

Language-measure-theoretic optimal control of probabilistic finite-state systems

I. CHATTOPADHYAY and A. RAY*

Mechanical Engineering Department, The Pennsylvania State University,
University Park, PA 16802, USA

(Received 10 October 2006; in final form 17 February 2007)

Supervisory control theory for discrete event systems, introduced by Ramadge and Wonham, is based on a non-probabilistic formal language framework. However, models for physical processes inherently involve modelling errors and noise-corrupted observations, implying that any practical finite-state approximation would require consideration of event occurrence probabilities. Building on the concept of signed real measure of regular languages, this paper formulates a comprehensive theory for optimal control of finite-state probabilistic processes. It is shown that the resulting discrete-event supervisor is optimal in the sense of elementwise maximizing the renormalized language measure vector for the controlled plant behaviour and is efficiently computable. The theoretical results are validated through several examples including the simulation of an engineering problem.

1. Introduction

Supervisory control theory (SCT) of discrete-event systems (DES), pioneered by Ramadge and Wonham (1987), models a physical or human-engineered process as a finite-state language generator and constructs a supervisor that attempts to constrain the “supervised” plant behaviour within a specification language. The original theory is based on a deterministic language framework. Although allowing non-determinism in the sense that more than one continuation of a generated event trace (i.e., a string) is possible, no attempt is made to quantify this randomness. As Wonham himself observes in Lawford and Wonham (1993), “the choice of a possible continuation of a string is made by some internal structure unmodeled by the systems designer”. The notion of probabilistic languages in the context of studying qualitative stochastic behaviour of discrete-event systems first appears in Garg (1992a, b), where the concept of p-languages (‘p’ implying probabilistic) is introduced and an algebra

is developed to model probabilistic languages based on concurrency (Milner 1989). A regular p-language is essentially a set of prefix-closed traces of events, generated by a finite-state automaton with probabilities associated with the transitions. A p-language-theoretic model differs in several important aspects from other discrete-event models of stochastic analysis such as Markov chains (Cassandras and Lafortune 1999), stochastic Petri nets (Molloy 1982, Chung *et al.* 1994), probabilistic automata (Rabin 1963, Paz 1971, Doberkat 1981), and fuzzy models (Lee and Zadeh 1969). Garg *et al.* (1999) and Kumar and Garg (2001) provide a brief comparison of the p-language-theoretic modelling paradigm with the above-mentioned theories.

Lawford and Wonham (1993) have attempted to extend discrete-event (SCT) to plants modelled by p-languages, where a formal statement of the probabilistic supervisory control problem (PSCP) first appears and the notion of probabilistic supervision is introduced by random disabling of controllable events. The key difference from other stochastic supervision approaches (e.g., Mortzavian 1993) lies in the fact that the computed probabilistic supervisor is not allowed to change the

*Corresponding author. Email: axr2@psu.edu

underlying plant dynamics in the following sense: “The probabilistic effect of random disablement is determined entirely by the plant”. The control objective is specified as a p-language and necessary and sufficient conditions are derived for existence of a probabilistic supervisor that attempts to restrict the plant language within the control specification in a probabilistic sense. The theory of supervision of p-languages is further developed by Kumar and Garg (2001), where the control objective is specified as upper and lower bound constraints. The upper bound is a non-probabilistic language that serves as a legality constraint, while the lower bound is a p-language. This relatively relaxed approach to control objective specification allows for a non-probabilistic supervisor that attempts to cut down illegal event traces, while ensuring that legal traces occur with probabilities greater than or equal to that specified by the lower bound. Intuitively, the designed supervisor stops “bad” strings from occurring while guaranteeing that “good” strings occur with some minimum pre-set frequency. However, construction of such a control objective specification may not be possible in many applications (e.g., battlefield command, control, communications, and intelligence (C³I) (Phoha *et al.* 2002)), especially if the decisions are to be made in real time. For the theory to be useful in practice, one must generate the specification from the definition of the physical problem at hand. Given that one has to come up with a non-probabilistic language to serve as the upper legality constraint and a probabilistic language to serve as the lower bound, this goal may not always be achievable. The situation becomes worse for non-stationary stochastic environments, where the control specifications may have to be updated online.

A significantly simplified approach to the above problem is reported by Ray (2005) and Ray *et al.* (2005), where the control objective is specified as characteristic weights on the states of the plant automaton. These weights are normalized in the interval $[-1, 1]$ with positive weights assigned to good states and negative weights to bad states. A signed real measure of regular languages (of event traces) is defined as a function of the characteristic weights and the state transition probabilities; and supervisory control laws are synthesized by elementwise maximizing the language measure vector (Ray *et al.* 2004, 2005). Intuitively, the supervisor ensures that the generated event traces cause the plant to visit the “good” states while attempting to avoid the “bad” states in a probabilistic sense. As mentioned earlier, Kumar and Garg’s work on supervisory control of probabilistic automata (Kumar and Garg 2001) also has a notion of “good” and “bad” strings. However, the classification is strictly binary; the theory has no way of saying if one “good” string is “better” than

another “good” string and *vice versa*. This implies that the supervisor must eliminate all bad strings and hence may not be optimal, or fail to exist if the conditions defined in Kumar and Garg (2001) are not satisfied. In contrast, in the measure-theoretic approach (Ray 2005), the event traces are less or more desirable in a continuous scale with the supervisor optimizing the controlled plant behaviour to ensure that the “most” desirable strings occur “most” often. This has an immediate advantage that the problem of existence disappears; the optimal supervisor always exists and can be computed effectively with polynomial complexity. The latter approach is, in one sense, closer to Markov chain modelling since the control specification is state-based. However, as shown by Ray (2005), this does not restrict the modeling power of the technique. It is shown in Kumar and Garg (2001) that, in general, maximally permissive supervisors are non-unique. For the measure-theoretic approach, however, the optimization is shown to yield unique maximal permissiveness among all optimal supervisors (Ray *et al.* 2004, 2005).

Optimal control in the context of discrete event dynamic systems has been addressed earlier by several investigators as cited in (Ray *et al.* 2004). For example, Sengupta and Lafortune (1998) have analysed non-probabilistic DES with assigned event and control costs; the optimal supervisor is computed in the framework of dynamic programming (DP) with two critical assumptions to guarantee polynomial complexity of the solution: all costs are strictly positive and there is only one marked state (Sengupta and Lafortune 1998, p. 34). The work reported in Ray (2005) and Ray *et al.* (2005) is different in the sense that the latter deals with probabilistic automata and the optimization, even in the completely general case, has guaranteed polynomial complexity of $O(n^3)$, where n is the number of states in the unsupervised plant model. The measure-theoretic approach was originally reported for a restricted class of terminating p-languages Ray (2005) and Ray *et al.* (2005); and this restriction has been eliminated in a subsequent publication (Chattopadhyay and Ray 2006a).

The notion of terminating and non-terminating automata is originally due to Garg (19992a, b). A probabilistic automaton is terminating if there exist states at which the sum of the probabilities of all defined events is strictly less than 1. The interpretation is that the difference of the sum from 1 is the probability that the plant terminates operation at that particular state. It is shown in Ray (2005) that the language measure vector can be expressed as $[\mathbb{I} - \mathbf{\Pi}]^{-1}\boldsymbol{\chi}$ where $\mathbf{\Pi}$ is the transition probability matrix and $\boldsymbol{\chi}$ is the characteristic vector, where π_{ij} is the probability of transition from the i th state to the j th state and χ_i is the characteristic weight of the

state i). A sufficient condition for the inverse of $[\mathbf{I} - \mathbf{\Pi}]$ to exist is that $\sum_j \pi_{ij} < 1 \ \forall i$, i.e., the plant has a strictly non-zero probability of termination from each state.

This paper eliminates the above restrictive assumption by adopting the recently reported renormalized measure of regular languages (Chattopadhyay and Ray 2006a) as the performance index. It also extends the measure-theoretic concept for optimal control of terminating plants (Ray *et al.* 2004) to non-terminating plant models, which requires a minor modification of the control philosophy as explained below.

Supervisors in the SCT paradigm are allowed to affect the underlying plant behaviour by selectively disabling controllable events (Ramadge and Wonham 1987). In case of terminating p-languages, a similar approach suffices; the supervisor selectively nulls the occurrence probability of controllable events to achieve the desired control objective. However, the non-terminating case poses a problem since any such disabling action converts the system to a terminating p-language (i.e., the probabilities of the events defined at a state fail to add up to 1). The solution (Kumar and Garg 2001) is to proportionately increase the probabilities of the remaining enabled events at the state at which event disabling is undertaken. An alternative approach is proposed in this paper, where each disabled event creates a self loop at the state (at which the event was generated) with occurrence probability of the original transition.

The paper is organized in six sections and an appendix. Section 2 lays down the basic framework of the analysis and briefly reviews the original notion of language measure (Ray 2005) and its renormalization (Chattopadhyay and Ray 2006a). Section 3 formulates the optimal control problem based on the concept of renormalized measure and presents the key results. Section 4 presents a solution of the optimal control problem and derives the necessary algorithms for its implementation. Section 5 presents an engineering example, where the optimal supervisor is designed for a three-processor message decoding system. The paper is summarized and concluded in §6 along with recommendations for future research. Appendix G establishes bounds on the derivatives of the renormalized measure that is necessary for formulation of the optimal control law in §3.

2. Preliminary concepts

This section briefly reviews the concept of signed real measure of regular languages Ray (2005) and Ray *et al.* (2005) followed by a review of the notion of renormalized measure and the pertinent notations used in the sequel.

2.1 Brief review of language measure

Let the plant behaviour be modelled as a deterministic finite state automaton (DFSA) as $G_i \triangleq (Q, \Sigma, \delta, q_i, Q_m)$, where Q is the finite set of states with $|Q| = n$, and $q_i \in Q$ is the initial state; Σ is the (finite) alphabet of events with $|\Sigma| = m$; the Kleene closure of Σ is denoted as Σ^* that is the set of all finite-length strings of events including the empty string ε ; the (possibly partial) function $\delta: Q \times \Sigma \rightarrow Q$ represents state transitions and $\delta^*: Q \times \Sigma^* \rightarrow Q$ is an extension of δ ; and $Q_m \subseteq Q$ is the set of marked (i.e., accepted) states.

Definition 1: The language $L(G_i)$ generated by the DFSA G_i is defined as $L(G_i) = \{s \in \Sigma^* \mid \delta^*(q_i, s) \in Q\}$.

Definition 2: The marked language $L_m(G_i)$ by the DFSA G_i is defined as $L_m(G_i) = \{s \in \Sigma^* \mid \delta^*(q_i, s) \in Q_m\}$.

The language $L(G_i)$ of the DFSA G_i is partitioned as the non-marked and the marked languages, $L^o(G_i) \triangleq L(G_i) - L_m(G_i)$ and $L_m(G_i)$, consisting of event strings that, starting from $q_i \in Q$, terminate at one of the non-marked states in $Q - Q_m$ and one of the marked states in Q_m , respectively. The set Q_m is partitioned into Q_m^+ and Q_m^- where Q_m^+ contains all good marked states that one may desire to terminate on, and Q_m^- contains all bad marked states that one would attempt to avoid terminating on, although it may not always be possible to bypass a bad state before reaching a good state. The marked language $L_m(G_i)$ is further partitioned into $L_m^+(G_i)$ and $L_m^-(G_i)$ consisting of good and bad strings that, starting from q_i , terminate on Q_m^+ and Q_m^- , respectively.

A signed real measure $\mu: 2^{\Sigma^*} \rightarrow R \triangleq (-\infty, \infty)$ is constructed for quantitative evaluation of every event string $s \in \Sigma^*$. The language $L(G_i)$ is decomposed into null, i.e., $L^o(G_i)$, positive, i.e., $L_m^+(G_i)$, and negative, i.e., $L_m^-(G_i)$ sublanguages.

Definition 3: The language of all strings that, starting at a state $q_i \in Q$, terminates on a state $q_j \in Q$, is denoted as $L(q_i, q_j)$. That is,

$$L(q_i, q_j) \triangleq \{s \in L(G_i) : \delta^*(q_i, s) = q_j\}. \quad (1)$$

Definition 4: The characteristic function that assigns a signed real weight to each state $q_i, i = 1, 2, \dots, n$, is defined as: $\chi: Q \rightarrow [-1, 1]$ such that

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

Definition 5: The event cost is conditioned on a *DFSA* state at which the event is generated, and is defined as $\tilde{\pi} : \Sigma^* \times Q \rightarrow [0, 1]$ such that $\forall q_j \in Q, \forall \sigma_k \in \Sigma, \forall s \in \Sigma^*$,

1. $\tilde{\pi}[\sigma_k, q_j] \triangleq \tilde{\pi}_{jk} \in [0, 1]; \sum_k \tilde{\pi}_{jk} < 1;$
2. $\tilde{\pi}[\sigma, q_j] = 0$ if $\delta(q_j, \sigma)$ is undefined; $\tilde{\pi}[\epsilon, q_j] = 1;$
3. $\tilde{\pi}[\sigma_k s, q_j] = \tilde{\pi}[\sigma_k, q_j] \tilde{\pi}[s, \delta(q_j, \sigma_k)].$

The event cost matrix, denoted as $\tilde{\Pi}$ -matrix, is defined as

$$\tilde{\Pi} = \begin{bmatrix} \tilde{\pi}_{11} & \dots & \tilde{\pi}_{1m} \\ \vdots & \ddots & \vdots \\ \tilde{\pi}_{n1} & \dots & \tilde{\pi}_{nm} \end{bmatrix}$$

An application of the induction principle to part (3) in Definition 5 shows $\tilde{\pi}[st, q_j] = \tilde{\pi}[s, q_j] \tilde{\pi}[t, \delta^*(q_j, s)]$. The condition $\sum_k \tilde{\pi}_{jk} < 1$ provides a sufficient condition for the existence of the real signed measure as discussed in Ray (2005) along with additional comments on the physical interpretation of the event cost.

Now let us define the measure of a sublanguage of the plant language $L(G_i)$ in terms of the signed characteristic function χ and the non-negative event cost $\tilde{\pi}$.

Definition 6: The signed real measure μ of a singleton string set $\{s\} \subseteq L(q_i, q_j) \subseteq L(G_i) \in 2^{\Sigma^*}$ is defined as

$$\mu(\{s\}) \triangleq \tilde{\pi}(s, q_i) \chi(q_j) \quad \forall s \in L(q_i, q_j).$$

The signed real measure of $L(q_i, q_j)$ is defined as

$$\mu(L(q_i, q_j)) \triangleq \sum_{s \in L(q_i, q_j)} \mu(\{s\})$$

and the signed real measure of a *DFSA* G_i , initialized at the state $q_i \in Q$, is denoted as

$$\mu_i \triangleq \mu(L(G_i)) = \sum_j \mu(L(q_i, q_j)).$$

Definition 7: The state transition cost of the *DFSA* is defined as a function $\pi : Q \times Q \rightarrow [0, 1)$ such that

$$\pi(q_j, q_k) = \begin{cases} 0 & \text{if } \{\sigma \in \Sigma : \delta(q_j, \sigma) = q_k\} = \emptyset \\ \sum_{\sigma \in \Sigma : \delta(q_j, \sigma) = q_k} \tilde{\pi}(\sigma, q_j) \triangleq \pi_{jk} & \text{otherwise} \end{cases} \quad (2)$$

The state transition cost matrix, denoted as Π -matrix, is defined as

$$\Pi = \begin{bmatrix} \pi_{11} & \dots & \pi_{1n} \\ \vdots & \ddots & \vdots \\ \pi_{n1} & \dots & \pi_{nn} \end{bmatrix}.$$

It has been shown in (Ray 2005 and Ray *et al.* 2005) that the measure $\mu_i \triangleq \mu(L(G_i))$ of the language $L(G_i)$, with the initial state q_i , can be expressed as $\mu_i = \sum_j \pi_{ij} \mu_j + \chi_i$ where $\chi_i \triangleq \chi(q_i)$. Equivalently, in vector notation

$$\mu = \Pi \mu + \chi \implies \mu = [\mathbb{I} - \Pi]^{-1} \chi, \quad (3)$$

where the measure vector $\mu \triangleq [\mu_1 \mu_2 \dots \mu_n]^T$ and the characteristic vector $\chi \triangleq [\chi_1 \chi_2 \dots \chi_n]^T$; and the condition $\sum_j \tilde{\pi}_{ij} < 1 \in i$ (see Definition 5) is sufficient for the inverse to exist.

Although the preceding analysis reported in (Ray 2005 and Ray *et al.* 2005) was intended for non-probabilistic regular languages, the event costs can be easily interpreted as occurrence probabilities. As such the $\tilde{\Pi}$ -matrix is analogous to the morph matrix of a Markov chain in the sense that an element $\tilde{\pi}_{ij}$ represents the probability of the j th event occurring at the i th state with the exception that the strict inequality condition $\sum_j \tilde{\pi}_{ij} < 1$ is enforced instead of satisfying the equality. Equivalently, the Π -matrix is analogous to the state transition probability matrix of a Markov chain in the sense that an element π_{jk} is analogous to the transition probability from state q_j to state q_k with the exception that the strict inequality condition $\sum_k \pi_{jk} < 1$ is enforced instead of satisfying the equality. It follows that the preceding analysis is applicable to the case of terminating probabilistic languages (Garg *et al.* 1992a, b) that have a strictly non-zero probability of termination at each state.

Let Σ^u denote the set of all unmodelled events in the terminating plant. A new unmarked absorbing state q_{n+1} , called the dump state (Ramadge and Wonham 1987), is created and the transition function δ is extended to $\delta_{\text{ext}} : (Q \cup \{q_{n+1}\}) \times (\Sigma \cup \Sigma^u) \rightarrow (Q \cup \{q_{n+1}\})$. The residue $\theta_j = 1 - \sum_k \pi_{jk}$ denotes the probability of transition from q_j to q_{n+1} . Consequently, the Π -matrix (see Definition 7) is augmented to obtain the stochastic state transition probability matrix as

$$\Pi_{\text{aug}} = \begin{bmatrix} \pi_{11} & \pi_{12} & \dots & \pi_{1n} & \theta_1 \\ \pi_{21} & \pi_{22} & \dots & \pi_{2n} & \theta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \pi_{n1} & \pi_{n2} & \dots & \pi_{nn} & \theta_n \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}. \quad (4)$$

Since the dump state q_{n+1} is not marked (Ramadge and Wonham 1987), it follows from Definition 4 that the corresponding state weight $\chi_{n+1} = 0$. Hence, the χ -vector is augmented as

$$\chi_{\text{aug}} = [\chi^T \ 0]^T. \quad (5)$$

Denoting $\Theta = [\theta_1 \ \theta_2 \ \dots \ \theta_n]^T$, where $\theta_i \in (0, 1)$ is the probability of transition from the state q_i to the dump state, it follows from equations (4) and (5) that the measure of the augmented system (Chattopadhyay and Ray 2006a) is

$$\mu_{\text{aug}}(\Theta) = [\mu(\Theta)^T \ 0]^T. \quad (6)$$

Then, the event cost matrix (see Definition 5) and the state transition cost matrix (see Definition 7) can be represented as

$$\tilde{\Pi}(\Theta) = \left[\mathbb{I} - \text{Diag}[\Theta] \right] \tilde{\mathbf{P}} \quad \text{and} \quad \Pi(\Theta) = \left[\mathbb{I} - \text{Diag}[\Theta] \right] \mathbf{P}, \quad (7)$$

where $\tilde{\mathbf{P}}$ and \mathbf{P} are both stochastic matrices (Bapat and Raghavan 1997), i.e., $\sum_j \tilde{\mathbf{P}}_{ij} = 1 \ \forall i \in \{1, \dots, m\}$ and $\sum_j \mathbf{P}_{ij} = 1 \ \forall i \in \{1, \dots, n\}$.

If the probability of termination (or equivalently the probability of transition to the dump state) is equal for all states, $q_i \in Q$, i.e., $\theta_i = \theta \ \forall i \in \{1, 2, \dots, n\}$, then equation (6) is expressed as

$$\mu_{\text{aug}}(\theta) = [\mu(\theta)^T \ 0]^T \quad (8)$$

Consequently, $\tilde{\Pi}$ and Π in equation (7) are represented as

$$\tilde{\Pi}(\theta) = (1 - \theta)\tilde{\mathbf{P}} \quad \text{and} \quad \Pi(\theta) = (1 - \theta)\mathbf{P} \quad (9)$$

where θ is the uniform probability of termination at all states; and both $\tilde{\mathbf{P}}$ and \mathbf{P} retain the properties of stochastic matrices (Bapat and Raghavan 1997).

2.2 Renormalization of language measure

The notion of language measure has been recently extended to non-terminating models by assuming a uniform non-zero probability of termination (θ) at each state, renormalizing the language measure vector with respect to the probability of termination and computing the limit of the renormalized measure (Chattopadhyay and Ray 2006a) as $\theta \rightarrow 0^+$. As the probability of termination approaches zero (i.e., $\theta \rightarrow 0^+$), and the plant coincides with the desired non-terminating model in the limit. The construction of renormalized measure is briefly outlined below.

The regular language generated by the *DFSA* under consideration is a sublanguage of the Kleene closure Σ^* of the alphabet Σ , for which the automaton states can be merged into a single state. In that case, the state transition cost matrix $\Pi(\theta)$ degenerates to the 1×1 matrix $[1 - \theta]$ and the normalized state weight

vector χ becomes one-dimensional and can be assigned as $\chi = 1$; consequently, the measure vector $\mu(\theta)$ degenerates to the scalar measure θ^{-1} . To alleviate the singularity of the matrix operator $[I - \Pi(\theta)]$ at $\theta = 0$, the measure vector $\mu(\theta)$ in (3) is normalized with respect to θ^{-1} to obtain the renormalized measure vector.

Definition 8: The renormalized measure is defined as

$$\mathbf{v}(\theta) = \theta \mu(\theta) = \theta [\mathbb{I} - \Pi(\theta)]^{-1} \chi \quad (10)$$

and it follows from (8) that

$$\theta \mu_{\text{aug}}(\theta) = [\mathbf{v}(\theta)^T \ 0]^T. \quad (11)$$

3. Optimal control problem: formulation

The following notations are needed for elementwise comparison of finite-dimensional vectors and matrices for the analysis developed in the sequel.

Notation 1: Let V^a and V^b be $(m \times n)$ real matrices. The following elementwise equality and inequalities imply that

$$\begin{aligned} \left(V^a \equiv_{\mathbf{E}} V^b \right) &\Leftrightarrow \left(V_{ij}^a = V_{ij}^b \right) \ \forall i \in \{1, \dots, n\} \ \forall j \in \{1, \dots, m\} \\ \left(V^a \neq_{\mathbf{E}} V^b \right) &\Leftrightarrow \left(V_{ij}^a \neq V_{ij}^b \right) \ \exists i \in \{1, \dots, n\}, j \in \{1, \dots, m\} \\ \left(V^a \geq_{\mathbf{E}} V^b \right) &\Leftrightarrow \left(V_{ij}^a \geq V_{ij}^b \right) \ \forall i \in \{1, \dots, n\} \ \forall j \in \{1, \dots, m\} \\ \left(V^a >_{\mathbf{E}} V^b \right) &\Leftrightarrow \left(V_{ij}^a > V_{ij}^b \right) \ \forall i \in \{1, \dots, n\} \ \forall j \in \{1, \dots, m\}. \end{aligned}$$

For the terminating plant, investigated in (Ray 2005 and Ray *et al.* 2005), the optimal supervisor selectively disables controllable transitions by setting their occurrence probabilities to zero. This implies that if Π^* and Π are the transition probability matrices for the optimally supervised plant and the unsupervised plant, respectively, then

$$\Pi^* \leq_{\mathbf{E}} \Pi, \quad \text{i.e.,} \quad \pi_{ij}^* \leq \pi_{ij}.$$

Since for any non-trivial supervisor, there is at least one disabled transition in the supervised plant, i.e.,

$$\exists i, j \text{ such that } \pi_{i,j} > 0 \quad \text{and} \quad \pi_{i,j}^* = 0$$

it follows that if the unsupervised plant is non-terminating, then any non-trivial supervision will result in a terminating model. The policy of Kumar and

Garg 2001 maintains the non-termination property by proportionately increasing the probabilities of the remaining enabled events at the state at which the disabling action is applied. The first issue here is that the supervisor must be able to affect the event occurrence probabilities, which is more than just inhibiting a transition. The second issue is that there is a possibility of disabling all events defined at a given state if all these events are controllable. In that case, the row sum cannot be maintained at 1 as it becomes strictly equal to zero. Thus, it is necessary to impose special constraints on the unsupervised plant to circumvent this situation. This paper investigates an alternative approach as described below.

Definition 9 (control philosophy): Disabling any transition σ at a given state q results in reconfiguration of the automaton structure as: Set the self-loop $\delta(q, \sigma) = q$ with the occurrence probability of σ from the state q remaining unchanged in the supervised and unsupervised plants.

This is equivalent to adding a self-loop to the state at which the event is being disabled, with the same occurrence probability as the disabled transition.

Proposition 1: For the control philosophy in Definition 9, a supervised plant is non-terminating if and only if the unsupervised plant is non-terminating.

Proof: The proof follows from two lemmas.

Lemma 1: Each row sum of the Π -matrix remains unchanged after supervisory actions for the control philosophy in Definition 9.

Proof: Let Π and Π^\dagger be the transition probability matrices for the unsupervised and supervised plants, respectively. Let there be exactly one disabled transition, in which a (controllable) event σ at state q_i is disabled and let the occurrence probability of σ at state q_i be $\tilde{\pi}$. If $\delta(q_i, \sigma) = q_k$, then it follows that

$$\Pi^\dagger = \Pi + \begin{matrix} & & & & \begin{matrix} \text{kth column} \\ \downarrow \end{matrix} \\ \begin{matrix} 0 & 0 & \dots & \dots & 0 & 0 \\ \vdots & \ddots & & & \vdots & \vdots \\ 0 & \dots & \tilde{\pi} & \dots & -\tilde{\pi} & 0 \\ \vdots & \dots & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & \vdots & & \ddots & \vdots \\ 0 & \dots & \dots & \dots & \dots & 0 \end{matrix} & \leftarrow \begin{matrix} \text{i th row} \end{matrix} \end{matrix}$$

implying $\sum_j \pi_{ij}^\dagger = \sum_j \pi_{ij} \forall i$. The proof follows by induction on the number of disabled events. \square

Lemma 2: Self-loops cannot be disabled.

Proof: For the control philosophy in Definition 9, disabling a self-loop at any given state causes regeneration of the self-loop at the same state with identical occurrence probability. \square

It is evident from the above two lemmas that each row sum of the reconfigured Π -matrix remains invariant. The proof of Proposition 1 is thus complete. \square

Remark 1: The control philosophy in Definition 9 is natural in the following sense. If $q_i \xrightarrow{\sigma} q_k$, and the controllable event σ is disabled at state q_i , then the sole effect of the supervisory action is to prevent the plant from making a transition to the state q_k . That is, the plant is forced to stay at the original state q_i and this is represented by the additional self-loop at state q_i instead of the original arc from q_i to q_k .

The notion of controllability is now clarified in the context of the present paper.

Definition 10 (controllable transitions): For a given plant, transitions that can be disabled in the sense of Definition 9 are defined to be controllable transitions in the sequel.

In view of Definition 10, controllability becomes state-based, i.e., transitions labelled by the same event may be controllable from one state and uncontrollable from some other state. This implies that the event alphabet Σ cannot be partitioned into uncontrollable and controllable events sets as proposed in Ramadge and Wonham (1987). Thus, a controllable transition $q_i \xrightarrow{\sigma} q_k$ refers to a triple $\{q_i, \sigma, q_k\}$ and the set of all such transitions is denoted by \mathcal{E} .

3.1 Model specification

Plant models considered in this paper are deterministic finite state automata (DFSA) with well-defined event occurrence probabilities. In other words, the occurrence of events is probabilistic, but the state at which the plant ends up, given a particular event has occurred, is deterministic. Furthermore, no emphasis is laid on the initial state of the plant and it is assumed that the plant may start from any state. Furthermore, having defined the characteristic state weight vector χ , it may not be necessary to specify the set of marked states, because if $\chi_i = 0$, then q_i is not marked and if $\chi_i \neq 0$, then q_i is marked. Therefore, plant models with an arbitrary uniform termination probability $\theta \in (0, 1)$, i.e., $\theta_i = \theta \forall i \in \{1, 2, \dots, n\}$, can be completely specified by a sextuple as

$$G(\theta) = (Q, \Sigma, \delta, \tilde{\Pi}(\theta), \chi, \mathcal{E}), \quad (12)$$

where $\tilde{\Pi}(\theta)_{ij}$ is the occurrence probability of event σ_j from state q_i and $\sum_j \tilde{\Pi}(\theta)_{ij} = 1 - \theta \forall i$. An application of (7) with uniform uniform termination probability θ yields an alternative representation of the sextuple in (12).

$$G(\theta) = (Q, \Sigma, \delta, (1 - \theta)\tilde{\mathbf{P}}, \chi, \mathcal{C}), \quad (13)$$

where $\tilde{\mathbf{P}}$ is the the morph matrix of the underlying Markov chain.

As $\theta \rightarrow 0^+$, the resulting non-terminating plant model is denoted as

$$G(0) = (Q, \Sigma, \delta, \tilde{\mathbf{P}}, \chi, \mathcal{C}). \quad (14)$$

Definition 11: Given $\theta \in (0, 1)$, a terminating plant $G(\theta) = (Q, \Sigma, \delta, (1 - \theta)\tilde{\mathbf{P}}, \chi, \mathcal{C})$ is defined to be the θ -neighbour of the non-terminating plant $G(0) = (Q, \Sigma, \delta, \tilde{\mathbf{P}}, \chi, \mathcal{C})$.

For a given non-terminating plant $G(0)$ and a fixed $\theta_0 \in (0, 1)$, there is exactly one θ_0 -neighbour $G(\theta_0)$.

Notation 2: Let $\theta \in (0, 1)$ be the uniform probability of termination for a terminating plant $G(\theta) = (Q, \Sigma, \delta, (1 - \theta)\tilde{\mathbf{P}}, \chi, \mathcal{C})$. Let \mathbf{P} be the state transition probability matrix of the underlying Markov chain, which is generated from δ and $\tilde{\mathbf{P}}$ (see equation (2)). Then, the (renormalized) language measure vector (see Definition 8) is obtained as

$$\mathbf{v}(\theta) = \theta \left[\mathbb{1} - (1 - \theta)\mathbf{P} \right]^{-1} \chi \quad (15)$$

where $(1 - \theta)\mathbf{P}$ is the sub-stochastic transition probability matrix for the terminating plant. Similarly, for a non-terminating plant $G(0) = (Q, \Sigma, \delta, \tilde{\mathbf{P}}, \chi, \mathcal{C})$ having the stochastic transition probability matrix \mathbf{P} , the (renormalized) measure vector (Chattopadhyay and Ray 2006a) is denoted as

$$\mathbf{v}(0) = \lim_{\theta \rightarrow 0^+} \mathbf{v}(\theta) = \lim_{\theta \rightarrow 0^+} \theta \left[\mathbb{1} - (1 - \theta)\mathbf{P} \right]^{-1} \chi \quad (16)$$

In the sequel, renormalized measure \mathbf{v} in equations (10) and (11) is referred to as measure for brevity.

3.2 Construction of an optimal supervisor

A supervisor disables a subset of the set \mathcal{C} of controllable transitions and hence there is a bijection between the set of all possible supervision policies and the power set $2^{\mathcal{C}}$. That is, there exists $2^{|\mathcal{C}|}$ possible supervisors and each supervisor is uniquely identifiable with a subset of \mathcal{C} and the language measure \mathbf{v} allows a quantitative comparison of different supervision policies.

Definition 12: For an unsupervised (non-terminating) plant $G(0) = (Q, \Sigma, \delta, \tilde{\mathbf{P}}, \chi, \mathcal{C})$, let G^\dagger and G^\ddagger be the supervised plants with sets of disabled transitions, $\mathcal{D}^\dagger \subseteq \mathcal{C}$ and $\mathcal{D}^\ddagger \subseteq \mathcal{C}$, respectively, whose measures are \mathbf{v}^\dagger and \mathbf{v}^\ddagger . Then, the supervisor that disables \mathcal{D}^\dagger is defined to be superior to the supervisor that disables \mathcal{D}^\ddagger if $\mathbf{v}^\dagger \geq_{\mathbf{E}} \mathbf{v}^\ddagger$ and strictly superior if $\mathbf{v}^\dagger >_{\mathbf{E}} \mathbf{v}^\ddagger$.

Definition 13 (Optimal supervision problem): Given a (non-terminating) plant $G(0) = (Q, \Sigma, \delta, \tilde{\mathbf{P}}, \chi, \mathcal{C})$, the problem is to compute a supervisor that disables a subset $\mathcal{D}^* \subseteq \mathcal{C}$, such that

$$\mathbf{v}^* \geq_{\mathbf{E}} \mathbf{v}^\dagger \quad \forall \mathcal{D}^\dagger \subseteq \mathcal{C}$$

where \mathbf{v}^* and \mathbf{v}^\dagger are the measure vectors of the supervised plants G^* and G^\dagger under \mathcal{D}^* and \mathcal{D}^\dagger , respectively.

Remark 2: For a non-trivial plant $G(0) = (Q, \Sigma, \delta, \tilde{\mathbf{P}}, \chi, \mathcal{C})$ (i.e., $|Q| > 1$), there may exist two supervisors that are not comparable in the sense of Definition 12. For example, given a two-state unsupervised plant G , if G^\dagger and G^\ddagger are supervised plants under two different supervisors with disabled transition sets, \mathcal{D}^\dagger and \mathcal{D}^\ddagger , respectively, then the following situation may occur for the indices $i \neq j$.

$$\left(v_i^\dagger > v_i^\ddagger \right) \wedge \left(v_j^\dagger < v_j^\ddagger \right),$$

where v_i^\dagger and v_i^\ddagger are the i th elements of the measure vectors for G^\dagger and G^\ddagger , respectively. It is shown in the next section that, for a given plant, an optimal supervisor (in the sense of Definition 13) does exist for which the measure vector is elementwise greater than or equal to the measure vector of the plant under any other supervision policy.

Terminating plant models have sub-stochastic transition probability matrices (see Definition 7). By postulating the existence of unmodelled transitions, such plants can be transformed to non-terminating models as explained below. For uniform termination probability $\theta \in (0, 1)$, equations (8) and (11) suggest the possibility of computing optimal supervision policies for terminating plants based on the analysis of non-terminating plants.

4. Optimal control problem: solution

This section presents a solution to the optimal supervision problem by assuming a uniform non-zero probability of termination, θ , at each state. Then, it is shown that the solution for the corresponding non-terminating plant can be obtained from the control policy of the terminating plant and the bounds on the derivatives of the language measure (see Appendix A).

Let $\theta \in (0, 1)$ be the uniform termination probability of an unsupervised plant $G(\theta) = (Q, \Sigma, \delta, (1 - \theta)\tilde{\mathbf{P}}, \boldsymbol{\chi}, \mathcal{C})$. The resulting (substochastic) state transition cost matrix is $\mathbf{\Pi}(\theta) = (1 - \theta)\mathbf{P}$. For such plants with uniform non-zero termination probability, the following lemma states existence of an augmented plant model.

Lemma 3: For the terminating plant $G(\theta) = (Q, \Sigma, \delta, (1 - \theta)\tilde{\mathbf{P}}, \boldsymbol{\chi}, \mathcal{C})$, let the corresponding augmented non-terminating plant be $G_{\text{aug}} = (Q_{\text{aug}}, \Sigma_{\text{aug}}, \delta_{\text{aug}}, \tilde{\mathbf{P}}_{\text{aug}}, \boldsymbol{\chi}_{\text{aug}}, \mathcal{C})$. Let $\mathbf{v}^*(\theta)$ and $\mathbf{v}^\dagger(\theta)$ be the measures of the terminating plant with the respective sets of disabled transitions $\mathcal{D}^* \subseteq \mathcal{C}$ and $\mathcal{D}^\dagger \subseteq \mathcal{C}$. Then,

$$\exists \mathcal{D}^* \subseteq \mathcal{C} \quad \text{s.t.} \quad \mathbf{v}^*(\theta) \succeq_{\mathbf{E}} \mathbf{v}^\dagger(\theta) \quad \forall \mathcal{D}^\dagger \subseteq \mathcal{C} \quad \forall \theta \in (0, 1) \quad (17)$$

which implies that an optimal supervisor for G_{aug} exists (in the sense of Definition 13) which disables $\mathcal{D}^* \subseteq \mathcal{C}$.

Proof: The first n elements of the measure vectors of the augmented plant and the unaugmented plant are identically equal as seen in equation (11). Then, the proof follows from Definition 12. \square

The remainder of this section derives an algorithm for a supervision policy that elementwise maximizes the measure of the terminating plant $G(\theta)$. Lemma 3 guarantees that the optimal policy is based on a non-terminating plant.

Proposition 2 (Monotonicity): Let $\mathbf{\Pi}(\theta)$ and $\mathbf{v}(\theta)$ be the state transition cost matrix and the measure vector of an unsupervised plant $G(\theta) = (Q, \Sigma, \delta, (1 - \theta)\tilde{\mathbf{P}}, \boldsymbol{\chi}, \mathcal{C})$, respectively. Let a supervisor be constructed to reconfigure the plant by disabling a set of controllable transitions $\mathcal{D}^\dagger \subseteq \mathcal{C}$ such that $\mathbf{\Pi}$ is modified to $\mathbf{\Pi}^\dagger$ by following Algorithm 1. Then, denoting the measure vector of the supervised plant by \mathbf{v}^\dagger , it follows that $\mathbf{v}^\dagger \succeq_{\mathbf{E}} \mathbf{v}$; and equality holds if and only if $\mathbf{\Pi}^\dagger = \mathbf{\Pi}$.

Algorithm 1: Monotonic Increase of Measure \mathbf{v}

```

begin
  for i = 1 to n do
    for j = 1 to n do
       $\left. \begin{array}{l} \Pi_{ij}^\dagger = \Pi_{ij} + \beta_{ij} \\ \Pi_{ii}^\dagger = \Pi_{ii} - \beta_{ij} \end{array} \right\} \text{if } v_j > v_i \text{ with } \beta_{ij} > 0$ 
       $\left. \begin{array}{l} \Pi_{ij}^\dagger = \Pi_{ij} \\ \Pi_{ii}^\dagger = \Pi_{ii} \end{array} \right\} \text{if } v_j = v_i$ 
       $\left. \begin{array}{l} \Pi_{ij}^\dagger = \Pi_{ij} - \beta_{ij} \\ \Pi_{ii}^\dagger = \Pi_{ii} + \beta_{ij} \end{array} \right\} \text{if } v_j < v_i \text{ with } \beta_{ij} > 0$ 
    endfor
  endfor
end

```

Proof: It follows from equation (15) in Notation 2 that

$$\begin{aligned} \mathbf{v}^\dagger - \mathbf{v} &= \theta[\mathbf{0} - \mathbf{\Pi}^\dagger]^{-1} - \theta[\mathbf{0} - \mathbf{\Pi}]^{-1} \boldsymbol{\chi} \\ &= \theta[\mathbf{0} - \mathbf{\Pi}^\dagger]^{-1} ([\mathbf{0} - \mathbf{\Pi}] - [\mathbf{0} - \mathbf{\Pi}^\dagger]) [\mathbf{0} - \mathbf{\Pi}]^{-1} \boldsymbol{\chi} \\ &= \theta[\mathbf{0} - \mathbf{\Pi}^\dagger]^{-1} (\mathbf{\Pi}^\dagger - \mathbf{\Pi}) \mathbf{v}. \end{aligned}$$

Defining the matrix $\Delta \triangleq \mathbf{\Pi}^\dagger - \mathbf{\Pi}$, and the i th row of Δ as Δ_i , it follows that

$$\Delta_i^T \mathbf{v} = \sum_j \Delta_{ij} v_j = \sum_j \theta \beta_{ij} \Gamma_{ij} \quad (18)$$

where

$$\Gamma_{ij} = \begin{cases} (v_i - v_j) & \text{if } v_i > v_j \\ 0 & \text{if } v_i = v_j \implies \Gamma_{ij} \geq 0 \quad \forall i, j. \\ (v_j - v_i) & \text{if } v_i < v_j \end{cases}$$

Since $\sum_{i=1}^n \mathbf{\Pi}_{ij} = \sum_{i=1}^n \mathbf{\Pi}_{ij}^\dagger$, $\forall j, k$, it follows from non-negativity of $\mathbf{\Pi}$, that $[\mathbf{I} - \mathbf{\Pi}^\dagger]^{-1} >_{\mathbf{E}} \mathbf{0}$. Since $\beta_i \geq 0 \quad \forall i$, it follows that $\Delta_i^T \mathbf{v} \geq 0 \quad \forall i \implies \mathbf{v}^\dagger \succeq_{\mathbf{E}} \mathbf{v}$. For $v_j \neq 0$ and Δ as defined above, $\Delta_i^T \mathbf{v}^k = 0$ if and only if $\Delta = \mathbf{0}$. Then, $\mathbf{\Pi}^\dagger = \mathbf{\Pi}$ and $\mathbf{v}^\dagger = \mathbf{v}$. \square

Corollary 1: Under an identical situation to that assumed in the statement of Proposition 2, let the plant be reconfigured as given in Algorithm 2. Then, denoting the measure vector of the modified plant by \mathbf{v}^\dagger , it follows that $\mathbf{v}^\dagger \preceq_{\mathbf{E}} \mathbf{v}$; and equality holds if and only if $\mathbf{\Pi}^\dagger = \mathbf{\Pi}$.

Algorithm 2: Monotonic Decrease of Measure \mathbf{v}

```

begin
  for i = 1 to n do
    for j = 1 to n do
       $\left. \begin{array}{l} \Pi_{ij}^\dagger = \Pi_{ij} + \beta_{ij} \\ \Pi_{ii}^\dagger = \Pi_{ii} - \beta_{ij} \end{array} \right\} \text{if } v_j < v_i \text{ with } \beta_{ij} > 0$ 
       $\left. \begin{array}{l} \Pi_{ij}^\dagger = \Pi_{ij} \\ \Pi_{ii}^\dagger = \Pi_{ii} \end{array} \right\} \text{if } v_j = v_i$ 
       $\left. \begin{array}{l} \Pi_{ij}^\dagger = \Pi_{ij} - \beta_{ij} \\ \Pi_{ii}^\dagger = \Pi_{ii} + \beta_{ij} \end{array} \right\} \text{if } v_j > v_i \text{ with } \beta_{ij} > 0$ 
    endfor
  endfor
end

```

Proof: The proof is similar to that of Proposition 26. \square

Proposition 2 facilitates formulation of the algorithm for computing an optimal supervisor for plants with

uniform non-zero probability of termination at each state. Let the k th iteration of the algorithm compute a set $\mathcal{Q}^{[k]} \subseteq \mathcal{C}$ of controllable transitions to be disabled in the sense of the control philosophy in Definition 9. The language measure vector computed in the k th iteration of the algorithm is denoted by $\mathbf{v}^{[k]}$. The algorithm terminates at the $(k+1)$ th iteration if $\mathcal{Q}^{[k]} = \mathcal{Q}^{[k+1]}$, which is the optimal set of disabled transitions computed by the algorithm and is denoted by \mathcal{Q}^* . The algorithm is started with the unsupervised plant (i.e., with all controllable transitions enabled) and hence $\mathcal{Q}^{[0]} = \emptyset$. A formal description is given in Algorithm 3.

Algorithm 3: Optimal Supervisor Algorithm

```

input :  $\Pi(\theta) = (1 - \theta)\mathbf{P}, \chi, \mathcal{C}$ 
output: Optimal set of disabled transitions  $\mathcal{Q}^*$ 
1 begin
2   Set  $\mathcal{Q}^{[0]} = \emptyset$ ;           /* Initial disabling set */
3   Set  $k = 1$ ;
4   Set Terminate = false;
5   while (Terminate == false) do
6     Compute  $\mathbf{v}^{[k]}$ ;
7     for  $j = 1$  to  $n$  do
8       for  $k = 1$  to  $n$  do
9         Disable all controllable transitions  $q_k \xrightarrow{\sigma} q_j$ 
          such that  $v_j < v_k$ ;
10        Enable all controllable transitions  $q_k \xrightarrow{\sigma} q_j$ 
          such that  $v_j \geq v_k$ ;
11      endfor
12    endfor
13    Compute  $\mathcal{Q}^{[k]}$ ;           /*  $k$ th disabling set */
14    if  $\mathcal{Q}^{[k]} == \mathcal{Q}^{[k-1]}$  then
15      | Terminate = true;
16    else
17      |  $k = k + 1$ ;
18    endif
19  endw
20   $\mathcal{Q}^* = \mathcal{Q}^{[k]}$ ;           /* Optimal disabling set */
21 end

```

Proposition 3: Let $\mathbf{v}^{[k]}$ be the language measure vector computed in the k th iteration of Algorithm 3. The measure vectors computed by the algorithm form an elementwise non-decreasing sequence, i.e., $\mathbf{v}^{[k+1]} \geq_{\mathbf{E}} \mathbf{v}^{[k]} \forall k$.

Proof: Let the state transition probability matrix in the k th iteration of Algorithm 3 be denoted by $\Pi^{[k]}$. Then, the matrix $\Pi^{[k+1]}$ is generated from $\Pi^{[k]}$ by following the procedure as described in Proposition 2. Hence, $\mathbf{v}^{[k+1]} \geq_{\mathbf{E}} \mathbf{v}^{[k]}$. \square

Proposition 4 (effectiveness): Algorithm 3 is an effective procedure (Hopcroft *et al.* 2001), i.e., it is guaranteed to terminate.

Proof: Let $G(\theta) = (Q, \Sigma, \delta, (1 - \theta)\tilde{\mathbf{P}}, \chi, \mathcal{C})$ be the unsupervised plant and let $\text{Card}(\mathcal{C}) = \ell \in \mathbb{N}$. Denoting the

set of all permutations of the vector $[1 \ 2 \ \dots \ \ell]^T$ by $\mathcal{P}(\ell)$, a function $\xi: 2^{\mathcal{C}} \rightarrow \mathcal{P}(\ell)$ is defined as

1. $\forall \mathcal{Q}^{\dagger} \subseteq \mathcal{C}, (v_{i_1}^{\dagger} > v_{i_2}^{\dagger} > \dots > v_{i_n}^{\dagger})$
 $\implies (\xi(\mathcal{Q}^{\dagger}) = [i_1 \ i_2 \ \dots \ i_n]^T)$
2. $(v_{i_s}^{\dagger} = v_{i_t}^{\dagger}) \wedge (i_s > i_t)$
 $\implies (\text{if } \xi(\mathcal{Q}^{\dagger})_s = i_s \text{ and } \xi(\mathcal{Q}^{\dagger})_t = i_t \text{ then } s > t).$

Let $\mathbf{v}^{[k]}$ be the measure vector computed in the k th iteration of Algorithm 3. Then, $\mathbf{v}^{[k]} = \mathbf{v}^{[k+1]}$ implies that Algorithm 3 terminates in $k+1$ iterations according to its stopping rule.

Next let $\mathcal{Q}^{[k_1]}$ and $\mathcal{Q}^{[k_2]}$ be the disabling sets in iterations k_1 and k_2 , respectively. If $\xi(\mathcal{Q}^{[k_1]}) = \xi(\mathcal{Q}^{[k_2]})$, then $\mathbf{v}^{[k_1+1]} = \mathbf{v}^{[k_2+1]}$. Since $\xi(\mathcal{Q}^{[k_1]}) = \xi(\mathcal{Q}^{[k_2]})$, it follows from the definition of ξ that if $v_i^{[k_1]} > v_j^{[k_1]}$, then $v_i^{[k_2]} \geq v_j^{[k_2]}$. If $v_i^{[k_2]} > v_j^{[k_2]}$ then controllable transitions $q_i \rightarrow_{\sigma} q_j$ are disabled in both iterations k_1+1 and k_2+1 . If $v_i^{[k_1]} = v_j^{[k_1]}$, then disabling or enabling controllable transitions $q_i \rightarrow_{\sigma} q_j$ does not affect the measure vector. Hence, it follows that $\mathbf{v}^{[k_1+1]}$ and $\mathbf{v}^{[k_2+1]}$ can be obtained by disabling the same set of controllable transitions, thus implying $\mathbf{v}^{[k_1+1]} = \mathbf{v}^{[k_2+1]}$. Since the measure vectors can repeat only at the final iteration, Algorithm 3 is guaranteed to terminate within $\text{Card}(\mathcal{P}(\ell)) = \ell!$ iterations. Therefore, effectiveness of Algorithm 3 is established. \square

Next it is established that Algorithm 3 is correct in the sense that an optimal supervision policy is generated.

Proposition 5 (Optimality): For a terminating plant $G(\theta) = (Q, \Sigma, \delta, (1 - \theta)\tilde{\mathbf{P}}, \chi, \mathcal{C})$, the supervision policy computed by Algorithm 3 is optimal in the sense of Definition 13.

Proof: Let $G(\theta)$ have the state transition cost matrix Π , measure $\mathbf{v}^{[0]}$, no disabled events, i.e., $\mathcal{Q}^0 = \emptyset$. Let $G(\theta)$ be configured as the supervised plant $G^*(\theta)$ by application of Algorithm 3 when it stops.

Let G^{\dagger} be another configured plant distinct from G^* . Let $\mathcal{Q}^* \subseteq \mathcal{C}$ and $\mathcal{Q}^{\dagger} \subseteq \mathcal{C}$ be the respective sets of disabled transitions and \mathbf{v}^* and \mathbf{v}^{\dagger} be the respective measures for G^* and G^{\dagger} ; and $\mathcal{Q}^* \neq \mathcal{Q}^{\dagger}$.

Let the following set differences be denoted as: $\Delta\mathcal{Q} \triangleq \mathcal{Q}^* \setminus \mathcal{Q}^{\dagger}$ and $\nabla\mathcal{Q} \triangleq \mathcal{Q}^{\dagger} \setminus \mathcal{Q}^*$. An application of Algorithm 3 yields

- $\forall i, j \ v_i^* > v_j^* \implies$ all controllable transitions $q_i \xrightarrow{\sigma} q_j$ are disabled.
- $\forall i, j \ v_i^* \leq v_j^* \implies$ all controllable transitions $q_i \xrightarrow{\sigma} q_j$ are enabled.

To change the plant configuration from G^* to G^\dagger , all transitions in $\Delta\mathcal{Q}$ are enabled and all transitions in $\nabla\mathcal{Q}$ are disabled. Since any such change requires us to either disable a transition $q_i \rightarrow_\sigma q_j$ where $v_i^* \leq v_j^*$ or enable a disabled transition $q_i \rightarrow_\sigma q_j$ where $v_i^* > v_j^*$, it follows from Corollary 1 that $\mathbf{v}^\dagger \preceq_{\mathbf{E}} \mathbf{v}^*$.

Since G^\dagger is an arbitrary configuration distinct from G^* , it follows that G^* is an optimal supervision policy in the sense of Definition 13. \square

In the reported work on discrete event control of non-probabilistic regular languages (e.g., (Ramadge and Wonham 1987)), the emphasis is on computing the maximally permissive supervisor in the sense that the supervised plant language is the supremal controllable sub-language of the specification. A similar approach is taken for probabilistic regular languages (Garg 1992a, b). In contrast, the measure-theoretic concept in this paper computes a policy that maximizes the elements of the language measure vector elementwise to find a supervisor with maximal performance. Proposition 5 shows that there exists at least one optimal supervisor. Now it is shown that the optimal supervisor computed by Algorithm 3 is unique in the sense of being maximally permissive among all policies that guarantee optimal performance of the supervised plant.

Proposition 6 (uniqueness): Given an unsupervised plant $G(\theta)$, the optimal supervisor $G^*(\theta)$, computed by Algorithm 3, is unique in the sense that it is maximally permissive among all possible supervision policies with optimal performance. That is, if \mathcal{D}^* and \mathcal{D}^\dagger are the disabled transition sets, and \mathbf{v}^* and \mathbf{v}^\dagger are the language measure vectors for G^* and an arbitrarily supervised plant G^\dagger , respectively, then

$$\mathbf{v}^* \preceq_{\mathbf{E}} \mathbf{v}^\dagger \implies \mathcal{D}^* \subset \mathcal{D}^\dagger \subseteq \mathcal{C}. \quad (19)$$

Proof: If G^* and G^\dagger are distinct, then $\mathcal{D}^\# \neq \mathcal{D}^*$. Given $\mathbf{v}^* \preceq_{\mathbf{E}} \mathbf{v}^\dagger$, let G^* be reconfigured to G^\dagger by disabling and/or re-enabling appropriate controllable transitions. It follows from equation (18) that

$$\begin{aligned} 0 &= \mathbf{v}^\dagger - \mathbf{v}^* = [\mathbf{I} - \mathbf{\Pi}^\dagger]^{-1} (\mathbf{\Pi}^\dagger - \mathbf{\Pi}^*) \mathbf{v}^* \\ &\implies (\mathbf{\Pi}^\dagger - \mathbf{\Pi}^*) \mathbf{v}^* = 0. \end{aligned} \quad (20)$$

The i th element of $(\mathbf{\Pi}^\dagger - \mathbf{\Pi}^*) \mathbf{v}^*$ is expressed as the finite sum of real numbers

$$0 = \left((\mathbf{\Pi}^\dagger - \mathbf{\Pi}^*) \mathbf{v}^* \right)_i = \sum_{r=1}^{\tau} T_r^i, \quad (21)$$

where $0 \leq \tau \leq 2^{\text{Card}(\mathcal{C})}$ and each T_r^i is of the form:

$$T_r^i = \begin{cases} \alpha_r^i (v_i^* - v_j^*) > 0, & \text{if } T_r^i \text{ arises due to disabling} \\ & q_i \xrightarrow{\sigma} q_j \text{ for some } q_j \in \mathcal{Q} \\ \alpha_r^i (v_j^* - v_i^*) \geq 0, & \text{if } T_r^i \text{ arises due to enabling} \\ & q_i \xrightarrow{\sigma} q_j \text{ for some } q_j \in \mathcal{Q} \end{cases} \quad (22)$$

because each α_r^i represents event occurrence probabilities and hence are positive, and the logic of disabling and re-enabling follows Algorithm 3. Therefore, it follows from equation (22) that $T_r^i = 0 \forall r \in \{1, \dots, \tau\}$.

Hence, it is necessary to re-enable controllable transitions $q_i \rightarrow_\sigma q_j$ and disable the self loop at q_i such that $v_i^\dagger = v_j^\dagger$ for reconfiguration from G_θ^\dagger to G_θ^* . Note that all such transitions are guaranteed to be enabled in G_θ^* (see line 10 in Algorithm 3). Therefore, given $\mathbf{v}^* \preceq_{\mathbf{E}} \mathbf{v}^\dagger$, it follows that $\mathcal{D}^* \subseteq \mathcal{D}^\dagger$. That is, $G^*(\theta)$ is unique for all $\theta \in (0, 1)$ in the sense that the configured plant is maximally permissive among all other configurations that yield the same optimal measure $\mathbf{v}^*(\theta)$. \square

4.1 Optimal control of non-terminating plants

This section presents the optimal supervision problem for non-terminating plants (i.e., with termination probability $\theta=0$ at each state) having the structure $G(0) = (\mathcal{Q}, \Sigma, \delta, \mathbf{P}, \boldsymbol{\chi}, \mathcal{C})$ and the corresponding stochastic transition probability matrix is \mathbf{P} . The rationale for working on a terminating plant, instead of the non-terminating plant is explained below.

By maximizing the measure $\mathbf{v}(\theta)$ for a given $\theta \in (0, 1)$, an optimal control law can be derived based on the state transition cost matrix $\mathbf{\Pi}(\theta) = (1 - \theta)\mathbf{P}$ of the supervised plant language and the originally assigned $\boldsymbol{\chi}$ -vector. Such an optimal control law is sought to be θ -independent in the sense of having the same disabling set $\mathcal{D} \subseteq \mathcal{C}$ for a given range of θ , where θ might be restricted to be not too far away from 0^+ . On the other hand, from the perspective of numerical stability and accuracy in computation of $\mathbf{v}(\theta)$ (see Definition 8), it is desirable to have a relatively large positive value of θ . The results derived in this section serve toward establishing upper bounds on θ for which the optimal control law should be θ -independent and the associated computation is numerically stable. The main objective is summarized below.

A uniform non-zero probability of termination $\theta_* \in (0, 1)$ is to be computed such that the terminating plant $G(\theta_*)$ and the

non-terminating plant $G(0)$ shall have the same the disabling set $\mathcal{D} \subseteq \mathcal{C}$. However, in general, their measures could be different, i.e., $\nu(\theta_*) \neq_{\mathbb{E}} \nu(0)$.

Proposition 7: Let $(1 - \theta)\mathbf{P}$ and $\nu(\theta)$ be the state transition cost matrix and the measure of the plant $G(\theta) = (Q, \Sigma, \delta, (1 - \theta)\tilde{\mathbf{P}}, \chi, \mathcal{C})$. Then, for all $q_i, q_j \in Q$, there exists $\theta_{ij}^* \in (0, 1]$ such that $\forall \theta \in (0, \theta_{ij}^*)$, the sign of $(\nu_i(\theta) - \nu_j(\theta))$ is fixed (i.e., positive, negative or zero); and θ_{ij}^* can be computed as an explicit function of the stochastic matrix \mathbf{P} and state characteristic vector χ .

Proof: Let $\gamma_{ij}(\theta) \triangleq \nu_i(\theta) - \nu_j(\theta) \forall \theta \in (0, 1)$, which is a smooth function of θ , and $\gamma_{ij}(0) = \lim_{\theta \rightarrow 0^+} \gamma_{ij}(\theta)$. The proof is based on the following two cases.

Case 1: No sign change of $\gamma_{ij}(\theta)$ in $(0, 1) \Rightarrow \theta_{ij}^* = 1$. This includes: $\gamma_{ij}(0) = 0$ and $(d^k \gamma_{ij}(\theta)/d\theta^k)|_{\theta=0} = 0$ for all $k \geq 0$ because $\gamma_{ij}(\theta) = 0 \forall \theta \in (0, 1)$ by Proposition A.3.

Case 2: $\gamma_{ij}(\theta)$ changes sign in $(0, 1)$; and $(\partial^r \gamma_{ij}(\theta)/\partial \theta^r)|_{\theta=0} = \lambda_{ij} \neq 0$ for some integer $r \geq 0$.

If $r=0$, there exists $\tau_1 \in (0, 1)$ such that $\gamma_{ij}(\tau_1) = 0$ for the first time. If $r > 0$, it is possible that $\gamma_{ij}(0) = 0$. Then, as θ is increased from zero, $\gamma_{ij}(\theta)$ becomes non-zero and there exists $\tau_1 \in (0, 1)$ such that $\gamma_{ij}(\tau_1) = 0$ again. Smoothness of $\gamma_{ij}(\theta)$ necessitates that $(\partial^r \gamma_{ij}(\theta)/\partial \theta^r)|_{\theta=\theta_{ij}^*} = 0$ for some $\theta_{ij}^* \in (0, \tau_1)$. Then, it follows from the Mean value Theorem that there exists $\tau_2 \in (0, \theta_{ij}^*)$ such that

$$\frac{\partial^{r+1} \gamma_{ij}(\theta)}{\partial \theta^{r+1}} \Big|_{\theta=\tau_2} = \frac{\lambda_{ij}}{\theta_{ij}^*}$$

for the given $r \geq 0$, Proposition A.2, triangular inequality, and the relation $\gamma_{ij}(\theta) = \nu_i(\theta) - \nu_j(\theta)$ yield

$$\begin{aligned} \theta_{ij}^* &= \frac{\left| (\partial^r \nu_i(\theta)/\partial \theta^r) \Big|_{\theta=0} - (\partial^r \nu_j(\theta)/\partial \theta^r) \Big|_{\theta=0} \right|}{(r+1)! 2^{r+3} \left(\inf_{\alpha \neq 0} \left\| \left[\mathbb{I} - P + \alpha \mathcal{P} \right]^{-1} \right\|_{\infty} \right)^{r+1}} \\ &= \begin{cases} \frac{\left| \frac{\{ [\mathbb{I} - P + \mathcal{P}]^{-1} \chi - \mathcal{P} \chi \}_i - \{ [\mathbb{I} - P + \mathcal{P}]^{-1} \chi - \mathcal{P} \chi \}_j \}}{8 \times (\inf_{\alpha \neq 0} \left\| [\mathbb{I} - P + \alpha \mathcal{P}]^{-1} \right\|_{\infty})} \right|, & \text{if } r = 0 \\ \frac{\left| \frac{\{ [\mathbb{I} - P + \mathcal{P}]^{-1} [\mathbb{I} - [\mathbb{I} - P + \mathcal{P}]^{-1}]^r \chi \}_i}{- \{ [\mathbb{I} - P + \mathcal{P}]^{-1} [\mathbb{I} - [\mathbb{I} - P + \mathcal{P}]^{-1}]^r \chi \}_j} \right|}{2^{r-3} \left(\inf_{\alpha \neq 0} \left\| [\mathbb{I} - P + \alpha \mathcal{P}]^{-1} \right\|_{\infty} \right)^{r+1}}, & \text{if } r > 0. \end{cases} \end{aligned} \quad (23)$$

□

Remark 3: For a non-terminating plant $G(0) = (Q, \Sigma, \delta, \tilde{\mathbf{P}}, \chi)$, let $\theta_* = \min_{i,j} \theta_{ij}^*$. Then, the plant configuration obtained by applying a single iteration of Algorithm 3 to the θ -parameterized plant $G(\theta) = (Q, \Sigma, \delta, (1 - \theta)\tilde{\mathbf{P}}, \chi, \mathcal{C})$ is identical for all $\theta \leq \theta_*$.

The procedure of computing θ_* is summarized as Algorithm 4.

Proposition 8: Complexity of computing a positive bound for θ_* is $O(n^3)$ where n is the number of plant states.

Algorithm 4: Computation of the Bound θ_*

```

input : P, χ
output: θ*
1 begin
2   Set θ* = 1;
3   Set θcurr = 0;
4   Compute ℘; /* See Algorithm 6 in Appendix A
   */
5   Compute M0 = [I - P + ℘]-1;
6   Compute M1 = [I - [I - P + ℘]-1];
7   Compute M2 = infα≠0 || [I - P + α℘]-1 ||∞;
8   for j = 1 to n do
9     for i = 1 to n do
10      if (℘χ)i - (℘χ)j ≠ 0 then
11        | θcurr = 1/(8M2) | (℘χ)i - (℘χ)j |
12      else
13        for r = 0 to n do
14          if (M0χ)i ≠ (M0χ)j then
15            | Break;
16          else
17            if (M0M1rχ)i ≠ (M0M1rχ)j then
18              | Break;
19            endif
20          endif
21        endif
22      if r = 0 then
23        | θcurr = |(M0 - ℘)χi - (M0 - ℘)χj| / (8M2);
24      else
25        if r > 0 AND r ≤ n then
26          | θcurr = |(M0M1rχ)i - (M0M1rχ)j| / (2r+3M2);
27        else
28          | θcurr = 1;
29        endif
30      endif
31    endif
32    θ* = min(θ*, θcurr);
33  endif
34 endfor
35 end

```

Proof: Referring to Algorithm 4, the part within the nested For loops (lines 10 to 32) is executed at most n^2 times and each iteration involves only single-iteration scalar operations. Thus the computational complexity of this part is of the order of $O(n^2)$. Lines 5 and 6 involve inversion of $n \times n$ dimensional non-singular matrices and hence the complexity of execution is of the order of $O(n^3)$. Proposition G (see Appendix) guarantees that the complexity of computing \mathcal{P} is, in general, of the order of $O(n^3)$. Line 7, which computes $M_2 = \inf_{\alpha \neq 0} \|\llbracket \mathbb{I} - \mathbf{P} + \alpha_0 \mathcal{P} \rrbracket^{-1}\|_\infty$, is a search problem. However, since M_2 appears only in the denominator of the expressions for θ_{curr} , it follows that, if for some $\alpha = \alpha_0 \neq 0$ and by using

$$M_2 = \left\| \llbracket \mathbb{I} - \mathbf{P} + \alpha_0 \mathcal{P} \rrbracket^{-1} \right\|_\infty \quad (24)$$

it is possible to obtain a positive lower bound of θ_* in Algorithm 4. Since the computation of $\|\llbracket \mathbb{I} - \mathbf{P} + \alpha_0 \mathcal{P} \rrbracket^{-1}\|_\infty$ is of the order of $O(n^3)$ due to the matrix inversion, it is concluded that a positive lower bound of θ_* can be computed with a complexity of $O(n^3)$. \square

Remark 4: It is shown in (Chattopadhyay and Ray 2006a) that for any stochastic matrix \mathbf{P}

$$\begin{aligned} \llbracket \mathbb{I} - \mathbf{P} + \alpha \mathcal{P} \rrbracket^{-1} &= \llbracket \mathbb{I} - \mathbf{P} + \mathcal{P} \rrbracket^{-1} \\ &+ \left(\frac{1-\alpha}{\alpha} \right) \mathcal{P} \quad \forall \alpha \neq 0 \\ \implies \left(\llbracket \mathbb{I} - \mathbf{P} + \mathcal{P} \rrbracket^{-1} - \mathcal{P} \right) &+ \left(\frac{1}{\alpha} \right) \mathcal{P}. \quad (25) \end{aligned}$$

Using $M_2 = \|\llbracket \mathbb{I} - \mathbf{P} + \mathcal{P} \rrbracket^{-1}\|_\infty$ instead of $M_2 = \inf_{\alpha \neq 0} \|\llbracket \mathbb{I} - \mathbf{P} + \alpha \mathcal{P} \rrbracket^{-1}\|_\infty$ (i.e., using $\alpha = 1$) in Algorithm 4 yields a value which satisfies the requirement stated in Remark 3 and therefore qualifies as θ_* . Thus, the major advantage of this approximation is having significantly smaller computational complexity because the search involved in computing the infimum is avoided at the cost of using a smaller value of θ_* , which may make subsequent computation of measure slightly more difficult due to possible ill-conditioning (see Definition 8).

On account of Proposition 7 and Remark 3, Algorithm 3 is modified to solve the optimal supervision problem for non-terminating plants and the modified version is formally presented in Algorithm 5.

Algorithm 5: Computation of Optimal Supervisor

```

input :  $\mathbf{P}, \chi, \mathcal{C}$ 
output: Optimal set of disabled transitions  $\mathcal{D}^*$ 
1 begin
2   Set  $\mathcal{D}^{[0]} = \emptyset$ ;           /* Initial disabling set */
3   Set  $\tilde{\Pi}^{[0]} = \tilde{\Pi}$ ;         /* Initial event prob. matrix */
4   Set  $\theta_\star^{[0]} = 0.99$ ;
5   Set  $k = 1$ ;
6   Set Terminate = false;
7   while (Terminate == false) do
8     Compute  $\theta_\star^{[k]}$ ;           /* Algorithm 4 */
9     Set  $\tilde{\Pi}^{[k]} = \frac{1-\theta_\star^{[k]}}{1-\theta_\star^{[k-1]}} \tilde{\Pi}^{[k-1]}$ ;
10    Compute  $\nu^{[k]}$ ;
11    for  $j = 1$  to  $n$  do
12      for  $i = 1$  to  $n$  do
13        Disable all controllable transitions  $q_i \xrightarrow{\sigma} q_j$ 
14        such that  $\nu_j^{[k]} < \nu_i^{[k]}$ ;
15        Enable all controllable transitions  $q_i \xrightarrow{\sigma} q_j$ 
16        such that  $\nu_j^{[k]} \geq \nu_i^{[k]}$ ;
17      endfor
18    endfor
19    Collect all disabled transitions in  $\mathcal{D}^{[k]}$ ;
20    if  $\mathcal{D}^{[k]} == \mathcal{D}^{[k-1]}$  then
21      Terminate = true;
22    else
23       $k = k + 1$ ;
24    endif
25  endwhile
26   $\mathcal{D}^* = \mathcal{D}^{[k]}$ ;           /* Optimal disabling set */

```

Proposition 9 (effectiveness): Algorithm 5 is an effective procedure (Hopcroft *et al.* 2001), i.e., it is guaranteed to terminate.

Proof: Comparison of Algorithm 3 and Algorithm 5 reveals that while the former assumes a fixed probability of termination θ at each state, the latter modifies this parameter, denoted as $\theta_\star^{[k]}$, at each iteration k . Let $\theta_{\min} = \min(\theta_\star^{[1]}, \theta_\star^{[2]})$ and let $\mathcal{D}^{[1]}(\theta_{\min})$ and $\mathcal{D}^{[2]}(\theta_{\min})$ be sets of disabled transition at the first and second iterations, respectively, for the terminating plant $G(\theta_{\min})$. Similarly, for the non-terminating plant $G(0)$, let $\mathcal{D}^{[1]}(0)$ and $\mathcal{D}^{[2]}(0)$ be the sets of disabled transitions at the first and second iterations, respectively. It follows from Remark 3 that $\mathcal{D}^{[1]}(0) = \mathcal{D}^{[1]}(\theta_{\min})$ and $\mathcal{D}^{[2]}(0) = \mathcal{D}^{[2]}(\theta_{\min})$.

Extending the above argument by induction based on k iterations of Algorithm 5 and denoting $\theta_{\min} = \min(\theta_\star^{[1]}, \dots, \theta_\star^{[k]})$, an application of Algorithm 3 on a terminating plant $G(\theta_{\min})$ yields

$$\mathcal{D}^{[r]}(0) = \mathcal{D}^{[r]}(\theta_{\min}) \quad \forall r \in \{1, \dots, k\}.$$

Proposition 4 states that, for an arbitrary plant, Algorithm 3 is guaranteed to terminate within finitely many iterations. Hence, Algorithm 5 is an effective procedure. \square

Next, it is shown that the plant configuration obtained by Algorithm 5 is optimal in the sense of Definition 13.

Proposition 10 (optimality): For a non-terminating plant $G(0) = (Q, \Sigma, \delta, \tilde{\mathbf{P}}, \chi, \mathcal{C})$, the supervision policy computed by Algorithm 5 is optimal in the sense of Definition 13.

Proof: Let the set of disabled transitions computed at the k th iteration Algorithm 5 be denoted by $\mathcal{D}_{\text{lim}}^{[k]}$ and the termination probability be denoted by $\theta_{\star}^{[k]}$. Let the set of disabled transitions at the convergence of Algorithm 5 be $\mathcal{D}_{\text{lim}}^{[m]}$. Let $\theta_{\min} = \min_{r \in \{1, \dots, \ell\}} (\theta_{\star}^{[1]}, \dots, \theta_{\star}^{[\ell]}) > 0$.

Let $G(\theta_{\min})$ be a terminating plant with $\Pi(\theta_{\min}) = (1 - \theta_{\min})\mathbf{P}$. It follows from the proof of Proposition 9 that applications of Algorithm 3 to $G(\theta_{\min})$ and Algorithm 5 to $G(0)$ yield the same set \mathcal{D} of disabled controllable events although the optimal measures, being θ -dependent would be different, i.e., $\mathbf{v}(\theta_{\min}) \neq_{\mathbf{E}} \mathbf{v}(0)$.

Proposition 5 implies that the optimal disabling set for a plant $G(\theta)$ generates the the same set of disabled controllable transitions for all $0 < \theta \leq \theta_{\min}$. Because of continuity of $\mathbf{v}(\theta)$ with respect to θ , it is argued that $G^{\star}(0)$ is optimal in the sense of Definition 13, i.e., $\mathbf{v}^{\star} \cong_{\mathbf{E}} \mathbf{v}^{\dagger}$, where $G^{\dagger}(0)$ is obtained by arbitrarily disabling controllable transitions in G . This completes the proof. \square

Next it is shown that the supervision policy computed by Algorithm 5 is unique in the same sense as Proposition 6.

Proposition 11 (Uniqueness): Let $G(0)$ be an unsupervised non-terminating plant and $G^{\star}(0)$ be the supervised plant configured by Algorithm 5. Then, G^{\star} is unique in the sense that it is maximally permissive among supervised plants that yield optimal performance based on θ -neighbours $G(\theta)$ of $G(0)$ (see Definition 11) for all $\theta \in (0, \theta_{\star})$, where θ_{\star} is computed by Algorithm 4. Equivalently, if $G^{\dagger}(\theta)$ is an arbitrarily supervised plant, then the following condition holds:

$$(\mathbf{v}^{\star}(\theta) \cong_{\mathbf{E}} \mathbf{v}^{\dagger}(\theta)) \wedge \left((\mathcal{D}^{\star} \subseteq \mathcal{D}^{\dagger}) \vee (\mathbf{v}^{\star}(\theta) \neq_{\mathbf{E}} \mathbf{v}^{\dagger}(\theta)) \right),$$

where \mathbf{v} and \mathcal{D} denote respective language measures and sets of disabled transitions.

Proof: It follows from Proposition 10 that $\mathbf{v}^{\star}(0) \cong_{\mathbf{E}} \mathbf{v}^{\dagger}(0)$. It also follows from Proposition that $\mathbf{v}^{\star}(\theta) \cong_{\mathbf{E}} \mathbf{v}^{\dagger}(\theta)$ for $\theta \in (0, \theta_{\star})$. If $\mathbf{v}^{\star}(\theta) \equiv_{\mathbf{E}} \mathbf{v}^{\dagger}(\theta)$, then $G^{\star}(\theta)$ and $G^{\dagger}(\theta)$ are both optimal supervised configurations of the unsupervised terminating plant $G(\theta)$. It follows from Proposition 6 that $\mathcal{D}^{\star} \subseteq \mathcal{D}^{\dagger}$; otherwise $\mathbf{v}^{\star}(\theta) \neq_{\mathbf{E}} \mathbf{v}^{\dagger}(\theta)$. \square

Proposition 12: Computational complexity of Algorithm 5 is of the same order as that of Algorithm 3.

Proof: Algorithm 5 computes θ_{\star} in each iteration and complexity of this computation is $O(n^3)$, where n is the number of states in the plant (see Proposition 8). Each iteration of both Algorithm 3 and Algorithm 5 involves computation of the measure vector \mathbf{v} , whose complexity is also $O(n^3)$ because of $n \times n$ matrix inversion. Hence, computational complexity of each iteration is $O(n^3)$ for both Algorithm 3 and Algorithm 5. Finally, the argument presented in Proposition 9 implies that the number of iterations in Algorithm 5 is of the same order as that in Algorithm 3. This completes the proof. \square

4.2 Testing of computational complexity

Proposition 4 shows that Algorithm 3 is an effective procedure (Hopcroft *et al.* 2001), i.e., the solution is guaranteed to converge in a finite number of iterations. Extensive simulation suggests that the the maximum number of iterations for Algorithm 3 is actually of polynomial order in n , where n is the number of states in the unsupervised plant. The result is illustrated in figure 6, where the maximum number of required iterations I_{\max} is plotted against number, n , of plant states. For each n , 10,000 simulation runs were conducted for synthesis of optimal plant configuration with randomly generated entries in the pair $((1 - \theta)\mathbf{P}, \chi)$; and I_{\max} was chosen to be the maximum number of iterations required by Algorithm 3 to converge; this is the most conservative case. The plot in figure 1 shows a distinct sub-linear variation. The following conjecture is made based on these observations.

Conjecture 1 (polynomial convergence): *Given a terminating plant $G(\theta)$ with a uniform non-zero probability of termination θ at each of the n plant states,*

1. *Algorithm 3 converges in at most $O(n)$ iterations.*
2. *Computational complexity of Algorithm 3 is bounded by $O(n^4)$.*

Statement 2 in Conjecture 1 follows from Statement 1 and the following facts: Each iteration has complexity of $O(n^3)$ due to matrix inversion in the computation of

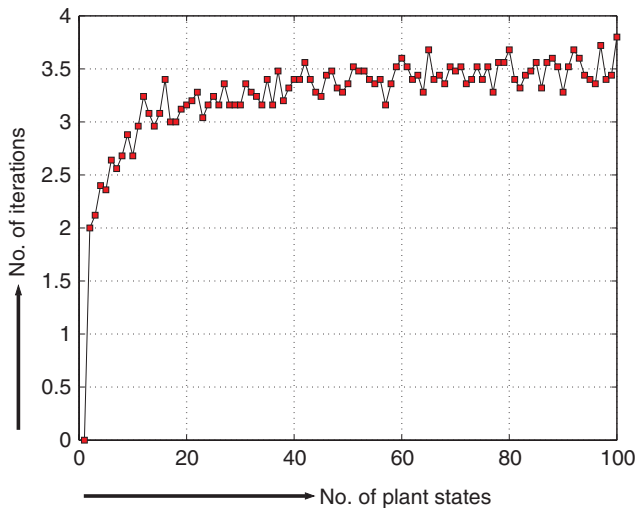


Figure 1. Number of iterations to converge in Algorithm 3.

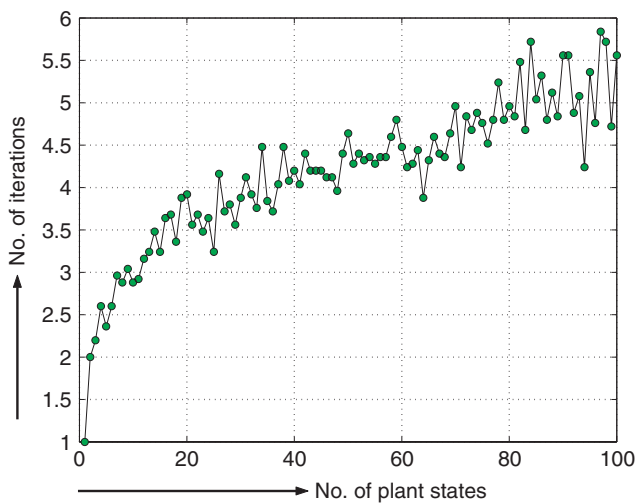


Figure 2. Number of iterations to converge in Algorithm 5.

the language measure vector, and matrix inversion has complexity of $O(n^3)$). Combination of Conjecture 1 and Proposition 12 implies that Algorithm 5 converges in $O(n)$ iterations and that complexity of the algorithm is $O(n^4)$. Similar to the procedure, described above for Algorithm 3, 10,000 random simulation runs for each n were conducted for testing Algorithm 5. Figure 2 shows the plot of average number of iterations required to converge at each value of n in contrast to figure 1, where the maximum number of iterations is plotted. As expected, the plot of figure 2 is also sub-linear.

5. Optimal control of three processor message decoding

This section presents the design of a discrete-event (controllable) supervisor for a multiprocessor message

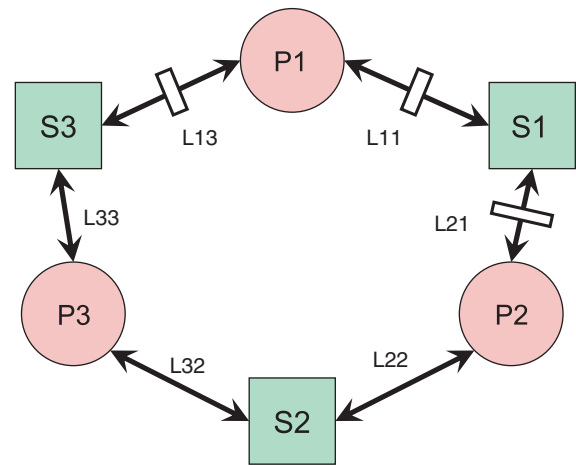


Figure 3. Arrangement of the processor links.

decoding system, described in an earlier publication (Ray *et al.* 2004). The optimal supervisory algorithm has been synthesized based on the algorithms presented in earlier sections.

Figure 3 depicts the arrangement of the message decoding system, where each of the three processors, p_1 , p_2 and p_3 , receives encoded messages that are to be decoded. The processor p_3 normally receives the most important messages, and p_1 receives the least important messages. There is a server between each pair of processors— s_1 between p_1 and p_2 ; s_2 between p_2 and p_3 ; and s_3 between p_3 and p_1 . Each server is connected to each of its two adjacent processors by a link—the server s_j is connected to the adjacent processors p_i and p_k through the links L_{ij} and L_{kj} , respectively. Out of these six links, each of the three links, L_{11} , L_{12} , and L_{21} , is equipped with a switch to disable the respective connection whenever it is necessary; each of the remaining three links, L_{22} , L_{32} , and L_{33} , always remain connected. Each server s_i is equipped with a decoding key k_i that, at any given time, can only be accessed by only one of the two processors, adjacent to the server, through the link connecting the processor and the server. In order to decode the message, the processor holds the information on both keys of the servers next to it, one at a time. After decoding, the processor simultaneously releases both keys so that other processors may obtain access to them.

Figure 4 depicts the unsupervised plant model of the decoding system as a finite state automaton, where state 1 is the initial state. The event p_{ij} indicates that processor p_i has accessed the key k_j ; and the event f_i indicates that the processor p_i has finished decoding and (simultaneously) released both keys in its possession upon completion of decoding. The events f_i are uncontrollable because, after the decoding is

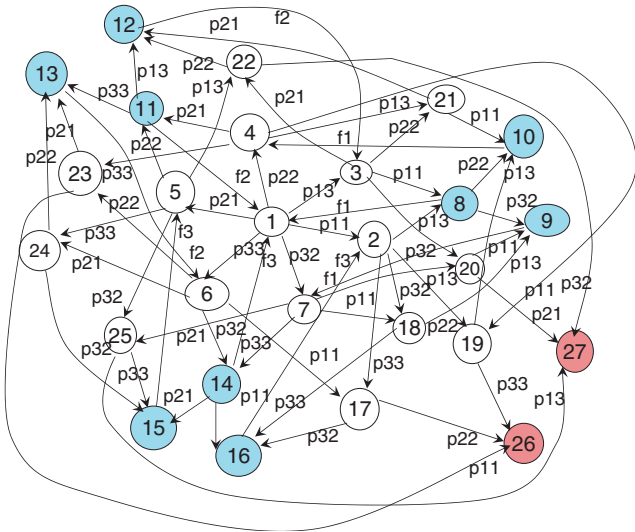


Figure 4. Finite state model of the message decoding system.

initiated, there is no control when a processor finishes decoding.

Table 1 lists the event cost matrix $\tilde{\Pi}$. Two different control specifications are investigated. The first set of specifications, which emphasizes avoiding deadlock, is represented by the χ vector in the first column of table 2. The second set of specifications, which focuses on increasing the throughput of processor 1, is represented by the χ vector in the second column of table 2. The positive elements of the χ vector are assigned to the states 8 to 16 that represent successful decoding of each processor. The χ values of the deadlock states 26 and 27, where each processor holds exactly one key and hence no processor releases its key, are assigned negative values. The remaining states are non-marked and are assigned zero weights.

Algorithm 5 is applied to obtain the sequence of measure vectors for the two control specifications. The results of successive iterations, enumerating the renormalized measure vectors, are presented in Table 3 and 4 respectively. The last column in each table is the optimal renormalized measure vector. The optimization requires 7 iterations in Case 1 and 5 iterations in Case 2.

The optimal configurations for the plant obtained under Algorithm 5 are depicted in figures 5 and 6 respectively. For supervisor policy 1, the controlled plant is not trim and, for supervisor policy 2, there are disconnected states in the controlled model. This is interpreted as the supervisor successfully preventing the plant from visiting these states. The critical values for the termination probability θ_* computed by the optimization algorithm for each control specification is shown in figure 7.

Table 1. Event occurrence probabilities for processor models.

p_{11}	p_{13}	p_{21}	p_{22}	p_{32}	p_{33}	f_1	f_2	f_3
0.16	0.04	0.16	0.16	0.16	0.32	0.00	0.00	0.00
0.00	0.16	0.00	0.26	0.26	0.32	0.00	0.00	0.00
0.37	0.00	0.21	0.21	0.21	0.00	0.00	0.00	0.00
0.32	0.11	0.26	0.00	0.00	0.32	0.00	0.00	0.00
0.00	0.11	0.00	0.28	0.28	0.33	0.00	0.00	0.00
0.25	0.00	0.25	0.25	0.25	0.00	0.00	0.00	0.00
0.28	0.11	0.28	0.00	0.00	0.33	0.00	0.00	0.00
0.00	0.00	0.00	0.39	0.39	0.00	0.22	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.14	0.00	0.00	0.00	0.43	0.00	0.43	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
0.33	0.00	0.33	0.00	0.00	0.00	0.00	0.00	0.34
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.25	0.00	0.00	0.00	0.75	0.00	0.00	0.00
0.50	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00
0.50	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
0.50	0.00	0.50	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.50	0.50	0.00	0.00	0.00	0.00
0.00	0.25	0.00	0.00	0.00	0.75	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.34
0.00	0.00	0.00	0.00	0.00	0.00	0.33	0.33	0.34

Table 2. Vectors for control specifications.

Case 1	Case 2
0.000	0.010
0.000	0.020
0.000	0.020
0.000	0.020
0.000	0.020
0.000	0.040
0.000	0.040
0.000	0.040
0.000	0.040
0.010	0.000
0.010	0.000

Next the stable probability distributions of the plant states are compared for the following three cases:

- Open-loop or unsupervised plant
- Plant with the optimal supervision policy for specification 1
- Plant with the optimal supervision policy for specification 2

The distributions are obtained by considering the first row of the matrix \mathcal{P} , based on the measure ν_1

Table 3. Iteration vectors for multi-processor model: case 1.

Itr 1	Itr 2	Itr 3	Itr 4	Itr 5	Itr 6	Itr 7
-0.0616	0.0006	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0001	0.0063	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0003	0.0055	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0011	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0012	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0000	0.0093	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0002	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0000	0.0093	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0003	0.0055	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0013	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0010	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0011	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0001	0.0063	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0001	0.0000	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0000	0.0093	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0009	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0000	0.0000	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0003	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0003	0.0000	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0014	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0012	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0012	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143
-0.0616	0.0007	0.0110	0.0124	0.0124	0.0143	0.0143

Table 4. Iteration vectors for multi-processor model: case 2.

Itr 1	Itr 2	Itr 3	Itr 4	Itr 5
0.0598	0.2076	0.2879	0.3245	0.3245
0.0598	0.2074	0.2880	0.3245	0.3245
0.0598	0.2101	0.2879	0.3245	0.3245
0.0598	0.2167	0.2876	0.3232	0.3245
0.0598	0.2109	0.2878	0.3236	0.3245
0.0598	0.2084	0.2882	0.3245	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2090	0.2879	0.3245	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2175	0.2875	0.3230	0.3245
0.0598	0.2089	0.2879	0.3245	0.3245
0.0598	0.2105	0.2879	0.3245	0.3245
0.0598	0.2086	0.2882	0.3245	0.3245
0.0598	0.2078	0.2879	0.3245	0.3245
0.0598	0.2114	0.2878	0.3235	0.3245
0.0598	0.2076	0.2880	0.3245	0.3245
0.0598	0.2080	0.2880	0.3245	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2216	0.2872	0.3241	0.3245
0.0598	0.2059	0.2878	0.3245	0.3245
0.0598	0.2147	0.2879	0.3245	0.3245
0.0598	0.2116	0.2879	0.3245	0.3245
0.0598	0.2084	0.2879	0.3245	0.3245
0.0598	0.2105	0.2878	0.3232	0.3245
0.0598	0.2110	0.2878	0.3236	0.3245
0.0598	0.2077	0.2879	0.3245	0.3245
0.0598	0.2077	0.2879	0.3245	0.3245

corresponding to state 1 which is the initial state in both cases. If the stochastic matrix \mathbf{P} is primitive (i.e., irreducible and acyclic), then all rows of \mathcal{P} would be identical. However, primitiveness of \mathbf{P} is not guaranteed even if the unsupervised plant model have this property because any subsequent event disabling may cause loss of reducibility or acyclic properties.

The results on evolution of the distribution are plotted in figure 8. While the unsupervised plant has a finite probability of reaching the deadlock states 26 and 27, the optimal supervisors in both cases successfully prevent occurrence of deadlock in the sense that the stable occupation probabilities for states 26 and 27 are zero for each supervisor. However, supervisor 2 increases the throughput of processor 1 as seen from the increased probability of occupying states 1 and 2.

6. Summary, conclusions, and recommendations for future work

This paper presents the theory, formulation, and validation of optimal supervisory control policies for dynamical systems, modelled as probabilistic finite

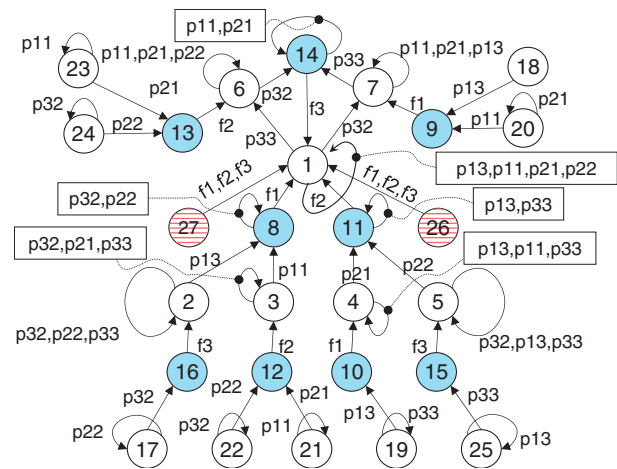


Figure 5. Optimal plant configuration for specification 1.

state automata. The procedure for synthesis of the optimal control policy relies on a (renormalized) signed real measure of regular languages (Chattopadhyay and Ray 2006a) to construct the performance index. The language measure is based on the state transition

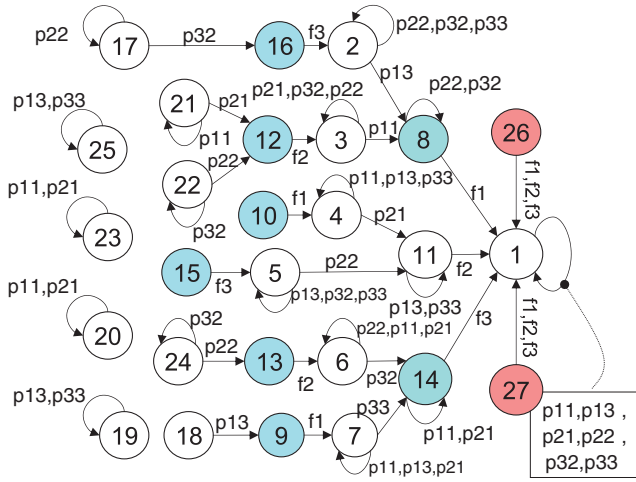


Figure 6. Optimal plant configuration for specification 2.

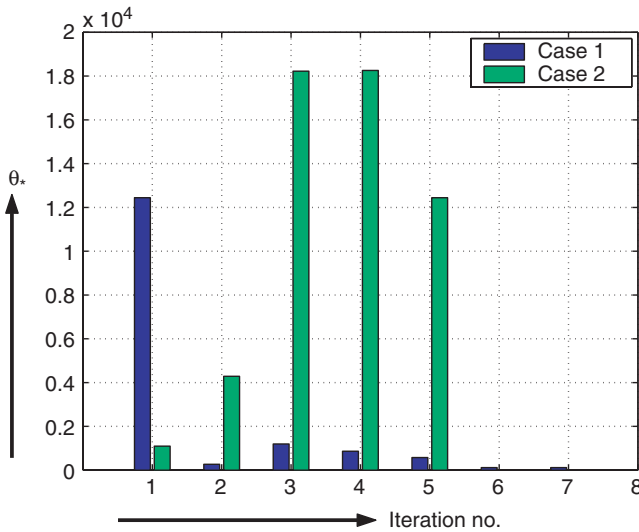


Figure 7. Computed θ_* for each iteration.

probability matrix of the underlying finite-state Markov chain model of the process and a characteristic vector of state weights, which serves as the control specification.

The main contribution of the paper is reformulation of the optimal supervisor synthesis algorithm (Ray *et al.* 2004, 2005) for probabilistic finite state plant models in terms of the renormalized measure and extension of the technique to general non-terminating probabilistic models. Specifically, the work reported in this paper removes a fundamental restriction of earlier analysis (Ray *et al.* 2004, Ray 2005), namely, each row sum of the state transition cost matrix Π being strictly less than one, instead of being exactly equal to one. The novel concept of language-based control synthesis, presented in this paper, allows quantification of plant

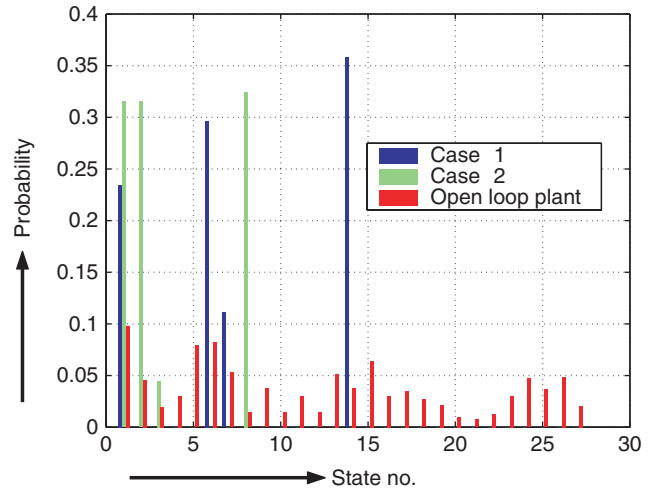


Figure 8. Stable state probability: unsupervised and supervised plants.

performance instead of solely relying on its qualitative performance (e.g., permissiveness), which is the current state of the art for discrete event supervisory control (Ramadge and Wonham 1987, Cassandras and Lafortune 1999).

The following conclusion is drawn in view of using the language measure for construction of the performance index for deriving an optimal control policy. Like any other optimization procedure, it is possible to choose different performance indices to arrive at different optimal policies for discrete event supervisory control. Nevertheless, usage of the language measure provides a systematic procedure for precise comparative evaluation of different supervisors so that the optimal control policy(ies) can be unambiguously identified. These theoretical results also lay the foundation for extension of the language-measure-theoretic framework to plant modelling and control, where all events may not be observable at the supervisory level.

The paper provides details of the algorithms that are required for synthesis of the optimal supervisory control policy. These algorithms are executable in real time on commercially available platforms. Computational complexity of the presented algorithms is polynomial in the number of plant model states. The concepts are elucidated with simple examples and a relevant engineering example. As such it is straight-forward to develop real-time software codes in standard languages, based on the algorithms provided in this paper.

There are several issues that need to be addressed for implementation of the theory of discrete-event supervisory control in an operating plant. For example, the events must be generated in real time, based on physical measurements, to provide the supervisor with the

current information on the plant; this is beyond what is done off-line for construction of the plant model and control synthesis. Similarly, the event disabling/enabling decisions of the supervisor must be translated in real time as appropriate actions to control the plant.

6.1 Recommendations for future research

Synthesis of supervisory control systems may become a significant challenge if some of the events are delayed, intermittent, or not observable at all, possibly due to sensor faults or malfunctions in network communication links. In that case, the control algorithms may turn out to be computationally very complex because of delayed or lost information on the plant dynamics. Future work in this direction should involve research on construction of language measures under partial observation (Chattopadhyay and Ray 2006b) and associated synthesis of optimal control policies to mitigate the detrimental effects of loss of observability. The latter research could be an extension of the work on optimal control under full observation, reported in this paper.

It would be a challenging task to extend the concept of (regular) language measure for languages higher up in the Chomsky hierarchy (Hopcroft *et al.* 2001) such as context-free and context-sensitive languages. This extension would lead to controller synthesis when the plant dynamics is modelled by non-regular languages such as the Petri net (Cassandras and Lafortune 1999, Murata 1989). The research thrust should focus on retaining the polynomial order of computational complexity.

Another critical issue is how to extend the language measure for timed automaton, especially if the events are observed with varying delays at different states. Another research topic that may also be worth investigating is: how to extend the $GF(2)$ field, over which the vector space of languages is defined (Ray 2005), to richer fields like the set of real numbers.

Areas of future research also include applications of the language measure in anomaly detection, model identification, model order reduction, and analysis and synthesis of interfaces between the continuously-varying and discrete-event spaces in the language-measure setting. Future research for advancement of the theory of optimal supervisory control for discrete event systems include the following areas:

- Robustness of the control policy relative to unstructured and structured uncertainties in the plant model including variations in the language measure parameters (Lagoa *et al.* 2005)
- Control synthesis under partial observation to accommodate loss of observability at the supervisory level

possibly due to sensor faults or communication link failures (Chattopadhyay and Ray 2006b)

- Construction of grammar-based measures, instead of memory-less state-based measures (Chattopadhyay and Ray 2005), for non-regular languages when details of transitions in plant dynamics cannot be captured by finitely many states

Acknowledgement

This work has been supported in part by the U.S. Army Research laboratory and the U.S. Army Research Office under Grant Nos. DAAD19-01-1-0646 and W911NF-06-1-0469.

Appendix A: Derivatives of renormalized measure

This appendix establishes bounds on the derivatives of the renormalized measure $\nu(\theta)$ for all $\theta \in (0, 1)$ and computes the limits of the derivatives as $\theta \rightarrow 0^+$ as an extension of what was reported in the previous publication (Chattopadhyay and Ray 2006a).

The main result on boundedness of the derivatives of $\nu(\theta)$ are presented as propositions. Specifically, the results reported in Chattopadhyay and Ray (2006a) are combined as the next two propositions.

Proposition A.1: *Let $\Psi(\theta) \triangleq \theta[\mathbb{I} - (1 - \theta)\mathbf{P}]^{-1}$, where \mathbf{P} is a $(n \times n)$ stochastic matrix and $n \in \mathbb{N}$. Then,*

$$\begin{aligned}
 & \text{(i) } \forall k \in \mathbb{N} \setminus \{1\} \\
 & \lim_{\theta \rightarrow 0^+} \frac{\partial^k \Psi(\theta)}{\partial \theta^k} = -k \lim_{\theta \rightarrow 0^+} \frac{\partial^k - 1 \Psi(\theta)}{\partial \theta^k - 1} [\mathbf{P} + \mathcal{P}][\mathbb{I} - \mathbf{P} + \mathcal{P}]^{-1} \\
 & \text{(ii) } \lim_{\theta \rightarrow 0^+} \frac{\partial^k \Psi(\theta)}{\partial \theta^k} \\
 & = \begin{cases} [\mathbb{I} - \mathbf{P} + \mathcal{P}]^{-1} - \mathcal{P}, & \text{if } k = 1 \\ (-1)^k k! [\mathbb{I} - \mathbf{P} + \mathcal{P}]^{-1} \\ \quad \times [\mathbb{I} - [\mathbb{I} - \mathbf{P} + \mathcal{P}]^{-1}]^{k-1}, & \text{if } k \in \mathbb{N} \setminus \{1\}. \end{cases}
 \end{aligned}$$

Proof: Given in Chattopadhyay and Ray (2006a, §3, pp. 1111–1112 as Corollary 3 and Corollary 6). \square

The next proposition establishes bounds on the derivatives of $\nu(\theta)$ in an elementwise sense by computing bounds on the induced sup-norm of the derivatives of $\Psi(\theta)$. Recall that χ has been defined to have infinity norm equal to 1.

Proposition A.2

$$\left\| \frac{\partial^k v(\theta)}{\partial \theta^k} \right\|_{\infty} \leq k! 2^{k+1} \left(\inf_{\alpha \neq 0} \left\| [\mathbf{0} - \mathbf{P} + \alpha \mathcal{P}]^{-1} \right\|_{\infty} \right)^k \quad \forall \theta \in [0, 1].$$

Proof: Given in of Chattopadhyay and Ray (2006a, §3, p. 1113 as Proposition 5). \square

Proposition A.3: Denoting the i th element of the k th derivative of the measure vector as $(\partial^k v(\theta)/\partial \theta^k)|_i$, it follows that

$$\begin{aligned} \forall k \in \{1, \dots, n\}, \frac{\partial^k v(\theta)}{\partial \theta^k} \Big|_i &= \frac{\partial^k v(\theta)}{\partial \theta^k} \Big|_j \\ \implies \forall \theta \in [0, 1], v(\theta) \Big|_i &= v(\theta) \Big|_j, \end{aligned}$$

where n is the number of states in the plant model.

Proof: First it is noted that

$$\begin{aligned} v(\theta) &= \theta \sum_{k=0}^{\infty} (1-\theta)^k \mathbf{P}^k \boldsymbol{\chi} \quad \forall \theta \in (0, 1] \\ &= \theta \sum_{k=0}^{\infty} \mathbf{\Pi}^k(\theta) \boldsymbol{\chi} \quad \forall \theta \in (0, 1]. \end{aligned} \quad (26)$$

Since $\mathbf{\Pi}(\theta)$ is a matrix of dimension $n \times n$, it follows from the Cayley–Hamilton Theorem (Bapat and Raghavan 1997) that integral powers of $\mathbf{\Pi}(\theta)$ can be expressed as polynomials of degree $n-1$ as follows:

$$\forall r \in \mathbb{N}, \mathbf{\Pi}^r(\theta) = \sum_{k=0}^{n-1} c_k \mathbf{\Pi}^k(\theta) \quad \text{with } c_k \in \mathbb{C}. \quad (27)$$

Since each term in the summation on the left hand side of equation (26) is a polynomial in θ of degree $n-1$, it follows that the summation is also a polynomial in degree $n-1$ (since the summation exists due to the sub-stochastic property of $\mathbf{\Pi}(\theta)$). Then it follows that each element of $v(\theta)$ is a polynomial of degree n . The result then follows from continuity. \square

Proposition A.4: For any stochastic matrix \mathbf{P} of dimension $n \times n$, the complexity of computing the limiting matrix \mathcal{P} is of the order of $O(n^3)$.

Proof: Since the limit $\lim_{k \rightarrow \infty} (1/k) \sum_{j=0}^{k-1} \mathbf{P}^j = \mathcal{P}$ always exists, it is possible to compute \mathcal{P} within any specified precision simply by computing the sum $\sum_{j=0}^{k-1} \mathbf{P}^j$ followed by division by k , for a large enough value of k . The procedure is summarized in Algorithm 6.

Algorithm 6: Computation of \mathcal{P}

```

input :  $\mathbf{P}$ , Desired precision  $\epsilon$ ps
output:  $\mathcal{P}$ 
1 begin
2   Set  $Q^{[0]} = \mathbf{0}$ ; /* Zero Matrix */
3   Set  $\mathbf{A} = \mathbf{I}$ ; /* Identity Matrix */
4   Set  $k = 1$ ;
5   Set Terminate = false;
6   while (Terminate == false) do
7      $Q^{[k]} = Q^{[k-1]} + \frac{1}{k} (\mathbf{A} - Q^{[k-1]})$ ;
8     if  $\|Q^{[k]} - Q^{[k-1]}\| < \epsilon$ ps then
9       Terminate == true;
10    else
11       $k = k + 1$ ;
12       $\mathbf{A} = \mathbf{A} \mathbf{P}$ ;
13    endif
14  endw
15   $\mathcal{P} = Q^{[k]}$ ;
16 end

```

Referring to Line 7 of Algorithm 6, it is observed that $Q^{[k]}$ is a stochastic matrix for all k and hence it follows that the algorithm is guaranteed to terminate in $(1/\epsilon)$ iterations, independent of n . Each iteration involves a single matrix multiplication ($\mathbf{P} \times \mathbf{A}$) and hence algorithmic complexity is of the same order as multiplication of two $n \times n$ matrices, i.e., $\leq O(n^3)$.

References

- R.B. Bapat and T.E.S. Raghavan, *Nonnegative matrices and Applications*, Cambridge, UK: Cambridge University Press, 1997.
- C.G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387333320.
- I. Chattopadhyay and A. Ray, “A complex measure for linear grammars”, *Demonstratio Mathematica*, 38:761–775, 2005.
- I. Chattopadhyay and A. Ray, “Renormalized measure of regular languages”, *Int. J. Control*, 79, pp. 1107–1117, 2006a.
- I. Chattopadhyay and A. Ray, “A language measure for partially observed discrete event systems”, *Int. J. Control*, 79, pp. 1074–1086, 2006b.
- S.L. Chung, S. Lafortune, and F. Lin, “Supervisory control using variable look-ahead policies”, *Discrete Event Dynamic Systems: Theory and Applications*, 4, pp. 237–268, July 1994.
- E-E. Doberkat, *Stochastic Automata: Stability, Nondeterminism, and Prediction*, volume 113 of *Lecture Notes in Computer Science*, Springer, Secaucus, NJ, USA, 1981. ISBN 3-540-10835-1.
- V.K. Garg, R. Kumar, and S.I. Marcus, “A probabilistic language formalism for stochastic discrete event systems”, *IEEE Transactions on Automatic Control*, 44, pp. 280–293, 1999.
- V.K. Garg, “An algebraic approach to modeling probabilistic discrete event systems”, *Proceedings of 1992 IEEE Conference on Decision and Control*, pp. 2348–2353, Tucson, AZ, December 1992.
- V.K. Garg, “Probabilistic languages for modeling of DEDs”, in *Proceedings of 1992 IEEE Conference on Information and Sciences*, pp. 198–203, Princeton, NJ, March 1992.
- J.E. Hopcroft, R. Motwani and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed., Massachusetts, USA: Addison-Wesley, 2001.
- R. Kumar and V.K. Garg, “Control of stochastic discrete event systems modeled by probabilistic languages”, *IEEE Transactions on Automatic Control*, 46, pp. 593–606, 2001. URL citeseer.ist.psu.edu/278517.html.

- C.M. Lagoa, J. Fu, and A. Ray, "Robust optimal control of regular languages", *Automatica*, 41, pp. 1339–1345, 2005.
- M. Lawford and W.M. Wonham, "Supervisory control of probabilistic discrete event systems", in *Proceedings of 36th Midwest Symposium on Circuits and Systems, Detroit, Michigan, USA*, pp. 327–331, 1993.
- E.T. Lee and L.A. Zadeh, "Note on fuzzy languages", *Information Sciences*, 1, pp. 421–434, 1969.
- R. Milner, *Communication and concurrency*, Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1989. ISBN 0-13-115007-3.
- M.K. Molloy, "Performance analysis using stochastic petri nets", *IEEE Transactions on Computers*, C-31, pp. 913–917, September 1982.
- H. Mortzavian, "Controlled stochastic languages", in *Proceedings of 1993 Allerton Conference*, pp. 938–947, Urbana, IL, 1993.
- T. Murata, "Petri nets: Properties, analysis and applications", in *Proceedings IEEE*, 77, pp. 541–540, April 1989.
- A. Paz, *Introduction to probabilistic automata (Computer science and applied mathematics)*. Orlando, FL, USA: Academic Press, Inc. 1971. ISBN 0125476507.
- S. Phoha, E. Peluso, and R.L. Culver, "A high fidelity ocean sampling mobile network (samon) simulator", *IEEE Journal of Oceanic Engineering, Special Issue on Autonomous Ocean Sampling Networks*, 26, pp. 646–653, 2002.
- M.O. Rabin, "Probabilistic automata", *Information and Control*, 6, pp. 230–245, 1963.
- P.J. Ramadge and W.M. Wonham, "Supervisory control of a class of discrete event processes", *SIAM J. Control and Optimization*, 25, pp. 206–230, 1987.
- A. Ray, J. Fu, and C.M. Lagoa, "Optimal supervisory control of finite state automata", *Int. J. Control*, 77, pp. 1083–1100, 2004.
- A. Ray, V. Phoha, and S. Phoha, *Quantitative measure for discrete event supervisory control*. Springer, New York, 2005.
- A. Ray, "Signed real measure of regular languages for discrete-event supervisory control", *Int. J. Control*, 78, pp. 949–967, 2005.
- R. Sengupta and S. Lafortune. An optimal control theory for discrete event systems, *SIAM Journal on Control and Optimization*, 36, pp. 488–541, 1998. URL citeseer.ist.psu.edu/sengupta98optimal.html.