcal{G}_i)) \leq |\mu|(\mathcal{L}(\mathcal{G}_i)) < \infty$.

The following definitions and propositions are introduced to compute $\mu(L)$ and $|\mu|(L)$ for any $L \subseteq \mathcal{L}(\mathcal{G}_i)$ and establish the convergence.

Definition 11: Given $q_i, q_k \in Q$, a non-empty string p of events (i.e. $p \neq \varepsilon$) starting from q_i and terminating at q_k is called a path. A path p from q_i to q_k is said to pass through q_j if $\exists$ strings $s \neq \varepsilon$ and $t \neq \varepsilon$ such that $p = st$, $\delta^*(q_i, s) = q_j$ and $\delta^*(q_j, t) = q_k$ where δ^*: $Q \times \Sigma^* \to Q$.

Definition 12: A path language p_{ik}^j is defined to be the set of all paths from q_i to q_k, which do not pass through any state q_ℓ for $\ell > j$. The path language p_{ik} is defined to be the set of all paths from q_i to q_k. Thus, the language $L(q_i, q_k)$ is obtained in terms of the path language p_{ik} as

$$L(q_i, q_k) = \begin{cases} p_{ii} \cup \{\varepsilon\} & \text{if } k = i \\ p_{ik} & \text{if } k \neq i \end{cases}$$

$$\Rightarrow v(L(q_i, q_k)) = \begin{cases} v(p_{ii}) + 1 & \text{if } k = i \\ v(p_{ik}) & \text{if } k \neq i \end{cases}$$

Based on the above definitions, we present the following propositions and lemmas to quantify the language measure μ.

Proposition 4: *Let $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$ be a DFSA with $|Q| = n$. Then, $p_{ik}^j = p_{ik}$ $\forall j \geq n$.*

Proof of Proposition 4: The proof relies on the fact that no string passes through a state numbered higher than n. □

Proposition 5: *Every path language is regular for a DFSA $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$.*

Proof of Proposition 5: Since p_{ik}^0 is a finite language and hence regular, it follows from the proof of Kleene's theorem (Martin 1997, p. 123) by the induction hypothesis that p_{ik}^{j+1} is regular if p_{ik}^j is regular for all $1 \leq j \leq n$. □

Proposition 6: *Let u and v be two known regular expressions and let r be an unknown regular expression that is governed by the implicit equation $r = ur + v$. Then, $\exists$ solutions $r = u^*v + \theta$ where θ satisfies the condition $u\theta + v = \theta + v$ and the solution $r = u^*v$ is unique if $\varepsilon \notin u$.*

Proof of Proposition 6: Existence is established by substituting $r = u^*v + \theta$ in $r = ur + v$ and then using the identity $u\theta + v = \theta + v$

$$\begin{aligned} ur + v &= u(u^*v + \theta) + v = uu^*v + (u\theta + v) \\ &= uu^*v + (\theta + v) = (uu^* + \varepsilon)v + \theta = u^*v + \theta = r \end{aligned}$$

If $\varepsilon \notin u$, then $u^* = u^*u + \varepsilon$ is a partition of u^*, which implies $u^*r = u^*ur + r$ is a partition of u^*r. It follows from $r = ur + v$ that $u^*r = u^*ur + u^*v \Rightarrow r \subseteq u^*v$.

Suppose $r \subset u^*v$. Partitioning of u^*v yields $u^*v = r + \phi$ for some $\varphi \neq \emptyset$. If follows from $r = ur + v$ that $r + \phi = u^*v = uu^*v + v$. Therefore, $u(r + \varphi) + v = r + u\varphi$ $\Rightarrow$ $\varphi \subseteq u\varphi$ which is a contradiction because $\varepsilon \notin u$. Hence, the solution $r = u^*v$ is unique if $\varepsilon \notin u$. An alternative proof is given by Drobot (1989). □

Proposition 7: *For a given DFSA $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$, the following recursive relation holds for $0 \leq \ell \leq n - 1$*

$$p_{ik}^0 = \{\sigma \in \Sigma : \delta(q_i, \sigma) = q_k\}$$

and

$$p_{ik}^{\ell+1} = p_{ik}^\ell \cup p_{i,\ell+1}^\ell (p_{\ell+1,\ell+1}^\ell)^* p_{\ell+1,k}^\ell$$

Proof of Proposition 7: Since the states are numbered from 1 to n in increasing order, $p_{ik}^0 + \{\sigma \in \Sigma : \delta(q_i, \sigma) = q_k\}$ follows directly from the state transition map $\delta : Q \times \Sigma \to Q$ and Definition 12. Given $p_{ik}^\ell \subseteq p_{ik}^{\ell+1}$, let us consider the set $p_{ik}^{\ell+1} - p_{ik}^\ell$ in which each string passes through $q_{\ell+1}$in the path from q_i to q_k and no string must pass through q_ℓ for $\ell > (j+1)$. Then, it follows that $p_{ik}^{\ell+1} - p_{ik}^\ell = p_{i,\ell+1}^\ell p_{\ell+1,k}^{\ell+1}$ where $p_{\ell+1,k}^{\ell+1}$ can be expanded as $p_{\ell+1,k}^{\ell+1} = (p_{\ell+1,\ell+1}^\ell p_{\ell+1,k}^{\ell+1}) \cup p_{\ell+1,k}^\ell$ that has a unique solution

$$p_{\ell+1,k}^{\ell+1} = (p_{\ell+1,\ell+1}^\ell)^* p_{\ell+1,k}^\ell$$

by Proposition 6 because $\varepsilon \notin p_{\ell+1,\ell+1}^\ell$ based on Definition 11. Therefore

$$p_{ik}^{\ell+1} = p_{ik}^\ell \cup p_{i,\ell+1}^\ell (p_{\ell+1,\ell+1}^\ell)^* p_{\ell+1,k}^\ell$$

An alternative proof is given in Martin (1997, p. 124). □

Proposition 8: *The following recursive relations hold for $0 \leq \ell \leq n - 1$*

$$v(p_{ik}^{\ell+1}) = v(p_{ik}^\ell) + \frac{v(p_{i,\ell+1}^\ell) v(p_{\ell+1,k}^\ell)}{1 - v(p_{\ell+1,\ell+1}^\ell)} \in [0, \infty)$$

Proof of Proposition 8: We need three lemmas to prove the proposition.

Lemma 1:

$$v\left((p_{kk}^0)^* \bigcup_{j \neq k} p_{kj}^0\right) \in [0, 1)$$

Proof of Lemma 1: Following Definition 8, $v(p_{kk}^0) \in [0, 1)$. Therefore, by convergence of a geometric series

$$v\left((p_{kk}^0)^* \bigcup_{j \neq k} p_{kj}^0\right) = \frac{\sum_{j \neq k} v(p_{kj}^0)}{1 - v(p_{kk}^0)} \in [0, 1)$$

because

$$\sum_j v(p_{kj}^0) < 1 \quad \Rightarrow \quad \sum_{j \neq k} v(p_{kj}^0) < 1 - v(p_{kk}^0)$$

□

Lemma 2: $v(p^j_{j+1,j+1}) \in [0, 1)$.

Proof of Lemma 2: The path $p^j_{j+1,j+1}$ may contain at most j loops, one around the states $q_1, q_2, \ldots, q_j$. If the path $p^j_{j+1,j+1}$ does not contain any loop, then $v(p^j_{j+1,j+1}) \in [0, 1)$ because it is a product of $\tilde{\pi}_{ik}$'s, each of which is a non-negative fraction. Next suppose there is loop around q_ℓ that does not contain any other loop; this loop must be followed by one or more events σ_k generated at q_ℓ and leading to some other states q_m where $m \in \{1, \ldots, j+1\}$ and $m \neq \ell$. By Lemma 1, $v(p^j_{j+1,j+1}) \in [0, 1)$. Proof follows by starting from the innermost loop and ending with all loops at q_j. □

Lemma 3: $v((p^j_{j+1,j+1})^*) \in [1, \infty)$.

Proof of Lemma 3: Since $v(p^j_{j+1,j+1}) \in [0, 1)$ from Lemma 2

$$v\Big((p^j_{j+1,j+1})^*\Big) = \frac{1}{1 - v(p^j_{j+1,j+1})} \in [1, \infty)$$

□

Now we proceed to prove Proposition 8. Since the languages $p^\ell_{j,\ell+1}, p^\ell_{\ell+1,\ell+1}$ and $p^\ell_{\ell+1,k}$ are mutually disjoint, it follows from Proposition 7 that

$$v(p^{\ell+1}_{ik}) = v(p^\ell_{ik}) + v(p^\ell_{i,\ell+1})v((p^\ell_{\ell+1,\ell+1})^*)v(p^\ell_{\ell+1,k})$$

The proof follows by applying Lemmas 1, 2 and 3 to the above expression. □

Now we present the main result that the cost of any path language is finite.

Proposition 9: *For a given DFSA $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$, the measure and total variation of every sublanguage $K \subseteq \mathcal{L}(\mathcal{G}_i)$ are finite. Specifically, $|\mu(K)| \leq |\mu|(K) < \infty$.*

Proof of Proposition 9: The proof follows from Proposition 3 and Lemma 3. □

To facilitate numerical computation of μ, we introduce the state transition cost π that can be used in place of the event cost $\tilde{\pi}$ in Definition 8.

Definition 13: The state transition cost of the DFSA is defined as a function π: $Q \times Q \to [0, 1)$ such that $\forall q_j, q_k \in Q$

$$\pi(q_k, q_j) = v(p^0_{jk}) = \sum_{\sigma \in \Sigma: \delta(q_j, \sigma) = qk} \tilde{\pi}(\sigma, q_j) = \pi_{jk}$$

and $\pi_{jk} = 0$ if $\{\sigma \in \Sigma$: $\delta(q_j, \sigma)\} = \emptyset$. The state transition cost matrix, denoted as Π-matrix: is defined as

$$\Pi = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1n} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2n} \\ \vdots & & \ddots & \vdots \\ \pi_{n1} & \pi_{n2} & \cdots & \pi_{nn} \end{bmatrix}$$

An algorithm to compute the language measure μ and total variation $|\mu|$ based on the recursive relation in Proposition 8 is presented below:

Step 1. For $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$, obtain the event cost matrix $[\tilde{\pi}_{ij}]$ and the characteristic vector χ.

Step 2. Generate the state transition cost matrix $[\pi_{ij}]$ from the event cost matrix $[\tilde{\pi}_{ij}]$ and the state transition function δ following Definition 13.

Step 3. Execute the following nested for loops:

```
Initialize v^0_jk = π_jk   ∀j, k ∈ {1, ···, n}
for ℓ = 0 to n − 1
  for j = 1 to n
    for k = 1 to n
      v(p^{ℓ+1}_jk) ← v(p^ℓ_jk) + v(p^ℓ_{j,ℓ+1}) v(p^ℓ_{ℓ+1,k}) / (1 − v(p^ℓ_{ℓ+1,ℓ+1}))
    end
  end
end
```

Step 4: Generate $v(L(q_k, q_j))$ from $v(p_{jk})$ using Definition 12.

Step 5: $\mu_i \leftarrow \sum_{q \in Q_m} v(L(q, q_i)\chi(q)$

The above algorithm can be generated in polynomial time. Specifically, the algorithm for numerically solving $v(p_{ik})$ requires three for-loops and hence, for a n-state automaton, the computation time is in the order of n^3.

5. Usage of the language measure

This section outlines usage of the language measure μ for analysis and synthesis of discrete-event supervisory control system. The following two subsections present construction of metric spaces of formal languages and construction of discrete-event supervisor control laws.

5.1. *Vector space of formal languages*

This subsection makes use of the language measure to construct a metric space of sublanguages of a regular language representing the DFSA $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$ where the total variation measure $|\mu|$ induces a metric on this space to quantify the distance function between any two sublanguages of $\mathcal{L}(\mathcal{G}_i)$.

Proposition 10: *Let $\mathcal{L}(\mathcal{G}_i)$ be the language of a DFSA $\mathcal{G}_i \equiv \langle Q, \Sigma, \delta, q_i, Q_m \rangle$. Let the binary operation of symmetric difference (i.e. exclusive-OR) $\oplus$: $2^{\mathcal{L}(\mathcal{G}_i)} \times 2^{\mathcal{L}(\mathcal{G}_i)} \to 2^{\mathcal{L}(\mathcal{G}_i)}$ be defined as*

$$(K_1 \oplus K_2) \equiv (K_1 \cup K_2) - (K_1 \cap K_2)$$

$\forall K_1, K_2 \subseteq \mathcal{L}(\mathcal{G}_i)$. Then, $\langle 2^{\mathcal{L}(\mathcal{G}_i)}, \oplus \rangle$ is a vector space over the Galois field $GF(2)$.

Proof of Proposition 10: We note that $\langle 2^{\mathcal{L}(\mathcal{G}_i)}, \oplus \rangle$ is an abelian group where $\emptyset$ is the zero element of the group and the unique inverse of every element $K \in 2^{\mathcal{L}(\mathcal{G}_i)}$ is

K itself because $K_1 \oplus K_2 = \emptyset$ if and only if $K_1 = K_2$. The associativity and distributivity of the space follows by defining the scalar multiplication of vectors as: $0 \otimes K \equiv \emptyset$ and $1 \otimes K \equiv K$. □

The collection of singleton languages made from each element of $\mathcal{L}(\mathcal{G}_i)$ forms a basis set of the vector space $\langle 2^{\mathcal{L}(\mathcal{G}_i)}, \oplus \rangle$ over $GF(2)$. Thus, $\mathcal{L}(\mathcal{G}_i)$ is bijective to any basis set of $\langle 2^{\mathcal{L}(\mathcal{G}_i)}, \oplus \rangle$.

Definition 14: Let $\mathcal{L}(\mathcal{G}_i)$ be the regular language for the DFSA $\mathcal{G}_i$. The distance function $d : 2^{\mathcal{L}(\mathcal{G}_i)} \times 2^{\mathcal{L}(\mathcal{G}_i)} \to [0, \infty)$ is defined in terms of the total variation measure $|\mu|$ as

$$\forall K_1, K_2 \subseteq \mathcal{L}(\mathcal{G}_i)$$

$$d(K_1, K_2) = |\mu|(K_1 \oplus K_2) = |\mu|((K_1 \cup K_2) - (K_1 \cap K_2))$$

The above distance function $d(\cdot, \cdot)$ quantifies the difference between two supervisors relative to the controlled performance of the DFSA plant.

Proposition 11: *The distance function* $d : 2^{\mathcal{L}(\mathcal{G}_i)} \times 2^{\mathcal{L}(\mathcal{G}_i)} \to [0, \infty)$ *is a pseudo-metric on the space* $2^{\mathcal{L}(\mathcal{G}_i)}$.

Proof of Proposition 11: We note that $\forall K_1, K_2 \in 2^{\mathcal{L}(\mathcal{G}_i)}$, $d(K_1, K_2) = |\mu|(K_1 \oplus K_2) \geq 0$ and $d(K_1, K_2) = d(K_2, K_1)$. The remaining property of triangular inequality follows from the inequality $|\mu|(K_1 \oplus K_2) \leq |\mu|(K_1) + |\mu|(K_2)$ based on the facts that $(K_1 \oplus K_2) \subseteq (K_1 \cup K_2)$ and $|\mu|(K_1) \leq |\mu|(K_2)$ $\forall K_1 \subset K_2$, following Definition 3. □

The pseudo-metric $|\mu| : 2^{\mathcal{L}(\mathcal{G}_i)} \to [0, \infty)$ can be converted to a metric of the space $\langle 2^{\mathcal{L}(\mathcal{G}_i)}, \oplus \rangle$ by clustering all languages that have zero total variation measure as the null equivalence class $\mathcal{N} \equiv \{K \in 2^{\mathcal{L}(\mathcal{G}_i)} : |\mu|(K) = 0\}$. This procedure is conceptually similar to what is done for defining norms in the L_p spaces (Rudin 1988). In that case, $\mathcal{N}$ contains all sublanguages of $\mathcal{L}(\mathcal{G}_i)$, which terminate on non-marked states starting from the initial state, i.e. $\mathcal{N} = \emptyset \cup (\bigcup_{q \notin Q_m} L(q_i, q))$. In the sequel, $|\mu|(\cdot)$ is referred to as a metric of the space $2^{\mathcal{L}(\mathcal{G}_i)}$. The metric $|\mu|(\cdot)$ can be generated from $d(\cdot, \cdot)$ as:

$$|\mu|(K) = d(K, J) \quad \forall K \in 2^{\mathcal{L}(\mathcal{G}_i)} \quad \forall J \in \mathcal{N}$$

Unlike the norms on vector spaces defined over infinite fields, the metric $|\mu|(\cdot)$ for the vector space, $\langle 2^{\mathcal{L}(\mathcal{G}_i)}, \oplus \rangle$ over $GF(2)$, is not a functional.

The metric space $\langle 2^{\mathcal{L}(\mathcal{G}_i)}, d \rangle$ can be completed by augmenting the state set Q with the additional dump state q_{n+1}. In that case, the state transition function δ becomes a total function with $\chi(q_{n+1}) = 0$ following Definitions 7 and 9. As the domain of the language measure μ is extended from $2^{\mathcal{L}(\mathcal{G}_i)}$ to 2^{Σ^*}, non-zero values of μ remain unchanged and the null equivalence class is expanded as $\mathcal{N} = \{K \in 2^{\Sigma^*} : |\mu|(K) = 0\}$.

5.2. *Optimal control of regular languages*

While the recursive solution of the language measure in § 4 is very useful for construction of executable codes in real time, a closed form solution is more amenable for analysis and synthesis of decision and control algorithms. Wang and Ray (2002) have shown that the measure $\mu_i \equiv \mu(\mathcal{L}(\mathcal{G}_i))$ of the language $\mathcal{L}(\mathcal{G}_i)$, with the initial state q_i, can be expressed as

$$\mu_i = \sum_j \pi_{ij} \mu_j + \chi_i$$

where $\chi_i \equiv \chi(q_i)$. Equivalently, in vector notation

$$\mu = \Pi \mu + \bar{\chi}$$

where the measure vector $\bar{\mu} \equiv [\mu_1 \; \mu_2 \; \cdots \; \mu_n]^T$ and the characteristic vector $\bar{\chi} \equiv [\chi_1 \; \chi_2 \; \cdots \; \chi_n]^T$. Following Definitions 8 and 13, there exists $\theta \in (0, 1)$ such that the induced infinity norm $\|\Pi\|_\infty = \max_i \Sigma_j \pi_{ij} = 1 - \theta$ and the matrix operator $[I - \Pi]$ is invertible. This implies that the inverse $[I - \Pi]^{-1}$ is a bounded operator with its induced infinity norm $\|[I - \Pi]^{-1}\|_\infty \leq \theta^{-1}$(Naylor and Sell 1982, p. 431). Therefore, the language measure vector $\bar{\mu}$ is uniquely determined in the closed form as

$$\bar{\mu} = [I - \Pi]^{-1} \bar{\chi}$$

Following the above closed form expression of the language measure, Fu *et al.* (2002, 2003 b) have established a theory for supervisory control of regular languages by selectively disabling controllable events so that the resulting optimal policy can be realized as a controllable supervisor. The plant model is first modified to satisfy the specified operational constraints, if any. Then, starting with the (regular) language of the unsupervised plant, the optimal policy maximizes the performance of the controlled sublanguage of the supervised plant without any further constraints. The performance index of the optimal policy is a signed real measure of the supervised sublanguage which is expressed in terms of the modified state transition cost matrix (due to disabling of certain controllable events) and the characteristic vector $\bar{\chi}$.

Let $\mathcal{S} \equiv \{S^0, S^1, \ldots, S^N\}$ be a set of DES control policies for the (unsupervised) plant automaton G where S^0 is the null controller (i.e. no event is disabled) implying that $L(S^0/G) = L(G)$. Therefore, the controller cost matrix $\Pi(S^0) = \Pi^0$ that is the Π-matrix of the open loop plant automaton G. For a supervisor S^k, $k \in \{1, 2, \ldots, N\}$, the control policy is required to selectively disable certain controllable events so that the following (elementwise) inequality holds: $\Pi^k \equiv \Pi(S^k) \leq \Pi^0$ and $L(S^k/G) \subseteq L(G)$ $\forall S^k \in \mathcal{S}$. The task is to construct an optimal cost matrix $\Pi^* \leq \Pi^0$ that maximizes the performance vector

$$\mu^* \equiv [I - \Pi]^{-1} X$$

i.e.

$$\mu^* \geq \mu^k \equiv [I - \Pi^k]^{-1} \mathrm{X} \qquad \forall \Pi^k \leq \Pi^0$$

where the inequalities are implied elementwise. More details on construction of the optimal control policy are reported in Fu *et al.* (2003 b). This is an area of ongoing research and the completed work is expected to be reported in a future publication.

6. An application example

As an example of state-based supervisory control, this section presents the design and performance analysis of discrete-event supervisory controllers for a twin-engine unmanned aircraft that is used for surveillance and data collection. Engine health and operating conditions are monitored in real time based on the information derived from the observed data. In the event of any abnormality being detected, the supervisor may decide to continue or abort the mission. Engine health and operating conditions, which are monitored in real time, are classified into three mutually exclusive and exhaustive categories: *Good*, *Unhealthy* (but operable) and *Inoperable*.

The plant automaton model in figure 1 has 13 states (excluding the dump state), of which three are marked states, and nine events, of which four are controllable and the remaining five are uncontrollable. All events are assumed to be observable. The states and events of the plant model are listed in table 1 and table 2, respectively.

The state transition function δ and the state-based event cost $\tilde{\pi}_{ij}$ (see Definition 8) are entered simultaneously in table 3. The fraction part in each entry denotes the corresponding state-based event cost $\tilde{\pi}_{ij}$ such that each row sum of the event cost matrix $\tilde{\Pi}$ is strictly less than one. The integer part (within parentheses) in each entry denotes the respective destination resulting from the occurrence of the event. The values of $\tilde{\pi}_{ij}$ were selected by extensive simulation experiments on gas turbine engine models and were also based on experience of gas turbine engine operation and maintenance, following the procedure reported by Wang *et al.* (2003 a). The dump state and any transitions to the dumped state are not shown in table 3. The elements of the characteristic vector (see Definition 7) were chosen as signed real weights based on the perception of each marked state's role on the gas turbine engine performance.

Table 4 lists the characteristic values of the 13 states in table 1. These parameters are selected by the designer based on the perception of each marked state's role in the system performance. As the states 1 to 10 are not marked, the first 10 elements of the characteristic vector $\tilde{\chi}$ in table 4 are zeros. The implication is that event strings terminating at states 1 to 10 have zero measure.

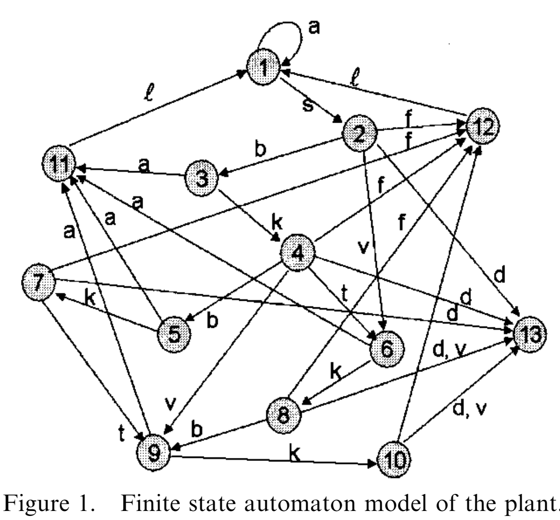

Figure 1. Finite state automaton model of the plant.

State#	State-description
1	Safe in base
2	Mission executing—two good engines
3	One engine unhealthy
4	Mission executing—one good and one unhealthy engine
5	Both engines unhealthy
6	One engine good and one engine inoperative
7	Mission execution with two unhealthy engines
8	Mission execution with only one good engine
9	One engine unhealthy and one engine inoperative
10	Mission execution with only one unhealthy engine
11	Mission aborted/not completed (Bad Marked State)
12	Mission successful (Good Marked State)
13	Aircraft destroyed (Bad Marked State)

Table 1. Plant automaton states.

Event	Event description
s	start and take-off (Controllable)
b	one good engine being unhealthy (Uncontrollable)
t	one unhealthy engine being inoperable (Uncontrollable)
v	one good engine being inoperable (Uncontrollable)
k	keep engine(s) running (Controllable)
a	mission abortion (Controllable)
f	mission completion (Uncontrollable)
d	destroyed aircraft (Uncontrollable)
ℓ	landing (Controllable)

Table 2. Plant event alphabet.

	s	b	t	v	k	a	f	d	l
1	0.5 (2)					0.02 (1)			
2		0.05 (3)		0.01 (6)			0.8 (12)	0.1 (13)	
3					0.45 (4)	0.45 (11)			
4		0.12 (5)	0.16 (6)	0.1 (9)			0.5 (12)	0.12 (13)	
5					0.45 (7)	0.45 (11)			
6					0.45 (8)	0.45 (11)			
7			0.25 (9)				0.5 (12)	0.2 (13)	
8		0.2 (9)		0.01 (13)			0.3 (12)	0.4 (13)	
9					0.45 (10)	0.45 (11)			
10			0.35 (13)				0.2 (12)	0.40 (13)	
11									0.95 (1)
12									0.95 (1)
13									

Table 3. State transition and event cost matrix.

$$\bar{\chi} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.05 & +0.25 & -1.0 \end{bmatrix}^{\mathrm{T}}$$

Table 4. Characteristic vector.

The state 12 is a good marked state having a positive χ value and the bad marked states 11 and 13 have negative χ values. Therefore, event strings terminating at states 12 have a positive measure and those terminating at states 11 and 13 have negative measures.

Three supervisory controllers were designed independently using a graphical interactive package (Wang *et al.* 2003 b) based on the following specifications:

- *Specification #1*: At least one of the two engines must be in good condition for mission continuation.
- *Specification #2*: Do not continue the mission if any one of the two engines is inoperable.
- *Specification #3*: Do not continue the mission unless both engines are in good condition.

Figures 2, 3 and 4 show the finite-state machine diagrams of the supervised plant under control specifications 1, 2 and 3, respectively. The dashed lines in these figures indicate that the transitions under corresponding controllers have been deleted from the plant model as a result of disabled (controllable) events. The performance measure μ_1 (i.e. with the initial state 1) of the uncontrolled plant is 0.0823 and μ_1 for three supervised plants under specifications #1, #2, and #3 were evaluated to be: 0.0807, 0.0822, and 0.0840, respectively. Therefore, the performance of the supervised plant under specifications #1, #2, and #3 is inferior, similar and superior to that of the unsupervised plant from perspectives of the mission objectives as described by the language measure parameters $\tilde{\pi}$ and χ. The supervisor #3 yields the best performance among the three supervisors based on language measure parameters in tables 3 and 4.

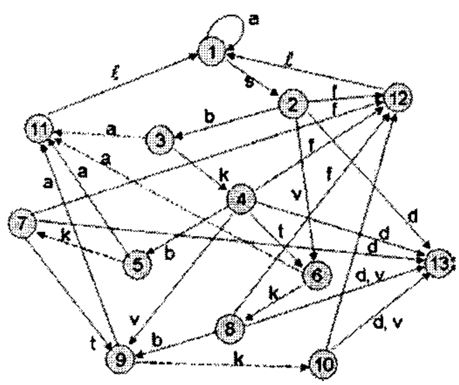

Figure 2. Supervised plant under specification #1.

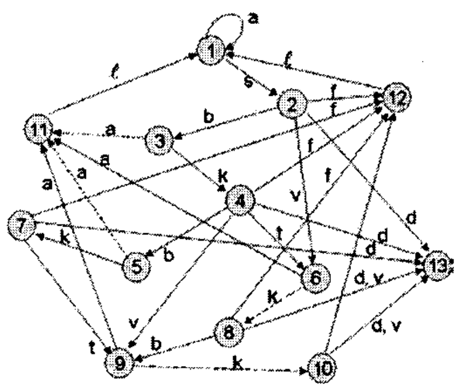

Figure 3. Supervised plant under specification #2.

An optimal supervisory controller was designed for the plant model as reported in Fu *et al.* (2003 b). The performance of each of these controllers is inferior to the performance, 0.0850, of the optimal controller as expected.

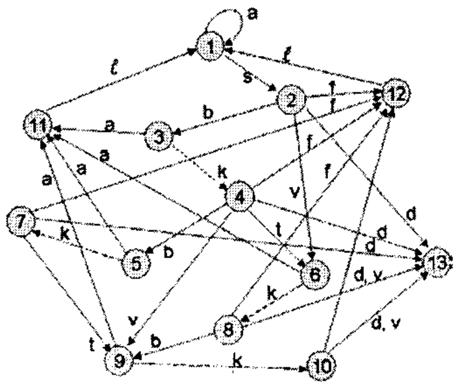

Figure 4. Supervised plant under specification #3.

7. Summary, conclusions and future work

This paper presents a signed real measure of regular languages, which is based on an event cost matrix and a characteristic vector. While the parameters of characteristic values are selected based on the perception of each marked state's role in the system performance, the even costs (i.e. $\tilde{\pi}_{ij}$'s) are identified from analysis of experimental observations or by extensive simulation experiments (Wang *et al.* 2003 a). Computational complexity of the language measure is polynomial in the number of states of the deterministic finite state automaton that is a minimal realization of the regular language.

The language measure provides a tool for performance analysis and comparison of the unsupervised plant automaton and supervised plant automata. The total variation of this language measure induces a metric on the vector space of sublanguages of the given regular language, which is defined over the Galois field *GF*(2).

The main motivation for constructing the language measure is quantitative analysis of the performance of finite-state automata that can be represented by regular languages. This facilitates quantified analysis and synthesis of discrete event supervisory (DES) control laws for complex dynamical systems. Feasibility of the language-measure-based approach for DES control system analysis is demonstrated on a finite-state machine model of a twin-engine unmanned aircraft for surveillance and data collection. Usage of the language measure for optimal DES control synthesis is an active research area. The initial phase of the work in this direction has been recently reported in technical literature (Fu *et al.* 2002, 2003 a,b).

The state-based measure, reported in this paper, is restricted to regular languages that represent finite-state machine models of dynamical systems. Since many physical processes may require more elaborate modelling (for example, Petri net representation), it is necessary to extend this measure to non-regular languages that, even in the simplest form (e.g. deterministic push-down automata Martin 1997), may not be represented by finitely many states. Future research in this direction is recommended. The initial effort for construction of non-regular languages is envisioned as follows:

Step 1. Construct the language measure in terms of its generating grammar without referring to states of the automaton.

Step 2. Generalize the signed real measure for regular languages to a complex measure for the class of non-regular languages generated by linear context free grammars (LCFG) (Martin 1997).

Acknowledgements

This work was supported in part by Army Research Office (ARO) under Grant No. DAAD19-01-1-0646; and NASA Glenn Research Center under Grant No. NAG3-2940.

References

Drobot, V., 1989, *Formal Languages and Automata Theory* (Rockville, MD: Computer Science Press).

Fu, J., Lagoa, C., and Ray, A., 2003 a, Robust optimal control of regular languages with event disabling cost. *42nd IEEE Conference on Decision and Control*, Maui, Hawaii, USA, December.

Fu, J., Ray, A., and Lagoa, C., 2002, Unconstrained optimal control of regular languages. *41st IEEE Conference on Decision and Control*, Las Vegas, Nevada, USA, December.

Fu, J., Ray, A., and Lagoa, C., 2003 b, Optimal control of regular languages with event cost. *American Control Conference*, Denver, Colorado, USA, June.

Hopcroft, H. E., Motwani, R., and Ullman, J. D., 2001, *Introduction to Automata Theory, Languages, and Computation*, 2nd edn (Reading, MA: Addison Wesley).

Martin, J. C., 1997, *Introduction to Languages and the Theory of Computation*, 2nd edn (New York, McGraw-Hill).

Naylor, A. W., and Sell, G. R., 1982, *Linear Operator Theory in Science and Engineering* (New York: Springer-Verlag).

Rudin, W., 1988, *Real and Complex Analysis*, 3rd edn (New York: McGraw Hill).

Ray, A., and Phoha, S., 2002, A language measure for discrete event automata. *International Federation of Automatic Control (IFAC) World Congress* b'02, Barcelona, Spain, July.

Ramadge, P. J., and Wonham, W. M., 1987, Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, **25**, 206–230.

Wang, X., and Ray, A., 2002, Signed real measure of regular languages. *American Control Conference*, Anchorage, Alaska, USA, May.

Wang, X., Ray, A., and Khatkhate, A., 2003 a, On-line identification of languages measure parameters for discrete event supervisory control. *42nd, IEEE Conference in Decision and Control*, Maui, Hawaii, USA.

Wang, X., Ray, A., Phoha, S., and Liu, J., 2003 b, DES: a graphical interactive package for analysis of discrete event systems. *American Control Conference*, Denver, Colorado, USA.