

A LANGUAGE MEASURE FOR PERFORMANCE QUANTIFICATION OF DISCRETE EVENT SUPERVISORY CONTROL SYSTEMS¹

XI WANG¹ AND ASOK RAY²

(Received 8 March 2003)

Abstract. This paper formulates a signed real measure of sub-languages of a regular language based on the principles of automata theory and real analysis. The measure allows total ordering of any set of partially ordered sublanguages of the regular language for quantitative evaluation of the controlled behaviour of deterministic finite state automata under different supervisors. The computational complexity of the language measure algorithm is of polynomial order in the number of states.

1. Introduction. The legal behavior of a physical plant is often modeled by a deterministic finite-state automaton abbreviated as DFSA in the sequel, which is equivalent to a regular language (Drobot 1989, Hopcroft et al. 2001, Martin 2001). A parallel combination of the plant model and the supervisor is a sublanguage of the plant language. This sublanguage enables restricted legal behavior of the controlled plant (Ramadge and Wonham 1987). *Based on the Myhill-Nerode Theorem*, the plant language is partitioned into equivalence classes of finite-length event strings. Each marked state is characterized by a signed real value that is chosen based on the designer's perception of the state's impact on the system performance. Conceptually similar to the conditional probability, each event is assigned a cost based on the state at which it is generated. This procedure permits an event string leading to a good (bad) marked state to have a positive (negative) measure. A supervisor can be designed in this setting such that the controlled sub-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 behavior under different supervisors. This feature can be formalized as follows :

Given that the relation \subseteq induces a partial ordering on a set of controlled sublanguages $\{L(S_j/G), j = 1, \dots, m\}$ of the plant language $L(G)$, the language measure μ induces a total ordering \leq on $\{\mu(L(S_j/G))\}$. In other words, the range of the set function μ is totally ordered while its domain could be partially ordered.

¹This work has been supported in part by Army Research Office (ARO) under Grant No. DAAD19-01-1-0646; and NASA Glenn Research Center under Grant No. NNC04GA49G.

Numerical evaluation of the language measure requires Gaussian elimination of a single variable involving a real square matrix of dimension equal to the number of states. As such the computational complexity of the algorithm is of polynomial order in the number of states.

The contribution of this paper is formulation of a novel concept for quantification of discrete event system (DES) performance. To this end, a measure of regular languages has been constructed based on the fundamental principles of real analysis and automata theory. This is a signed real measure that can be used to quantify any sublanguage of a given regular language based on specified parameters as explained in subsequent sections. As such, new performance indices of DES supervisors can be defined in terms of the proposed language measure, regardless of how the supervisor is designed (e.g., maximally permissive or not; blocking or non-blocking; and completely or partially observed).

2. DFSA Language Measure. Let $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ be a DFSA model that represents the discrete-event dynamic behavior of a physical plant. Let n denote the cardinality of the state set Q , i.e., $|Q| = n$, and $\mathcal{I} = \{1, \dots, n\}$ the index of Q ; Σ the (finite) alphabet of events; Σ^* is the set of all finite-length strings of events including the empty string ε ; $\delta : Q \times \Sigma \rightarrow Q$ is a (possibly partial) function of state transitions and $\delta^* : Q \times \Sigma^* \rightarrow Q$ is an extension of δ ; the state q_i is the initial state; and Q_m is the set of marked states $\emptyset \subseteq Q_m \subset Q$.

DEFINITION 2.1 *The language $L(G_i)$ generated by a DFSA G_i initialized at the state $q_i \in Q$ is defined as :*

$$L(G_i) = \{s \in \Sigma^* \mid \delta^*(q_i, s) \in Q\} \quad (1)$$

The language $L_m(G_i)$ marked by a DFSA G_i initialized at the state $q_i \in Q$ is defined as :

$$L_m(G_i) = \{s \in \Sigma^* \mid \delta^*(q_i, s) \in Q_m\} \quad (2)$$

The set Q_m of marked states is partitioned into Q_m^+ and Q_m^- , i.e., $Q_m = Q_m^+ \cup Q_m^-$ and $Q_m^+ \cap Q_m^- = \emptyset$, where Q_m^+ contains all good marked states that we desire to reach, and Q_m^- contains all bad marked states that we want to avoid, although it may always be possible to completely avoid the bad states while attempting to reach the good states. In general, the marked language $L_m(G_i)$ consists of both good and bad event strings that, starting from the initial state, q_i , respectively lead to Q_m^+ and Q_m^- . Any event string belonging to the language $L^0(G_i) = L(G_i) - L_m(G_i)$ leads to one of the non-marked states belonging to $Q - Q_m$ and $L^0(G_i)$ does not contain any one of the good or bad strings. Ray and Phoha (2002) have provided a detailed explanation on partitioning of the language into positive, negative, and zero measures following the Hahn Decomposition Theorem (Rudin 1987).

DEFINITION 2.2 *For every $q_k \in Q$, let $L(q_i, q_k)$ denote the set of all strings that, starting from the state q_i , terminate at the state q_k , i.e.,*

$$L(q_i, q_k) = \{s \in \Sigma^* \mid \delta^*(q_i, s) = q_k \in Q\} \quad (3)$$

Based on the equivalence classes defined in the Myhill-Nerode Theorem (Hopcroft, Motwani and Ullman 2001), the regular languages $L(G_i)$ and $L_m(G_i)$ can be expressed as :

$$L(G_i) = \bigcup_{q_k \in Q} L(q_i, q_k) = \bigcup_{k=1}^n L(q_i, q_k) \quad (4)$$

$$L_m(G_i) = \bigcup_{q_k \in Q_m} L(q_i, q_k) = L_m^+ \cup L_m^- \quad (5)$$

where the sublanguage $L(q_i, q_k) \subseteq G_i$ having the initial state q_i is uniquely labeled by the terminal state $q_k, k \in \mathcal{I}$ and $L(q_i, q_j) \cap L(q_i, q_k) \equiv \emptyset \forall j \neq k$; and

$$L_m^+ \equiv \bigcup_{q \in Q_m^+} L(q, q_i)$$

$$\text{and } L_m^- \equiv \bigcup_{q \in Q_m^-} L(q_i, q)$$

are good and bad sublanguages of $L_m(G_i)$, respectively. Then,

$$L^0(G_i) = \bigcup_{q \notin Q_m} L(q_i, q)$$

$$\text{and } L(G_i) = L^0(G_i) \cup L_m^+(G_i) \cup L_m^-(G_i).$$

DEFINITION 2.3 Θ be a σ -algebra of $L(G_i)$. Then, the set function $\mu : \Theta \rightarrow \mathbf{R} \equiv (-\infty, +\infty)$, is called a signed real measure if the following two conditions are satisfied. (Rudin 1987)

$$(1) \mu(\emptyset) = 0;$$

$$(2) \mu \left(\bigcup_{j=1}^{\infty} K_j \right) = \sum_{j=1}^{\infty} \mu(K_j) \forall K_j \in \Theta \text{ and } K_j \cap K_k = \emptyset \text{ if } j \neq k.$$

Now we construct a signed real measure $\mu : 2^{L(G_i)} \rightarrow \mathbf{R}$ for a given DFSA such that :

$$\forall q \in Q, \quad \mu(L(q, q_i)) \begin{cases} = 0, & q \notin Q_m \\ > 0, & q \in Q_m^+ \\ < 0, & q \in Q_m^- \end{cases} \quad (6)$$

To achieve the above goal of signed measure, we characterize the marked states such that each state in Q_m^+ is assigned a positive weight and each state in Q_m^- a negative weight; and each unmarked state is assigned the zero weight. The weights are chosen by the designer based on the perception of each marked state's role in the system performance.

DEFINITION 2.4 The characteristic function $\chi : Q \rightarrow [-1, 1]$ that assigns a signed real weight to state-based sublanguages $L(q_i, q)$ is defined as :

$$\forall q \in Q, \quad \chi(q) \in \begin{cases} [-1, 0), & q \in Q_m^- \\ \{0\}, & q \notin Q_m \\ (0, 1], & 1 \in Q_m^+ \end{cases} \quad (7)$$

The state weighting vector, denoted by $\mathbf{X} = [\chi_1 \chi_2 \cdots \chi_n]^T$, where $\chi_k \equiv \chi(q_k) \forall k$, is called the \mathbf{X} -vector. The k -th element χ_k of \mathbf{X} -vector is the weight assigned to the corresponding terminal state q_k .

To compute the measure of the language $L(q_i, q)$, we assign a cost to each string terminating at the state q starting from the initial state q_i . To this end, the event cost is defined conceptually similar to the conditional transition probability, assuming that the DFSA model is Markov.

DEFINITION 2.5 The event cost of the DFSA G_i is defined as a (possibly partial) function $\tilde{\pi} : \Sigma^* \times Q \rightarrow [0, 1)$ such that $\forall q_i \in Q, \forall \sigma_j \in \Sigma, \forall s \in \Sigma^*$,

- (1) $\tilde{\pi}[\sigma_j, q_i] \equiv \tilde{\pi}_{ij} \in [0, 1)$; $\sum_j \tilde{\pi}_{ij} < 1$;
- (2) $\tilde{\pi}[\sigma_j, q_i] = 0$ if $\delta(q_i, \sigma_j)$ is undefined; $\tilde{\pi}[\varepsilon, q_i] = 1$;
- (3) $\tilde{\pi}[\sigma_j s, q_i] = \tilde{\pi}[\sigma_j, q_i] \tilde{\pi}[s, \delta(q_i, \sigma_j)]$.

Now we introduce the language measure in terms of the event cost function $\tilde{\pi}$ and the characteristic function χ .

DEFINITION 2.6 The signed real measure of every singleton string set $\{s\} \in 2^{L(G_i)}$, where $s \in L(q, q_i)$, is defined as :

$$\mu(\{s\}) \equiv \tilde{\pi}[s, q_i] \chi(q)$$

implying that

$$\forall s \in L(q_i, q), \quad \mu(\{s\}) \begin{cases} = 0, & q \notin Q_m \\ > 0, & q \in Q_m^+ \\ < 0, & q \in Q_m^- \end{cases} \quad (8)$$

It follows from 2.6 that the signed measure of the sub-language

$$L(q_i, q) \subseteq L(G_i),$$

of all events starting at q_i and terminating at q is :

$$\mu(L(q_i, q)) = \left(\sum_{s \in L(q_i, q)} \tilde{\pi}[s, q_i] \right) \chi(q) \quad (9)$$

DEFINITION 2.7 *The signed real measure of the language of a DFSA G_i initialized at a state $q_i \in Q$, is defined as :*

$$\mu_i \equiv \mu(L(G_i)) = \sum_{q \in Q} \mu(L(q_i, q)) \quad (10)$$

The language measure vector, denoted as

$$\mu = [\mu_1 \ \mu_2 \ \cdots \ \mu_n],$$

is called the μ -vector.

Since $\mu(L(q, q_i)) = 0 \ \forall q \notin Q_m$ (Definition 2.6), it follows from Definition 2.7 that μ_i is the signed measure of the marked language $L_m(G_i)$ of the DFSA G_i , i.e.,

$$\mu_i = \mu(L_m(G_i)).$$

3. Language Measure Computation. Various methods of obtaining regular expressions for DFSAs are reported in Hopcroft (2001) and Drobot (1989). While computing the measure of a given DFSA, the same event may have different significance when emanating from different states. This requires assigning (possibly) different values to the same event defined on different states. Therefore, it is necessary to obtain a regular expression which explicitly yields the state-based event sequences. A procedure for language measure computation, which is more elegant than that proposed by Ray and Phoha (2002), is presented below.

DEFINITION 3.1 *Let $L_i \equiv L(G_i), i \in \mathcal{I}$, denote the regular expression representing the marked language of an n -state DFSA $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ where q_i is the initial state.*

DEFINITION 3.2 *Let σ_j^k denote the set of event(s) $\sigma \in \Sigma$ that is defined on the state q_j and leads to the state $q_k \in Q$, where $j, k \in \mathcal{I}$, i.e., $\delta(q_j, \sigma) = q_k, \forall \sigma \in \sigma_j^k \subseteq \Sigma$.*

LEMMA 3.1 *Let u, v be two known regular expression and r be an unknown regular expression that satisfies the following algebraic identity :*

$$r = ur + v \quad (11)$$

Then, the following relations are true :

- (1) $r = u^*v$ is a solution to equation 11
- (2) If $\varepsilon \notin u$, then $r = u^*v$ is the unique solution to equation 11.

The proof of Lemma 3.1 can be found in (Drobot 1989).

EXAMPLE 3.1 *Let $\Sigma = \{a, b\}$, $Q = \{1, 2, 3\}$, the unitial state is 1 the sole marked state is 2 in Figure 1. Let the set of linear algebraic equations represent the transitions at each state of the DFSA.*

$$\begin{cases} L_1 = a_1^1 L_1 + b_1^2 L_2 \\ L_2 = a_2^1 L_1 + b_2^3 L_3 + \varepsilon \\ L_3 = a_3^1 L_1 + b_3^2 L_2 \end{cases} \quad (12)$$

where the 'forcing' term ε is introduced on the right side of the i -th equation whenever $q_i \in Q_m$, $i \in \mathcal{I}$. By application of Lemma 3.1, the regular expression for the marked language $L_m(G_1)$ is :

$$= L_1 - (a_1^2)^* b_1^2 \left(a_2^1 (a_1^1)^* b_1^2 + b_2^3 a_3^1 (a_1^1)^* b_1^2 + b_2^3 a_3^2 \right)^*$$

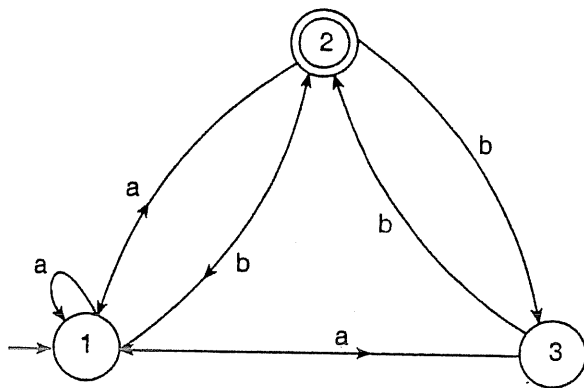


Fig. 1. Example 1

The method of system description in Example 3.1 can be extended to the general case without any difficulty. Given a DFSA $G_i = (Q, \Sigma, \delta, q_i, Q_m)$ with $|Q| = n$, we proceed to obtain the system equation by a set of regular expressions L_i of the language $L(G_i)$, $i \in \mathcal{I}$, as follows :

$$\forall q_i \in Q, \quad L_i = \sum_j R_{i,j} + \mathcal{E}_i, \quad i \in \mathcal{I} \quad (13)$$

where $\forall i$, $R_{i,j}$ is defined as :

1. If $\exists \sigma \in \Sigma$, such that $\delta(q_i, \sigma) = q_j \in Q$, $j \in \mathcal{I}$, then $R_{i,j} = \sigma_i^j L_j$, otherwise, $R_{i,j} = \emptyset$.
2. If $q_i \in Q_m$, $\mathcal{E}_i = \varepsilon$, otherwise, $\mathcal{E}_i = \emptyset$.

The set of symbolic equations may be written as :

$$L_i = \sum_j \sigma_i^j L_j + \mathcal{E}_i \quad (14)$$

We note the following special cases.

1. If $\mathcal{E}_i = \emptyset$, $\forall L_i$, then $L_m(G) = \emptyset$. This implies that the DFSA has no marked state.
2. If $\exists q_i \in Q$ such that $L_i = \varepsilon$, then q_i is marked. Furthermore, q_i is a deadlock state.

In order to convert the symbolic equations (14) into a set of algebraic equations, we introduce the (one-hop) state transition cost that is defined below.

DEFINITION 3.3 The state transition cost of the DFSA G_i is defined as a function $\pi : Q \times Q \rightarrow [0, 1)$ such that $\forall q_i, q_j \in Q, \pi[q_i, q_j] = \sum_{\sigma \in \Sigma; \delta(q_i, \sigma) = q_j} \tilde{\pi}[\sigma, q_i] \equiv \pi_{ij}$ and $\pi_{ij} = 0$ if $\{\sigma \in \Sigma : \delta(q_i, \sigma) = q_j\} = \emptyset$. The $n \times n$ state transition cost matrix is defined as :

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

and is referred to as the $\mathbf{\Pi}$ -matrix in the sequel.

Now we present an alternative form of language measure (Definition 2.7) in terms of the state transition cost (Definition 3.3) instead of event cost (Definitions 2.5) as delineated below.

THEOREM 3.1 The language measure of the symbolic equation 14 is given by

$$\mu_i = \sum_j \pi_{ij} \mu_j + \chi_i \tag{15}$$

Proof: Following Equation 13 and Definition 2.4 :

$$\forall i \in \mathcal{I} \quad \mu(\mathcal{E}_i) = \begin{cases} \chi_i & \text{if } \mathcal{E}_i = \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Therefore, $\mathbf{X} = [\chi_1 \ \chi_2 \ \cdots \ \chi_n]^T$ is the forcing vector in Equation 14. Starting from the state q_i , the measure

$$\begin{aligned} \mu_i &= \mu(\dot{L}_i) = \mu \left(\sum_j \sigma_i^j L_j + \mathcal{E}_i \right) \\ &= \mu \left(\sum_j \sigma_i^j L_j \right) + \mu(\mathcal{E}_i) \\ &= \sum_j \mu(\sigma_i^j L_j) + \mu(\mathcal{E}_i) \\ &= \sum_j \pi(\sigma_i^j) \mu(L_j) + \mu(\mathcal{E}_i) \\ &= \sum_j \pi_{ij} \mu(L_j) + \mu(\mathcal{E}_i) \\ &= \sum_j \pi_{ij} \mu_j + \chi_i \end{aligned}$$

The third equality in the above derivation follows from the fact that $\mathcal{E}_i \cap \sigma_i^j L_j = \emptyset$. It is also true that

$$\forall j \neq k, \quad \sigma_i^j L_j \cap \sigma_i^k L_k = \emptyset \quad (17)$$

since each string in $\sigma_i^j L_j$ starts with an event in σ_i^j while each string in $\sigma_i^k L_k$ starts from an event in σ_i^k for some $k \neq j$. This justifies the fourth equality. Since the DFSA model is modeled to be Markov, $\mu(\sigma_i^j L_j) = \mu(\sigma_i^j) \mu(L_j)$. Therefore, by Definitions 2.5 and 3.2, $\mu(\sigma_i^j L_j) = \pi[q_j|q_i] \mu(L_j) = \pi_{ij} \mu(L_j)$.

In vector notation, Equation 15 in Theorem 3.1 is expressed as : $\mu = \Pi \mu + X$ whose solution is given by :

$$\mu = (I - \Pi)^{-1} X \quad (18)$$

provided that the matrix $I - \Pi$ is nonsingular. Definitions 2.5 and 3.3 state that each element in the Π -matrix is non-negative and each row sum is less than 1. These conditions make the Π -matrix a contraction operator that is sufficient for the matrix $(I - \Pi)^{-1}$ to be a bounded linear operator (Naylor and Sell 1982). Therefore, Definitions 2.5 and 3.3 provide a sufficient condition for the language measure μ of the DFSA G to be finite.

The j -th element of the i -th row of the $(I - \Pi)^{-1}$ matrix, denoted as ν_i^j , is the language measure of a DFSA with the same state transition function δ as G and having the following properties : (i) the initial state is q_i ; (ii) q_j is the only marked state; and (iii) the χ -value of q_j is equal to 1. Then, $\mu_i \equiv \mu(L_m(G_i))$ is given by $\mu_i = \sum_j \nu_i^j \chi_j$. Numerical evaluation of the language measure of G_i requires Gaussian elimination of the single variable μ_1 involving the real square matrix $(I - \Pi)$. As such the computational complexity of the language measure algorithm is of polynomial order in the number of states.

In general, the language measure of G_i does not require computation of μ_j , $j \neq i$. However, on-line supervisory control may require the information on the performance of the automation starting from the current state q_i , $i \neq 1$. In that case, $\mu_i = \sum_j \nu_i^j \chi_j$, $i \neq 1$ should be computed.

EXAMPLE 3.2 (Example 3.1 revisited) *Let us assign the Π -matrix and X -vector in Example 3.1 as follows :*

$$\Pi = \begin{pmatrix} 0.3 & 0.4 & 0 \\ 0.2 & 0 & 0.6 \\ 0.5 & 0.4 & 0 \end{pmatrix}, \quad X = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

then $\mu = (I - \Pi)^{-1} X = [1.2048 \ 2.1084 \ 1.4458]^T$, therefore, $\mu_1 = 1.2048$, $\mu_2 = 2.1084$, and $\mu_3 = 1.4458$.

4. Usage of the Language Measure for Supervisor Synthesis. The (signed) language measure μ could serve as an index for synthesis of an optimal control policy that maximizes the performance of a controlled sublanguage. The salient concept is succinctly presented below.

Let $\mathcal{S} \equiv \{S^0, S^1, \dots, S^N\}$ be a set of supervisory control policies for the open loop plant automation G where S^0 is the null controller (i.e., no event is disabled) implying that

I
F
C
—
C
H
co
M
Ne

$L(S^0/G) = L(G)$. Therefore the controller cost matrix $\Pi(S^0) = \Pi^0$ that is the Π -matrix of the open loop plant automation G . For a supervisor $S^k, k \in \{1, 2, \dots, N\}$, 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 synthesize an optimal cost matrix $\Pi^* \leq \Pi^0$ that maximizes the performance vector $\mu^* \equiv [I - \Pi^*]^{-1}X$, i.e., $\mu^* \geq \mu^k \equiv [I - \Pi^k]^{-1}X \forall \Pi^k \leq \Pi^0$ where the inequalities are implied elementwise. The research work in this direction is in progress and some of the results are reported in recent publications (Fu et al. 2002, 2003).

5. Summary and Conclusions. This paper presents the concept, formulation and validation of a signed real measure for any regular language and its sublanguages. Specifically, the relative performance of supervisors can be quantitatively evaluated in terms of the measure of the controlled sublanguages. Positive weights are assigned to good marked states and negative weights to bad marked states so that a controllable supervisor is regarded (penalized) for deleting strings leading to bad (good) marked states. As such the measure of the (open loop) plant language may be less than that of a (proper) controlled sublanguage.

Cost assignment to each event based on the state, where it is generated, is conceptually similar to the conditional probability of the event. The procedure of controller evaluation in terms of its language measure is validated by the well-known example of dining philosophers for three different supervisors. A relatively less permissive supervisor could be more effective than another supervisor that may not adequately delete event strings leading to bad marked states. The computational complexity of the language measure algorithm is of polynomial order in the number of states.

Potential applications of the language measure are model identification, model order reduction, and analysis and synthesis of robust and optimal control and diagnostic systems in the DES setting.

6. Acknowledgements. The authors acknowledge benefits of discussion with Mr. Amit Surana and Mr. Jinbo Fu.

References

- Drobot, V. (1989) : *Formal languages and automata theory*, Computer Science Press.
- Fu, J., Ray, A. and Lagoa, C. (2002) : *Unconstrained optimal control of regular languages*, 41st IEEE Conference on Decision and Control.
- (2003) *Optimal control of regular languages with event disabling cost*, Submitted to American Control Conference.
- Hopcroft, J.E., Motwani, R. and Ullman, J.D. (2001) : *Introduction to automata theory, languages, and computation*, 2nd ed., Addison-Wesley.
- Martin, J.C. (2001) : *Introduction to languages and the theory of computation*, 2nd ed., McGraw-Hill.
- Naylor, A.W. and Sell, G.R. (1982) : *Linear operator theory in engineering and science*, Springer-Verlag.

Ramadge, P.J. and Wonham, W.M. (1987) : Supervisory control of a class of discrete event processes, *SIAM J. Control and Optimization*, **25**, 1, 206.

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

Rudin, W. (1987) : *Real and complex analysis*, 3rd ed., McGraw-Hill.

DEPARTMENT OF MECHANICAL ENGINEERING
THE PENNSYLVANIA STATE UNIVERSITY
UNIVERSITY PARK, PA 16802

¹E-MAIL : xxw117@psu.edu

²E-MAIL : axr2@psu.edu

Corresponding Author: ASOK RAY

l
s
g
e
in

sc
in
ma
Mc
sp
sp
den

DEL
(i)
(ii)
(iii)
(2
 $\cap\{F :$