

# A language measure for partially observed discrete event systems

I. CHATTOPADHYAY and A. RAY\*

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

(Received 14 November 2005; in final form 12 April 2006)

Recent literature has introduced and validated a signed real measure of regular languages for quantitative analysis and synthesis of discrete-event supervisory (DES) control systems, where all events are assumed to be observable. This paper presents a modification of the language measure for supervisory control under partial observation and shows how to generalize the analysis when some of the events may not be observable at the supervisory level. In the context of DES control synthesis, the language measure of partially observable discrete-event processes is expressed in a closed form which is structurally similar to that of completely observable discrete-event processes. Examples are provided to elucidate the concept of DES control under partial observation.

## 1. Introduction

According to the paradigm of discrete event supervisory (DES) control (Ramadge and Wonham 1987), the automaton model of a physical plant is a language generator whose behaviour is constrained by a supervisor to meet a given specification. The (controlled) sublanguage of plant behaviour could be different under different supervisors that satisfy their own respective specifications. Such a partially ordered set of sublanguages requires a quantitative measure for total ordering of their respective performance. To address this issue, a signed real measure of regular languages has been reported in recent literature (Ray 2005) to provide a mathematical framework for quantitative comparison of regular languages. This measure formalizes synthesis of DES control systems for finite state automaton plants, as an alternative to the procedure of Ramadge and Wonham (1987). Another recent paper (Ray *et al.* 2004) has reported optimal supervisory control of finite state automata based on the language measure to formalize quantitative analysis and synthesis of DES control laws. The approach is state-based and the language measure parameters are identified from

experiments on the physical process or from simulation experiments on a deterministic finite state automaton (DFSA) model of the plant. However, the language measure used in the optimal control synthesis has not addressed the issue of unobservable events at the supervisory level. This problem is of significant practical importance because the model of a given physical process may have inherently unobservable transitions. Furthermore, in complex engineering systems, events may become unobservable due to failures of sensors or communication links in one or more locations.

This paper extends the concept of language measure (Ray 2005) to more general scenarios, where the set of unobservable events (at the supervisor) is allowed to be non-empty. The main idea behind this generalization is to realize the fact that the observed initiation and/or termination state of a given string may not be the true initiation and/or termination state due to the presence of unobservable transitions. Although partial observability with respect to the supervisory control theory of Ramadge and Wonham (1987) have been widely studied (Lin and Wonham 1988b, Wong 1997, Wong and Wonham 2004), such generalizations for the measure theoretic approach (Ray 2005) has not been reported in literature. The results presented in this paper lay the foundation for extending the existing theory

---

\*Corresponding author. Email: axr2@psu.edu

(Ray *et al.* 2004) of optimal control synthesis under full observation to that under partial observation.

This paper is organized in five sections, including the present section, and two appendices. Section 2 presents the proposed generalization of the measure to automata with a (possibly) non-empty set of unobservable events. An analysis of the worst case effect of unobservability is presented in §3 through the introduction of the unobservability index. Section 4 presents three examples to elucidate the concept of DES control under partial observability. The paper is summarized and concluded in §5 along with recommendations for future research. Appendix A briefly describes the notations and background materials for deriving a signed real measure of regular languages (Ray 2005) and Appendix B derives necessary relations for the analysis in §2.

## 2. Unobservability in discrete event systems

The generated language  $L(G_i)$  for a DFSA  $G_i \equiv (Q, \Sigma, \delta, q_i, Q_m)$ , defined in Appendix A, is the set of all event strings, accepted by the DFSA  $G_i$  with  $Q_m = Q$  and the initial state  $Q_i \in Q$ . Since  $L(G_i)$  is a prefix-closed language (Ramadge and Wonham 1987) for every  $i \in I_Q$ , a change in the initial state from  $q_i$  to  $q_j$  with  $i \neq j$ , leads to a different generated language  $L(G_j)$ . Thus, it is possible to define different prefix closed languages  $L(G_i)$ , one for each initial state  $q_i$ . In the union of these languages  $\cup_{i \in I_Q} L(G_i)$ , it may become impossible to distinguish between identical symbol sequences generated from different initial states. This difficulty is alleviated by labelling each string with the index (or *colour*) of its initial state. Note that the empty string  $\varepsilon$  has no colour.

**Definition 1:** For a given DFSA having the state set  $Q$  and symbol alphabet  $\Sigma$ , the colour alphabet is defined to be the state index set  $I_Q$ . Then, the unique label  $j$  associated with the initial state  $q_j$  of a string  $t \in \cup_{i \in I_Q} L(G_i)$  is called the colour of the string  $t$ .

The properties of the colour alphabet  $I_Q$  are summarized below:

- There exists a bijective mapping between  $I_Q$  and  $Q$ .
- $I_Q \cap \Sigma = \emptyset$ .
- $j\varepsilon = j \forall j \in I_Q$ .
- Each symbol in  $I_Q$  has zero string length, i.e.,  $|j| = 0, \forall j \in I_Q$ .

**Definition 2:** For a given DFSA  $G_i$  with the initial state  $q_i \in Q$ , the  $i$ th colour language associated with the colour alphabet  $I_Q$  is defined to be

$$L_i^c = iL(G_i) \cup \{\varepsilon\} \quad \text{and} \quad L^c = \bigcup_{i \in I_Q} L_i^c \quad (1)$$

and the  $i$ th total colour language is defined to be

$$L_i^c = i\Sigma^* \cup \{\varepsilon\} \quad \text{and} \quad L^c = \bigcup_{i \in I_Q} L_i^c. \quad (2)$$

The  $i$ th colour language is obtained from the generated language  $L(G_i)$  by prefixing all non-null strings with the colour of the starting state of the strings, namely  $i$ . Note that string lengths do not change due to the presence of the zero-length colour. Furthermore, if the state transition function  $\delta$  in the DFSA  $G_i \equiv (Q, \Sigma, \delta, q_i, Q_m)$  is a total function, then  $L_i^c = L_i^c$ .

The complete colour language  $L^c$  is the union of (disjoint) colour languages and hence the problem of distinguishing identical symbol sequences starting from different states is alleviated. In general,  $L^c \subseteq L^c$  with the equality holding if and only if  $L(G_i) = \Sigma^* \forall i \in I_Q$ .

**Remark 1:** For every non-empty (coloured) string  $s \in L^c$ , there exists  $j \in I_Q, t \in \Sigma^*$  such that  $s = jt$ .

**Definition 3:** The (possibly non-commutative) binary operation  $\circ: L^c \times L^c \rightarrow L^c$  is defined as

$$\left. \begin{array}{l} \forall s \in L^c, s \circ \varepsilon = \varepsilon \circ s = s \\ \text{and } \forall s, \tilde{s} \in L^c - \emptyset, s \circ \tilde{s} = (it) \circ (\tilde{i}\tilde{t}) \equiv i\tilde{t} \\ \text{for some } t, \tilde{t} \in \Sigma^*. \end{array} \right\} \quad (3)$$

In general,  $\circ$  is not a closed operation. That is, for some  $s, \tilde{s} \in L^c, s \circ \tilde{s}$  may not be an element of  $L^c$ . If  $s \circ \tilde{s} \in L^c$ , then  $s$  and  $\tilde{s}$  are said to be compatible. It should be noted that compatibility of  $s$  and  $\tilde{s}$  implies the colour of the terminating state of  $s$  is the same as the colour of the starting state of  $\tilde{s}$ . That is, if  $s \in L_i^c$  and  $\tilde{s} \in L_j^c$ , compatibility of  $s$  with  $\tilde{s}$  implies  $s \circ \tilde{s} \in L_j^c$ . In Definition 3, the operation  $\circ$  can be viewed as a special concatenation. The following extension is introduced by analogy to extension of string concatenation to language concatenation.

**Definition 4:** The extended binary operation  $\circ \times 2^{L^c} \times 2^{L^c} \rightarrow 2^{L^c}$  is defined as

$$\left. \begin{array}{l} \forall L \subseteq L^c, L \circ \varepsilon = \varepsilon \circ L = L \\ \text{and } \forall L, \tilde{L} \subseteq L^c, L \circ \tilde{L} = \bigcup_{s \in L} \bigcup_{\tilde{s} \in \tilde{L}} \{s \circ \tilde{s}\}. \end{array} \right\} \quad (4)$$

**Remark 2:** For any  $s \in L^c$  with  $|s| = k \in \{1, 2, 3, \dots\}$ , there exist unique coloured strings  $s_1, s_2, \dots, s_k$  with  $s_j \in L^c$  such that  $s_1 \circ s_2 \circ \dots \circ s_k = s$ . To see this, let  $s = i_1 \sigma_1 \dots \sigma_k$  with  $i_1 \in I_Q, \sigma_j \in \Sigma$ . This implies that  $s \in L^c(G_{i_1})$  and

$$q_{i_1} \xrightarrow{\sigma_1} q_{i_2} \xrightarrow{\sigma_2} \dots \xrightarrow{\sigma_{k-1}} q_{i_k} \xrightarrow{\sigma_k} q_{i_{k+1}}. \quad (5)$$

Hence, by defining  $s_k \equiv i_k \sigma_k$ , one can construct  $s_k \in L_{i_k}^c$ . It also follows from Definition 3 that  $s_1 \circ s_2 \circ \dots \circ s_k = s$ , which is unique because of the deterministic property

of the language  $L(G_i)$ . Furthermore, for any given non-empty  $s \in L^c$ , there are unique  $j \in I_Q$ ,  $\sigma \in \Sigma$  and  $\tilde{s} \in L^c$  such that  $s = (j\sigma) \circ \tilde{s}$ .

**Definition 5:** The colour map  $c : L^c \rightarrow I_Q \cup \{\varepsilon\}$  is defined as

$$\forall s \in L^c, \quad c(s) = \begin{cases} \varepsilon & \text{if } s = \varepsilon \\ i & \text{if } s = i\sigma \text{ for some } i \in I_Q, \sigma \in \Sigma^* \end{cases} \quad (6)$$

and  $c(s)$  denotes the colour of the string  $s$ .

**Definition 6:** The observation map  $p : L^c \rightarrow L^c$  is defined as follows:

$$\left. \begin{aligned} p(\varepsilon) &= \varepsilon, \\ \forall s \in L^c - \{\varepsilon\} \text{ such that } s &= s_1 \circ \dots \circ s_k, \\ p(s) &= s_{i_1} \circ \dots \circ s_{i_j} \circ \dots \circ s_{i_r}, \end{aligned} \right\} \quad (7)$$

where  $i_j \in \{1, \dots, k\} \forall j \in \{1, \dots, r\}$  and  $p(s)$  denotes the observed string for any given string  $s \in L^c$ .

**Remark 3:** The following facts hold in view of Definitions 5 and 6:

- The observed starting state of a string  $s \in L^c$  has the colour  $c(p(s))$ .
- Since  $s_{i_j}, s_{i_{j+1}}$  in Definition 6 are, in general, not compatible, it is possible that  $p(s) \notin L^c$  for a given string  $s \in L^c$ .

**Definition 7:** A string  $s \in L^c$  is said to be completely unobservable if  $p(s) = \varepsilon$ .

**Definition 8:** The observed language  $\mathcal{O}_p$  for a DFSA with respect to a given observation map  $p(\cdot)$ , is defined to be the image  $\text{Im}(p)$  of the observation map  $p$ , i.e.,

$$\mathcal{O}_p = p(L^c). \quad (8)$$

It follows from Definition 8 that the observed language  $\mathcal{O}_p \subseteq L^c$  and is not necessarily a subset of  $L^c$ . The observation map completely specifies unobservability of the plant automaton. The two simple cases of unobservability are presented below.

**Definition 9:** A DFSA plant is said to have state-dependent regular unobservability if

$$\left. \begin{aligned} \forall s \equiv s_1 \circ s_2 \circ \dots \circ s_k \in L^c, \\ p(s) = p(s_1) \circ p(s_2) \circ \dots \circ p(s_k). \end{aligned} \right\} \quad (9)$$

A plant is said to have state-independent regular unobservability if, in addition,

$$p(i\sigma) = p(j\sigma) \quad \forall \sigma \in \Sigma, \forall i, j \in I_Q. \quad (10)$$

**Remark 4:** Regular state-dependent unobservability in Definition 9 can be specified by marking certain

transitions in the graph of the DFSA as unobservable. Note that, in general, the same event may be observable at one state and unobservable at another state. When this possibility is precluded, the situation is state-independent regular unobservability. The state-independent regular unobservability is a special case of state-dependent regular unobservability; both cases have been jointly referred to as regular unobservability in the sequel. For regular unobservability, the observation map is completely specified by defining an event string in terms of unit-length strings  $\sigma \in L^c$ . However, the map  $p(\cdot)$  can be more complicated. For example, consider a single state DFSA with its language  $L(G_1) = \Sigma^*$ , where  $\Sigma^* = \{\sigma\}$ , and the observation map is defined by the condition that the symbols in the odd number positions of any string are not observed. Then,

$$\begin{aligned} 1\sigma &\mapsto \varepsilon \\ 1\sigma\sigma &\mapsto 1\sigma \\ 1\sigma\sigma\sigma &\mapsto 1\sigma \\ 1\sigma\sigma\sigma\sigma &\mapsto 1\sigma\sigma \\ 1\sigma\sigma\sigma\sigma\sigma &\mapsto 1\sigma\sigma \\ &\text{and so on.} \end{aligned}$$

Therefore,  $p(\cdot)$  cannot be defined by specifying its value on the unit length string  $\sigma$ . Interestingly, in this particular example,  $\mathcal{O}_p$  is still a regular language, namely  $\Sigma^*$ , which may not be the case in general.

**Remark 5:** In case of state-independent regular unobservability, the observation map  $p$  is a natural projection (Lin and Wonham 1988a,b, Wong 1997) and for state-dependent regular unobservability  $p$  has the more general structure of a causal reporter map (Wong and Wonham 1996, Wong 1997) (since the event alphabet can no longer be partitioned into observable and unobservable events). Note, in both cases of regular unobservability, the observation maps are prefix-preserving (Wong 1997). Specifically, if  $s$  is a prefix of  $s'$ , then  $p(s)$  is a prefix of  $p(s')$  for an unobservability map  $p$ .

**Definition 10:** The phantom language  $\mathfrak{U}_p$  of a DFSA plant with respect to a given observation map  $p$  is defined to be the kernel  $\ker(p)$ , i.e.,

$$\mathfrak{U}_p = \{s \in L^c \mid p(s) = \varepsilon\}. \quad (11)$$

It follows from Definition 10 that the phantom language is a sublanguage of  $L^c$ , but need not be a regular sublanguage. However, for regular unobservability, it will be shown that the phantom language is indeed a regular sublanguage of  $L^c$ .

**Definition 11:** Given a DFSA  $G_i$  and an observation map  $p$ , the phantom automaton  $\mathcal{P}(G_i)$  is defined to be a subautomaton of  $G_i$  such that the language  $L_i^o \subseteq L(G_i)$ , generated by  $\mathcal{P}(G_i)$ , consists of completely

unobservable strings, i.e.,  $p(s) = \epsilon$  for every  $s \in iL_i^o$  (see Definition 7). Therefore,

$$L_i^o = \bigcup_{j \in I_Q} L_{i,j}^o, \quad (12)$$

where  $L_{i,j}^o \equiv \{s \in L_i^o \mid s \text{ terminates on state } q_j\}$ .

For regular unobservability, the phantom automaton denoted by  $\mathcal{P}(G_i)$  of a DFSA  $G_i$  is given by Algorithm 1.

**Algorithm 1:** Derivation of phantom automation  $\mathcal{P}(G_i)$

**input:** DFSA  $G_i, I_Q, p$   
**output:** DFSA  $\mathcal{P}(G_i)$   
**begin**  
  **for**  $i \in I_Q$  **do**  
    **for**  $i \in \Sigma$  **do**  
      **if**  $p(i\sigma_j)$  is defined AND  $(i\sigma_j) \neq \epsilon$  **then**  
        Delete transition  $\sigma_j$  from state  $q_i$   
      **end**  
    **end**  
  **end**  
  Set the initial state of the automaton to  $q_i$   
**end**

Algorithm 1 gives an automaton with initial state  $q_i$ . Denote it by  $A_i$  and the language generated by it by  $L(A_i)$ . Obviously,  $L(A_i) \subseteq \Sigma^*$ . If  $s \in L(A_i)$ , then

$$\begin{aligned} p(is) &= p(i_1\sigma_1) \circ p(i_2\sigma_2) \circ \cdots \circ p(i_k\sigma_k) \quad (\text{Definition 9}) \\ &= \epsilon \circ \cdots \circ \epsilon \\ &= \epsilon \end{aligned}$$

which implies  $s \in L_i^o$ . Conversely, if  $s \in L_i^o$ , then

$$\begin{aligned} p(is) &= \epsilon = p(i_1\sigma_1) \circ p(i_2\sigma_2) \circ \cdots \circ p(i_k\sigma_k) \\ \Rightarrow p(i_j\sigma_j) &= \epsilon \forall j \in \{1, \dots, k\} \Rightarrow s \in L(A_i). \end{aligned}$$

**Remark 6:** Algorithm 1 holds only for regular unobservability. It follows that  $L_i^o$  is a regular sublanguage of  $L(G_i)$ . Thus,  $\mathfrak{U}_p = \cup_{i \in I_Q} iL_i^o$  is a regular sublanguage of  $L_c$ .

Next the measure of the language generated by the  $i$ th phantom automaton is defined in the sense of the measure construction presented in Appendix A. Denoting the measure of  $L_i^o$  as  $\mu_i^o$ , the equivalent matrix form is

$$\mu^o = (\mathbf{I} - \mathcal{P}(\Pi))^{-1} \chi, \quad (13)$$

where  $\mu^o \equiv [\mu_1^o \mu_2^o \cdots \mu_n^o]^T$ ; and the phantom transition cost matrix  $\mathcal{P}(\Pi)$  is defined to be the transition cost matrix of the DFSA  $\mathcal{P}(G_i)$  (see Definition A.5 in Appendix A). The inverse  $[\mathbf{I} - \mathcal{P}(\Pi)]^{-1}$  exists because  $\mathcal{P}(\Pi)$  is elementwise non-negative and bounded by  $\Pi$ , and  $\Pi$  is a contraction implying that  $\mathcal{P}(\Pi)$  is a contraction as well.

**Definition 12:** The  $i$ th colour component  $z_i: 2^{L^c - \{\epsilon\}} \rightarrow 2^{L(G_i)}$  is defined as:  $\forall L \equiv \{s_1, s_2, \dots, s_k, \dots\} \subseteq L^c - \{\epsilon\}$ ,

$$\begin{aligned} z_i(L) &= z_i(\{s_1, s_2, \dots, s_k, \dots\}) \\ &= \{z_i(\{s_1\}), z_i(\{s_2\}), \dots, z_i(\{s_k\}), \dots\}, \end{aligned} \quad (14)$$

where

$$z_i(jt) = \begin{cases} \{\epsilon\} & \text{if } i \neq j \\ \{t\} & \text{if } i = j \end{cases} \quad \text{and } z_i(\emptyset) = \emptyset.$$

Now a signed real measure  $\vartheta: 2^{L^c} \rightarrow \mathbb{R} \equiv (-\infty, +\infty)$  is constructed on the  $\sigma$ -algebra  $2^{L^c}$ , similar to the construction of language measure in Appendix A. With the choice of this  $\sigma$ -algebra, every singleton set made of a string  $s \in L^c$  is a measurable set, which qualifies itself to have a numerical quantity based on the following construction

$$\forall L \subseteq L^c, \quad \vartheta(L) = \sum_{i \in I_Q} \mu^i(z_i(L - \{\epsilon\})). \quad (15)$$

The set function  $\vartheta$  is a well-defined signed measure that satisfies the following conditions:

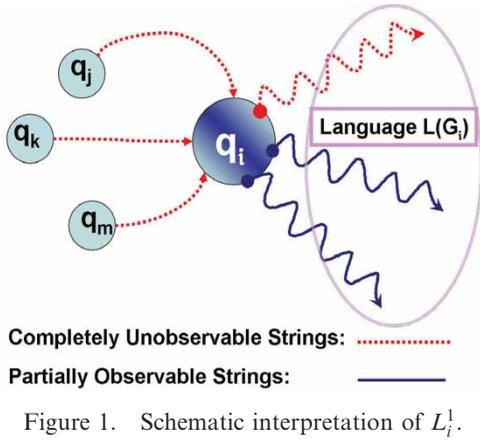
- $\vartheta(\emptyset) = 0$ .
- Finiteness and countable additivity of  $\vartheta$  inherited from the language measure  $\mu$  (see Appendix A).
- $\vartheta(iL) = \mu^i(L) \forall L \subseteq L(G_i)$ .
- $\vartheta(\{\epsilon\}) = \sum_{i \in I_Q} \mu^i(z_i(\emptyset)) = \sum_{i \in I_Q} \mu^i(\emptyset) = 0$ .

Construction of the language measure in Appendix A facilitates computation of a quantitative measure of the language  $L(G_i)$  for all  $i \in I_Q$ . This measure is adequate under complete observability and has been successfully used for optimal control under full observation (Lin and Wonham 1988a,b, Wong 1997). However, in the generalized case of partial observability, event strings that are observed to initiate from a given state may actually start from a different one. A control policy that does not take this effect into consideration may cause colossal errors. For example, the supervisor may enable a controllable event leading to a state of negative weight based on its observation which erroneously indicated that the particular transition will terminate at a state of positive weight. Two sublanguages of  $L^c$ , denoted by  $L_i^1$  and  $L_i^{11}$ , are introduced next to address these issues.

**Definition 13:** The language  $L_i^1$  is the set of all strings which pass through the state  $q_i$  with the restriction that the prefix of each string leading to the state  $q_i$  is completely unobservable. Formally,

$$L_i^1 = \{s \in L^c \mid s = s_1 \circ s_2 \text{ where } p(s_1) = \epsilon \text{ and } s_2 \in L_i^c\} \quad (16)$$

Figure 1 illustrates the situation.

Figure 1. Schematic interpretation of  $L_i^1$ .

By Definition 13,  $(s \in L_i^1) \Rightarrow (s = s_1 \circ s_2)$  and  $(s_2 \in L_i^c) \Rightarrow (s_1 \text{ terminates on state } q_i)$ , which follows from the fact that  $s_1, s_2$  has to be compatible since  $s_1 \circ s_2 \in L^c$ . Hence,  $s_1$  initiates from a state  $q_k \in Q$ , terminates on state  $q_i$ , and is completely unobservable by definition. Hence from Definition 10, it follows that  $s_1 \in kL_{k,i}^o$ . Hence,  $s \in L_i^1 \Rightarrow s \in \cup_{k \in I_Q} kL_{k,i}^o \circ L_i^c$ . The converse follows with a similar reasoning. Therefore,

$$L_i^1 = \bigcup_{k \in I_Q} kL_{k,i}^o \circ L_i^c \quad (17)$$

**Remark 7:** It is important to note that  $L_i^1$  is, in general, a superset of the set of strings which are observed to initiate at state  $q_i$ . For example, if  $(s \in L_i^1) \Rightarrow (s = s_1 \circ s_2)$  and  $s_2 = i\sigma \circ rs_r$ , where  $\sigma \in \Sigma, r \in I_Q, s_r \in L_r^c$  and  $p(i\sigma) = \varepsilon$ , then the string is observed to have initiated from state  $q_r$ , as seen in figure 1. The string that takes the dotted path after state  $q_i$  is not observed to initiate from  $q_i$ , i.e., both  $q_i$  and its prefix are missed. A solution to this problem is to consider the particular subset of  $L_i^1$  having only those strings for which at least the first transition after  $q_i$  is observable.

**Definition 14:** The language  $L_i^{11}$  is defined to be the subset of  $L_i^1$  such that, for any string in  $L_i^{11}$ , the first transition after  $q_i$  is unobservable. Formally,

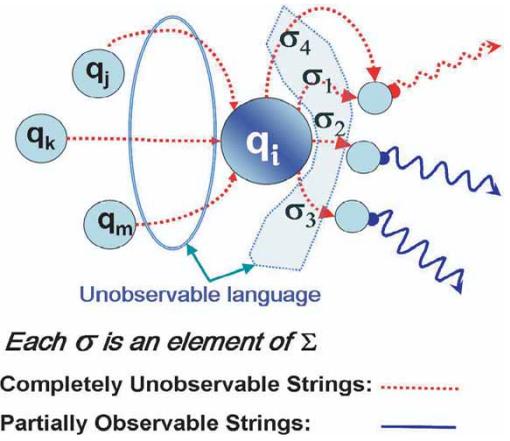
$$L_i^{11} = \{s \in L_i^1 : s = (s_1 \circ s_2) \Rightarrow (s = s_1 \circ i\sigma \circ s_r)\} \quad (18)$$

where  $s_1 \in L_{k,i}^o; \sigma \in \Sigma; s_r \in L_r^c(G_r)$  for some  $r, k \in I_Q$ ; and  $p(i\sigma) = \varepsilon$ .

It follows from figure 2 and Definition 14 that  $p(i\sigma_r) = \varepsilon$  if and only if  $\sigma_r \in \Sigma_i^o \subseteq \Sigma$ .

Hence,

$$s \in L_i^{11} \Rightarrow s \in \left( \bigcup_{k \in I_Q} kL_{k,i}^o \right) \circ \left( \bigcup_{\sigma_i \in \Sigma_i^o} i\sigma_i^j L(G_j) \right)$$

Figure 2. Schematic interpretation of  $L_i^{11}$ .

and vice versa. Therefore,

$$L_i^{11} = \left( \bigcup_{k \in I_Q} kL_{k,i}^o \right) \circ \left( \bigcup_{\sigma_r^o \in \Sigma_i^o} i\sigma_r^j L(G_j) \right) \quad (19)$$

where  $\Sigma_i^o$  is the set of unobservable events in  $\Sigma$ , occurring at state  $q_i$ ; and  $\sigma_r^j \in \Sigma_i^o$  such that there exists a transition  $\sigma_r^j$  in the phantom automaton  $\mathcal{P}(G_i)$ , with  $\delta(q_i, \sigma_r^j) = q_j$ .

It follows from Remark 7 that  $L_i^1 - L_i^{11}$  is the set of all event strings in  $L_i^1$  for which the first transition after the state  $q_i$  is observable; that is, these strings are observed to have initiated from state  $q_i$ . Figures 1 and 2 clarify the idea. Therefore, following Definitions 13 and 14, the set difference  $L_i^1 - L_i^{11}$  is the collection of all strings in  $L^c$  that are observed to have initiated from state  $q_i$ .

## 2.1 Computation of the language measure

This section presents the computation of the measures of  $L_i^1, L_i^{11}$  and  $L_i^1 - L_i^{11}$  from the closed form expressions that are presented as the following theorem.

**Theorem 1:** The closed form expression for the language measure under partial observation is obtained as follows:

$$\vartheta(L^1 - L^{11}) = \mathbb{D} \left( [I - \mathcal{P}(\Pi)]^{-T} \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix} \right) \times [I - \mathcal{P}(\Pi)][I - \Pi]^{-1} \chi, \quad (20)$$

where  $\mathbb{D}$  is the diagonalization map as defined in Appendix B.

**Proof:** Two lemmas that are necessary to prove the theorem are presented below.

**Lemma 1:** The measure  $\vartheta$  of the language  $L^1$  is expressed in closed form as

$$\vartheta(L^1) = \mathbb{D} \left( [\mathbf{I} - \mathcal{P}(\Pi)]^{-T} \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix} \right) [\mathbf{I} - \Pi]^{-1} \chi. \quad (21)$$

**Proof:**

$$\begin{aligned} \vartheta(L_i^1) &= \vartheta \left( \bigcup_{k \in I_Q} kL_{k,i}^e \circ L^c(G_i) \right) \\ &= \sum_{k \in I_Q} \vartheta(kL_{k,i}^e \circ L^c(G_i)) \\ &= \sum_{k \in I_Q} \mu^k(L_{k,i}^e L(G_i)) \\ &= \sum_{k \in I_Q} \left\{ \sum_{\omega \in L_{k,i}^e} \tilde{\pi}^e[\omega, q_k] \right\} \mu^i(L(G_i)) \\ &= \sum_{k \in I_Q} \left\{ \sum_{\omega \in L_{k,i}^e} \tilde{\pi}^e[\omega, q_k] \right\} \mu^i \\ &= \left\{ \mathbb{D} \left( [\mathbf{I} - \mathcal{P}(\Pi)]^{-T} \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix} \right) \mu \right\}_i \quad (\text{See Appendix B}) \\ &= \left\{ \mathbb{D} \left( [\mathbf{I} - \mathcal{P}(\Pi)]^{-T} \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix} \right) [\mathbf{I} - \Pi]^{-1} \chi \right\}_i. \quad \square \end{aligned}$$

**Lemma 2:** The closed form expression for  $\vartheta(L_i^{11})$  is given as

$$\vartheta(L^{11}) = \mathbb{D} \left( [\mathbf{I} - \mathcal{P}(\Pi)]^{-T} \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix} \right) \mathcal{P}(\Pi) [\mathbf{I} - \Pi]^{-1} \chi \quad (22)$$

**Proof:**

$$\begin{aligned} \vartheta(L_i^{11}) &= \vartheta \left( \left( \bigcup_{k \in I_Q} kL_{k,i}^e \right) \circ \left( \bigcup_{\substack{\sigma_r^j \in \Sigma_i^e \\ \delta(q_i, \sigma_r^j) = q_j}} \sigma_r^j L(G_j) \right) \right) \\ &= \sum_{k \in I_Q} \vartheta \left( kL_{k,i}^e \circ \left( \bigcup_{\substack{\sigma_r^j \in \Sigma_i^e \\ \delta(q_i, \sigma_r^j) = q_j}} i\sigma_r^j L(G_j) \right) \right) \end{aligned}$$

$$\begin{aligned} &= \sum_{k \in I_Q} \mu^k \left( \left( L_{k,i}^e \right) \left( \bigcup_{\substack{\sigma_r^j \in \Sigma_i^e \\ \delta(q_i, \sigma_r^j) = q_j}} i\sigma_r^j L(G_j) \right) \right) \\ &= \sum_{k \in I_Q} \left\{ \sum_{\omega \in L_{k,i}^e} \tilde{\pi}^e[\omega, q_k] \right\} \mu^i \left( \bigcup_{\substack{\sigma_r^j \in \Sigma_i^e \\ \delta(q_i, \sigma_r^j) = q_j}} \sigma_r^j L(G_j) \right) \\ &= \sum_{k \in I_Q} \left\{ \sum_{\omega \in L_{k,i}^e} \tilde{\pi}^e[\omega, q_k] \right\} \sum_{j \in I_Q} \left\{ \sum_{\substack{\sigma \in \Sigma_i^e \\ \delta(q_i, \sigma) = q_j}} \tilde{\pi}^e[\sigma, q_i] \right\} \mu_j \\ &= \sum_{k \in I_Q} \left\{ \sum_{\omega \in L_{k,i}^e} \tilde{\pi}^e[\omega, q_k] \right\} \sum_{j \in I_Q} \mathcal{P}(\Pi)_{ij} \mu_j \\ &= \sum_{k \in I_Q} \left\{ \sum_{\omega \in L_{k,i}^e} \tilde{\pi}^e[\omega, q_k] \right\} (\mathcal{P}(\Pi) \mu)_i \\ &= \left\{ \mathbb{D} \left( [\mathbf{I} - \mathcal{P}(\Pi)]^{-T} \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix} \right) \mathcal{P}(\Pi) \mu \right\}_i \\ &\quad (\text{See Appendix B}) \\ &= \left\{ \mathbb{D} \left( [\mathbf{I} - \mathcal{P}(\Pi)]^{-T} \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix} \right) \mathcal{P}(\Pi) [\mathbf{I} - \Pi]^{-1} \chi \right\}_i. \quad \square \end{aligned}$$

The proof of the theorem follows directly from Lemmas 1 and 2.  $\square$

**Remark 8:** Under complete observability,  $\mathcal{P}(\Pi) = 0$  and therefore,  $\vartheta(L_i^{11}) = \mu^i(L(G_i)) = \mu_i$ . Then, the set of strings observed to initiate from any given state  $q_i$  is the set of strings that do actually start from  $q_i$ . That is, under complete observability,  $L^1 = L(G_i)$  and  $L^{11} = \emptyset$ . Theorem 1 is consistent with this observation because the condition,  $\mathcal{P}(\Pi) = 0$ , holds under complete observability, and hence  $\vartheta(L^1 - L^{11}) = \mu^i(L(G_i))$ .

### 3. Quantification of unobservability

Given a plant model, described by the regular language  $L(G_i)$ , this section quantifies the effects of event unobservability at the supervisory level in terms of the following exact short sequence of monoids (Rotman 2002).

$$0 \rightarrow \mathfrak{U}_p \xrightarrow{t} L^c \xrightarrow{p} \mathcal{O}_p \rightarrow 0, \quad (23)$$

where  $t$  is the inclusion map and  $p$  is the natural projection onto  $\mathcal{O}_p$ . Exactness of the sequence results from the facts that  $t$  is injective,  $p$  is surjective and the kernel of  $p$  is the image under  $t$ , i.e.,  $\ker p = \text{Im } t$ . If this exact sequence is split, unobservability has no effect because each observed string would have a unique mapping back to an element of  $L^c$ . In general, the projection map  $p$  can be used to define an equivalence relation on  $L^c$  as follows:

$$\forall s_i, s_j \in L^c, ((p(s_i) = p(s_j)) \Rightarrow (s_i \sim s_j)). \quad (24)$$

This induces a partition  $\mathcal{P}$  on  $L^c$  with at most countable number of equivalence classes, and each equivalence class of  $\mathcal{P}$  has the property that every string in the equivalence class maps to the same element of  $\mathcal{O}_p$ . The cardinality bound on the number of equivalence classes is due to the fact that the number of strings in  $L^c$  itself is at most countable. Moreover, it follows that  $\forall s_j \in \mathcal{O}_p, p^{-1}(s_j) = \mathcal{P}_k$  for some  $k$ .

**Theorem 2:** *If the phantom language  $\mathbf{U}_p$  (see Definition 10) is a context free language (CFL), then  $\forall s_j \in \mathcal{O}_p, p^{-1}(s_j)$  is a context free sublanguage of  $L^c$ .*

**Proof:** Let  $s_j = j\sigma_1\sigma_2 \cdots \sigma_N$  for some  $j \in I_Q$ . Then,  $p^{-1}(s_j) = (\mathbf{U}_p\sigma_1 \circ \mathbf{U}_p\sigma_2 \circ \cdots \circ \mathbf{U}_p\sigma_N \circ \mathbf{U}_p) \cap L^c$ . Closure of context free languages under concatenation implies  $(\mathbf{U}_p\sigma_1 \circ \mathbf{U}_p\sigma_2 \circ \cdots \circ \mathbf{U}_p\sigma_N \circ \mathbf{U}_p)$  is a context free language. Then, closure under intersection implies  $p^{-1}(s_j)$  is context free as well. Detailed proofs of the closure properties are given in Hopcroft *et al.* (2001).

**Corollary 1:** *In case of regular unobservability,  $\forall s_j \in \mathcal{O}_p$ , the inverse image  $p^{-1}(s_j)$  is a regular sublanguage of  $L^c$ .*

**Definition 15:** The cumulative unobservability index  $\beta_{\text{cum}}$  for a given discrete event plant model under a specified unobservability situation is defined as follows:

$$\beta_{\text{cum}} = \sum_{s \in \mathcal{O}_p}^{\beta_{\text{cum}} \in \mathbb{R} \text{ such that}} \left| \sup_{\tau \in p^{-1}(s)} \vartheta(\tau) - \inf_{\tau \in p^{-1}(s)} \vartheta(\tau) \right|.$$

The cumulative unobservability index  $\beta_{\text{cum}}$  quantifies the maximum cumulative error one may incur from the viewpoint of language measure due to the presence of unobservable transitions.

The salient properties of  $\beta_{\text{cum}}$  are delineated below.

- $\beta_{\text{cum}} \in [0, \infty)$ . A specific upper bound is determined from the bounded total variation property of the language measure (Ray 2005).
- If the observation map  $p$  is injective, then  $\forall s \in \mathcal{O}_p, p^{-1}(s)$  is a singleton set and hence  $\beta_{\text{cum}} = 0$ . This is expected because it follows, from the short exact sequence, that an injective  $p$  implies  $\text{Im } t$  is trivial, i.e., the only unobservable string is the empty string  $\epsilon$ .

- $\beta = 0 \Rightarrow \forall s \in \mathcal{O}_p, \sup_{\tau \in p^{-1}(s)} \vartheta(\tau) = \inf_{\tau \in p^{-1}(s)} \vartheta(\tau)$ . From the bounded total variation of the language measure, it follows that, for  $\beta_{\text{cum}} = 0$ , if  $\text{Card}(p^{-1}(s))$  is countably infinite, then  $\vartheta(\tau) = 0 \forall \tau \in p^{-1}(s)$ , i.e.,  $p^{-1}(s)$  is a set of strings of zero measure. However, it is possible that  $\vartheta(\tau)$  has a non-zero value  $\forall \tau \in p^{-1}(s)$  if  $p^{-1}(s)$  is a finite set.

#### 4. Application examples

This section presents three examples to elucidate the concept of discrete event supervisory (DES) control under partial observation. These examples show how the theories, developed in §§ 2 and 3, can be applied to problems in different disciplines.

##### 4.1 Example 1: Language measure under partial observability

This example illustrates how the language measure under full observability (Ray 2005) is altered under partial observability. Figure 3 shows a deterministic finite state automaton (DFSA) with three states and an alphabet  $\Sigma = \{\alpha, \sigma\}$ , where the event  $\alpha$  is unobservable. The state transition cost matrix is as follows.

$$\Pi = \begin{bmatrix} 0 & 0.8 & 0.1 \\ 0.1 & 0 & 0.8 \\ 0.8 & 0.1 & 0 \end{bmatrix} \text{ and } \mathcal{P}(\Pi) = \begin{bmatrix} 0 & 0 & 0.1 \\ 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \end{bmatrix};$$

and assuming  $\chi = [1 \ 0 \ -1]^T$ , the language measures are given by

$$\mu = \begin{bmatrix} 0.7287 \\ -0.2834 \\ -0.4453 \end{bmatrix}; \quad \vartheta(L^1) = \begin{bmatrix} 0.8097 \\ -0.3149 \\ -0.4948 \end{bmatrix};$$

$$\vartheta(L^1 - L^{11}) = \begin{bmatrix} 0.8592 \\ -0.3959 \\ -0.4633 \end{bmatrix}.$$

The observed language  $\mathcal{O}_p$  has the regular expression  $I_Q\sigma^*$ , where the superscript  $*$  indicates Kleene closure.

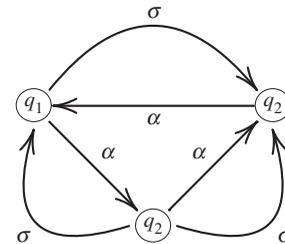


Figure 3. Finite state model with unobservable transitions.

Now, the unobservability index  $\beta_{cum}$  is calculated summing the unobservability  $\beta_i$  of individual subsets.

Considering the set of strings in  $p^{-1}(1\epsilon)$ , the maximum possible measure of a string is  $(0.2)^3 = 0.008$  achieved by the string  $\alpha\alpha\alpha$ , and the minimum possible measure is  $-0.2$  achieved by the singleton string  $\alpha$  and it weight  $\chi = -1$ . In the calculations that follow,  $\beta_i$  denotes the unobservability calculated from a subset  $\mathcal{O}_i \subseteq \mathcal{O}_p$  such that  $\forall s \in \mathcal{O}_i, c(s) = i$  (see Definition 5). Continuation of this process yields

$$\begin{aligned} \beta_1 &= (0.1 + 0.1^2) \times (0.8 + 0.8^4 + 0.8^7 + \dots) \\ &\quad + (1 + 0.1^2) \times (0.8^2 + 0.8^5 + 0.8^8 + \dots) \\ &\quad + (1 + 0.1) \times (1 + 0.8^3 + 0.8^6 + 0.8^9 + \dots) \\ &= 3.744. \end{aligned}$$

Similarly,  $\beta_2 = 3.771$  and  $\beta_3 = 3.421$ ; and sum of the three components together yields the unobservability index

$$\begin{aligned} \beta_{cum} &= 3.744 + 3.771 + 3.421 \\ &= 10.936. \end{aligned}$$

Calculations of the unobservability index for a general situation could be numerically cumbersome mainly due to computational complexity, which is a topic of future research.

#### 4.2 Example 2: Control of aircraft under partial observability

This example shows an engineering application on discrete event control of aircraft under partial loss of sensor information. It follows from the physics of flight mechanics that an aerofoil has a critical angle of attack at which the lift coefficient abruptly drops to zero as illustrated in figure 4. This is known as the stall

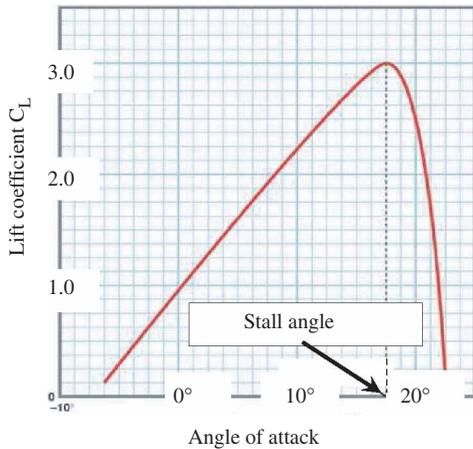


Figure 4. Profile of lift coefficient.

phenomenon that is, in general, a function of aerodynamic parameters. Hence, there is a limit to the steepness of the climb that a given aircraft can execute. A sensor malfunction may cause delay in recognition of an obstacle in the flight path. Consequently, it may become impossible for the pilot to avoid the obstacle without going into an accelerated stall, which may lead to rapid altitude loss and possible crash of the aircraft as displayed in figure 5.

Figure 6 depicts a finite state model of aircraft operation for obstacle avoidance, whose states and  $\chi$ -values, and events are listed in table 1 and table 2, respectively. The good marked states that have positive  $\chi$ -values are  $q_1$  and  $q_7$ ; the bad marked states that have negative  $\chi$ -values are  $q_5$  and  $q_6$ ; and the remaining states are unmarked and have zero  $\chi$ -values, implying that

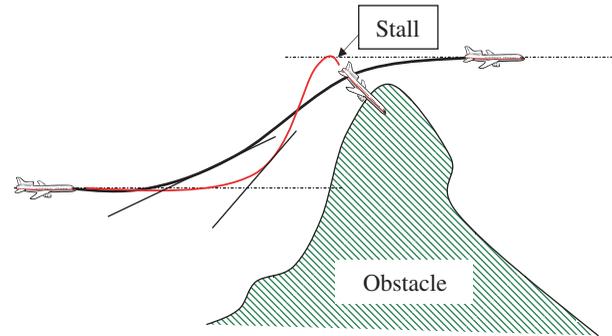


Figure 5. Obstacle avoidance failure caused by loss of observation.

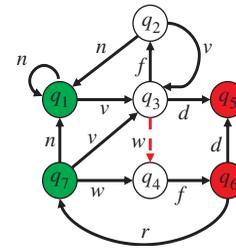


Figure 6. Finite state model of obstacle avoidance.

Table 1. List of plant states and corresponding  $\chi$ -values.

State	Description	$\chi$
$q_1$	Normal flight	0.2
$q_2$	Obstacle avoidance	0.0
$q_3$	Obstacle recognition	0.0
$q_4$	Obstacle too close	0.0
$q_5$	Aircraft destroyed	-1.0
$q_6$	Aircraft in accelerated stall	-0.9
$q_7$	Successful recovery from stall	0.05

termination on these states has no bearing on the flight performance. The event  $w$  from state  $q_3$  to  $q_4$  is unobservable (shown by a dashed line in figure 6), which relates to the physical situation of a delay in obstacle recognition or a sensor malfunction leading to incorrect obstacle distance reading. The event cost matrix  $\tilde{\Pi}$  is given in table 3. The state transition matrix and the language measures under complete and partial observability are computed based on the theories developed in §§2 and 3 and are shown in tables 4 and 5 respectively.

A comparison of the measures under perfect observation and partial observation reveals that the language of a plant operating mode may have a positive measure under loss of observability although the measure is actually negative. For example, the measure, with  $q_3$  (i.e., obstacle recognition) as the initial state, of the language under perfect observation is  $\mu_3 = -0.3006$  and, under loss of sensor information, it changes to  $\vartheta_3 = 0.0482$ . This is a false good representation of a truly bad scenario. Thus, a language-theoretic supervisory control under the assumption of perfect observation (Ray *et al.* 2004) may fail to take into account the effects of possible partial observation and thus make serious errors in decision making.

The cumulative unobservability index  $\beta_{cum}$  (see Definition 15) can be calculated from the following fact. An observed string  $s$  that has more than one point in its inverse image  $p^{-1}(s)$  must be of the form  $s = s_1 f$ , where  $s_1$  does not contain the event  $f$ . It is also noted

that  $p^{-1}(3f) = \{3f, 3wf\}$ . Hence, the cumulative unobservability index is

$$\beta_{cum} = \left\{ \sum_{i=1}^7 \mu^i(L(q_i, q_3)) \right\}_{\substack{\text{setting } \chi(q_i)=1 \\ \text{and deleting unobservable } w}} |\mu^3(wf)|$$

$$= \{4.3983\} \times |(0.3 \times 0.8 \times -0.9)|$$

$$= 0.95. \tag{25}$$

**4.3 Example 3: Pathogenic control under partial observability**

This example presents a simplified version of population control of pathogen within a living body. There are two antipathogens available,  $P_1$  for control of strain  $A$  and  $P_2$  for control of strain  $B$ . However,  $P_1$  acts as an accelerant for growth of strain  $B$  and similarly  $P_2$  acts as an accelerant for growth of strain  $A$ . Furthermore, the strains can mutate from one type to the other. Under these circumstances, it becomes critical to observe the specific mutations. A finite state model of pathogen population growth is given in figure 7 along with the lists of states and their respective  $\chi$ -values, and events in tables 6 and 7, respectively. The mutation events  $u$  are unobservable as shown by dashed arcs in figure 7. The event cost matrix and the state transition cost matrix for this specific case of pathogenic population control is given in tables 8 and 9 respectively. The language measures,  $\mu$  and  $\vartheta(L^1 - L^{11})$ , under complete

Table 2. List of events.

Event	Physical interpretation
$n$	Normal flight pattern recovered
$v$	Obstacle appeared ahead
$w$	Obstacle appeared close ahead
$f$	Initiating ascent to avoid obstacle
$r$	Recovery maneuvers initiated
$d$	Aircraft crashes

Table 3. Event cost matrix  $\tilde{\Pi}$  for obstacle avoidance.

$\tilde{\Pi}$	$n$	$v$	$w$	$f$	$r$	$d$
$q_1$	0.7	0.2	0.0	0.0	0.0	0.0
$q_2$	0.7	0.1	0.0	0.0	0.0	0.0
$q_3$	0.0	0.0	0.5	0.1	0.0	0.3
$q_4$	0.7	0.2	0.8	0.0	0.0	0.0
$q_5$	0.0	0.0	0.0	0.0	0.0	0.0
$q_6$	0.0	0.0	0.0	0.5	0.4	0.0
$q_7$	0.3	0.3	0.0	0.0	0.0	0.2

Table 4. State transition matrix  $\Pi$  for obstacle avoidance.

$\Pi$	$q_1$	$q_2$	$q_3$	$q_4$	$q_5$	$q_6$	$q_7$
$q_1$	0.7	0.0	0.2	0.0	0.0	0.0	0.0
$q_2$	0.7	0.0	0.1	0.0	0.0	0.0	0.0
$q_3$	0.0	0.5	0.0	0.3	0.1	0.0	0.0
$q_4$	0.0	0.0	0.0	0.0	0.0	0.8	0.0
$q_5$	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$q_6$	0.0	0.0	0.0	0.0	0.5	0.0	0.4
$q_7$	0.3	0.0	0.3	0.2	0.0	0.0	0.0

Table 5. Computed measures for obstacle avoidance.

States	$\mu$	$\vartheta(L^1 - L^{11})$
$q_1$	0.4663	0.4663
$q_2$	0.2963	0.2963
$q_3$	-0.3006	0.0482
$q_4$	-1.1625	-1.5112
$q_5$	-1.0000	-1.0000
$q_6$	-1.4531	-1.4531
$q_7$	-0.1328	-0.1328

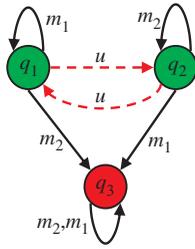


Figure 7. Finite state model for pathogen control.

Table 6. List of plant states and corresponding  $\chi$ -values.

State	Pathogen description	$\chi$
$q_1$	Stable population with strain A dominant	0.01
$q_2$	Stable population with strain B dominant	0.13
$q_3$	Population increasing at geometric rate	-1

Table 7. List of events.

Event	Description
$m_1$	Anti-pathogen $P_1$ administered
$m_2$	Anti-pathogen $P_2$ administered
$u$	Pathogen mutation

Table 8. Event cost matrix  $\tilde{\Pi}$  for pathogen control.

$\tilde{\Pi}$	$m_1$	$m_2$	$u$
$q_1$	0.60	0.01	0.30
$q_2$	0.02	0.40	0.40
$q_3$	0.40	0.40	0.00

Table 9. State transition matrix  $\Pi$  for pathogen control.

$\Pi$	$q_1$	$q_2$	$q_3$
$q_1$	0.60	0.30	0.01
$q_2$	0.40	0.40	0.02
$q_3$	0.00	0.00	0.80

observability and loss of observability, respectively, are shown in table 10.

Following the rationale in Example 2 (see §4.1), the measure with  $q_2$  (i.e., stable population with strain B) as the initial state, of the language under perfect observability is  $\mu_2 = -0.0333$  and, under loss of sensor information, it changes to  $\vartheta_2 = 0.0338$ . Hence, a DES control algorithm that does not take the effects of information loss into account may fail.

Table 10. Computed measures for pathogen control.

States	$\mu$	$\vartheta(L^1 - L^{11})$
$q_1$	-0.1250	-0.1921
$q_2$	-0.0333	0.0338
$q_3$	-5.0000	-5.0000

## 5. Summary and conclusions

Performance of DES control systems could be seriously affected by loss of information (e.g., due to sensor malfunctions or communication link failures). Ignoring the effects of unobservable transitions does not only affect the control system performance, but also may result in controller actions that are in direct contradiction to control objectives.

This paper addresses the problem by extending the signed real measure for regular languages, which has been developed under perfect observation (Ray 2005), i.e., if the supervisor receives the complete information on occurrence of events in the plant. The language measure under partial observability, reported in this paper, accounts for partial observability due to loss of information. Specifically, the language measure is modified to circumvent the detrimental effects resulting from loss of observability. The unobservability of events is statebased, implying that a given event may be observable at one state and unobservable at another state. The significant differences in language measures, under full observability and partial observability, are illustrated by physical examples.

Future work in this direction may involve research on modification of the existing theory of optimal control policies under full observation (Ray *et al.* 2004) to accommodate loss of observability and numerical computation of the unobservability index. The functional relationships of language measure with the amount of available information should also be investigated.

## Acknowledgements

This work has been supported in part by the US Army Research laboratory and the US Army Research Office under Grant No. DAAD19-01-1-0646.

The authors are grateful to the anonymous reviewers for their thoughtful comments suggesting critical changes that have certainly improved the quality of the paper.

## Appendix A

### Brief review of language measure

This section summarizes the signed real measure of regular languages; the details are reported in Ray (2005).

Let  $G_i \equiv (Q, \Sigma, \delta, q_i, Q_m)$  be a trim (i.e., accessible and co-accessible) finite-state automaton model that represents the discrete-event dynamics of a physical plant, where  $Q = \{q_k : k \in I_Q\}$  is the set of states and  $I_Q \equiv \{1, 2, \dots, n\}$  is the index set of states; the automaton starts with the initial state  $q_i$ ; the alphabet of events is  $\Sigma = \{\sigma_k : k \in I_\Sigma\}$ , having  $\Sigma \cap I_Q = \emptyset$  and  $I_\Sigma \equiv \{1, 2, \dots, \ell\}$  is the index set of events;  $\delta: Q \times \Sigma \rightarrow Q$  is the (possibly partial) function of state transitions; and  $Q_m \equiv \{q_{m_1}, q_{m_2}, \dots, q_{m_i}\} \subseteq Q$  is the set of marked (i.e., accepted) states with  $q_{m_k} = q_j$  for some  $j \in I_Q$ .

Let  $\Sigma^*$  be the Kleene closure of  $\Sigma$ , i.e., the set of all finite-length strings made of the events belonging to  $\Sigma$  as well as the empty string  $\varepsilon$  that is viewed as the identity of the monoid  $\Sigma^*$  under the operation of string concatenation, i.e.,  $\varepsilon s = s = s\varepsilon$ . The extension  $\delta^*: Q \times \Sigma^* \rightarrow Q$  is defined recursively in the usual sense (Ramadge and Wonham 1987).

**Definition A.1:** The language  $L(G_i)$  generated by a DFSA  $G$  initialized at the state  $q_i \in Q$  is defined as:

$$L(G_i) = \{s \in \Sigma^* | \delta^*(q_i, s) \in Q\}. \quad (26)$$

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

$$L_m(G_i) = \{s \in \Sigma^* | \delta^*(q_i, s) \in Q_m\}. \quad (27)$$

**Definition A.2:** For every  $q_j \in Q$ , let  $L(q_i, q_j)$  denote the set of all strings that, starting from the state  $q_i$ , terminate at the state  $q_j$ , i.e.,

$$L_{i,j} = \{s \in \Sigma^* | \delta^*(q_i, s) = q_j \in Q\}. \quad (28)$$

The set  $Q_m$  of marked states is partitioned into  $Q_m^+$  and  $Q_m^-$ , i.e.,  $Q_m = Q_m^+ \cup Q_m^-$  and  $Q_m^+ \cap Q_m^- = \emptyset$ , where  $Q_m^+$  contains all good marked states that we desire to reach, and  $Q_m^-$  contains all bad marked states that we want to avoid, although it may not always be possible to completely avoid the bad states while attempting to reach the good states. To characterize this, each marked state is assigned a real value based on the designer's perception of its impact on the system performance.

**Definition A.3:** The characteristic function  $\chi: Q \rightarrow [-1, 1]$  that assigns a signed real weight to state-based sublanguages  $L(q_i, q)$  is defined as:

$$\forall q \in Q, \quad \chi(q) \in \begin{cases} [-1, 0), & q \in Q_m^- \\ -\{0\} & q \notin Q_m \\ (0, 1], & q \in Q_m^+ \end{cases} \quad (29)$$

The state weighting vector, denoted by  $\chi = [\chi_1 \ \chi_2 \ \dots \ \chi_n]^T$ , where  $\chi_j \equiv \chi(q_j) \ \forall j \in I_Q$ , is called the  $\chi$ -vector. The  $j$ th element  $\chi_j$  of  $\chi$ -vector is the weight assigned to the corresponding terminal state  $q_j$ .

In general, the marked language  $L_m(G_i)$  consists of both good and bad event strings that, starting from the initial state  $q_i$ , lead to  $Q_m^+$  and  $Q_m^-$  respectively. Any event string belonging to the language  $L^0 = L(G_i) - L_m(G_i)$  leads to one of the non-marked states belonging to  $Q - Q_m$  and  $L^0$  does not contain any one of the good or bad strings. Based on the equivalence classes defined in the Myhill–Nerode Theorem, the regular languages  $L(G_i)$  and  $L_m(G_i)$  can be expressed as:

$$L(G_i) = \bigcup_{q_k \in Q} L_{i,k} \quad (30)$$

$$L_m(G_i) = \bigcup_{q_k \in Q_m} L_{i,k} = L_m^+ \cup L_m^- \quad (31)$$

where the sublanguage  $L_{i,k} \subseteq G_i$  having the initial state  $q_i$  is uniquely labelled by the terminal state  $q_k, k \in I_Q$  and  $L_{i,j} \cap L_{i,k} = \emptyset \ \forall j \neq k$ ; and  $L_m^+ \equiv \bigcup_{q_k \in Q_m^+} L_{i,k}$  and  $L_m^- \equiv \bigcup_{q_k \in Q_m^-} L_{i,k}$  are good and bad sublanguages of  $L_m(G_i)$ , respectively. Then,  $L^0 = \bigcup_{q_k \notin Q_m} L_{i,k}$  and  $L(G_i) = L^0 \cup L_m^+ \cup L_m^-$ .

A signed real measure  $\mu^i: 2^{L(G_i)} \rightarrow \mathbb{R} \equiv (-\infty, +\infty)$  is constructed on the  $\sigma$ -algebra  $2^{L(G_i)}$  for any  $i \in I_Q$ ; interested readers are referred to Ray (2005) for the details of measure-theoretic definitions and results. With the choice of this  $\sigma$ -algebra, every singleton set made of an event string  $s \in L(G_i)$  is a measurable set. By the Hahn decomposition theorem (Rudin 1988), each of these measurable sets qualifies itself to have a numerical value based on the above state-based decomposition of  $L(G_i)$  into  $L^0$ (null),  $L^+$ (positive), and  $L^-$ (negative) sublanguages.

In the following definition, each event is assigned a state-dependent cost that is conceptually similar to the conditional transition probability.

**Definition A.4:** The event cost of the DFSA  $G_i$  is defined as a (possibly partial) function  $\tilde{\pi}: \Sigma^* \times Q \rightarrow [0, 1]$  such that  $\forall q_i \in Q, \forall \sigma_j \in \Sigma, \forall s \in \Sigma^*$ ,

$$\tilde{\pi}[\sigma_j, q_i] = 0 \text{ if } \delta(q_i, \sigma_j) \text{ is undefined; } \tilde{\pi}[\varepsilon, q_i] = 1;$$

$$\tilde{\pi}[\sigma_j, q_i] \equiv \tilde{\pi}_{ij} \in [0, 1]; \quad \sum_{j \in I_\Sigma} \tilde{\pi}_{ij} < 1;$$

$$\tilde{\pi}[\sigma_j s, q_i] = \tilde{\pi}[\sigma_j, q_i] \tilde{\pi}[s, \delta(q_i, \sigma_j)].$$

Consequently, the  $n \times \ell$  event cost matrix is defined as:

$$\tilde{\Pi} = \begin{bmatrix} \tilde{\pi}_{11} & \tilde{\pi}_{12} & \dots & \tilde{\pi}_{1\ell} \\ \tilde{\pi}_{21} & \tilde{\pi}_{22} & \dots & \tilde{\pi}_{2\ell} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\pi}_{n1} & \tilde{\pi}_{n2} & \dots & \tilde{\pi}_{n\ell} \end{bmatrix} \quad (32)$$

**Definition A.5:** The state transition cost,  $\pi: Q \times Q \rightarrow [0, 1)$ , of the DFSA  $G_i$  is defined as follows:

$$\forall q_i, q_j \in Q, \pi_{ij} = \begin{cases} \sum_{\sigma \in \Sigma} \tilde{\pi}[\sigma, q_i], & \text{if } \delta(q_i, \sigma) = q_j \\ 0 & \text{if } \{\delta(q_i, \sigma) = q_j\} = \emptyset \end{cases} \quad (33)$$

Consequently, the  $n \times n$  state transition cost matrix is defined

$$\Pi = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1n} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{n1} & \pi_{n2} & \cdots & \pi_{nn} \end{bmatrix} \quad (34)$$

**Definition A.6:** Let  $\omega \in L(q_i, q_j) \subseteq 2^{L(G_i)}$ . The signed real measure  $\mu^i$  of every singleton string set  $\{\omega\}$  is defined as:

$$\mu^i(\{\omega\}) \equiv \tilde{\pi}(\omega, q_i) \chi(q_j). \quad (35)$$

The signed real measure of a sublanguage  $L_{i,j} \subseteq L(G_i)$  is defined as:

$$\mu_{i,j} \equiv \mu^i(L(q_i, q_j)) = \left( \sum_{\omega \in L(q_i, q_j)} \tilde{\pi}[\omega, q_i] \right) \chi_j. \quad (36)$$

Therefore, the signed real measure of the language of a DFSA  $G_i$  initialized at  $q_i \in Q$ , is defined as

$$\mu_i \equiv \mu^i(L(G_i)) = \sum_{j \in I_Q} \mu^i(L_{i,j}). \quad (37)$$

It is shown in Ray (2005) that the language measure in equation (37) can be expressed as

$$\mu_i = \sum_{j \in I_Q} \pi_{ij} \mu_j + \chi_i \quad (38)$$

The language measure vector, denoted as  $\mu = [\mu_1 \ \mu_2 \ \cdots \ \mu_n]^T$ , is called the  $\mu$ -vector. In vector form, equation (38) becomes

$$\mu = \Pi \mu + \chi \quad (39)$$

whose solution is given by

$$\mu = (\mathbf{I} - \Pi)^{-1} \chi \quad (40)$$

where the inverse in equation (40) exists because  $\Pi$  is a contraction operator (Ray 2005).

## Appendix B

### Measure of sets of strings terminating at a given state

The expression for the language measure in equation (40) can be restructured into the following form.

$$\begin{bmatrix} \mu_{1,1} & \mu_{1,2} & \cdots & \mu_{1,n} \\ \mu_{2,1} & \mu_{2,2} & \cdots & \mu_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{n,1} & \mu_{n,2} & \cdots & \mu_{n,n} \end{bmatrix} = [\mathbf{I} - \Pi]^{-1} \begin{bmatrix} \chi_1 & 0 & \cdots & 0 \\ 0 & \chi_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \chi_n \end{bmatrix}. \quad (41)$$

Hence the measure of the language  $L_i \equiv L(G_i)$  is given by

$$\begin{aligned} \mu_i &\equiv \sum_{j \in I_Q} \mu_{i,j} = \sum_{j \in I_Q} \left( \sum_{\omega \in L(q_i, q_j)} \tilde{\pi}[\omega, q_i] \right) \chi_j \\ &= \{[\mathbf{I} - \Pi]^{-1} \chi\}_i. \end{aligned} \quad (42)$$

Language measure, under partial observability, would require summation over the indices of initial states, in addition to summing over the indices of terminal states in equation (42).

Definition A.6 implies that  $\mu_i$  is the measure of the set of all strings that initiate from the state  $q_i$ . That is,  $\mu_i$  is obtained by summing up the  $i$ th row of the matrix on the left hand side of equation (41). However, the analysis developed in §2 also requires the computation of a closed form expression for the measure of the set of all strings terminating at a given state (say  $q_j$ ). This would require summing up the  $j$ th column of the matrix on the left hand side of equation (41). While for the initiating case, the result can be expressed as the matrix product in equation (40), the terminating case requires definition of the diagonalization map  $\mathbb{D}: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$  where

$$\mathbb{D} \left( \begin{bmatrix} \xi_1 \\ \vdots \\ \xi_n \end{bmatrix} \right) := \begin{bmatrix} \xi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \xi_n \end{bmatrix} \quad \forall \xi \in \mathbb{R}^n \quad (43)$$

and we have

$$\begin{aligned} \sum_{i \in I_Q} \mu_{i,j} &= \sum_{i \in I_Q} \left( \sum_{\omega \in L(q_i, q_j)} \tilde{\pi}[\omega, q_i] \right) \chi_j \\ &= \left\{ \mathbb{D} \left( [\mathbf{I} - \Pi]^{-T} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \right) \chi \right\}_j. \end{aligned} \quad (44)$$

Equation (44) follows from the fact that each diagonal entry of

$$\mathbb{D}\left([\mathbf{I} - \Pi]^{-T} \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix}\right)$$

is the corresponding column sum of  $[\mathbf{I} - \Pi]^{-1}$ . A simple example will clarify the point.

**Example B.1:** Assume the set of states  $Q = \{q_1, q_2\}$ . Let the state transition matrix be given by  $\Pi$  with

$$[\mathbf{I} - \Pi]^{-1} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \text{ and } \chi = \begin{bmatrix} \chi_1 \\ \chi_2 \end{bmatrix}. \quad (45)$$

Then it follows from equation (41),

$$\begin{bmatrix} \mu_{1,1} & \mu_{1,2} \\ \mu_{2,1} & \mu_{2,2} \end{bmatrix} = \begin{bmatrix} a_{11}\chi_1 & a_{12}\chi_2 \\ a_{21}\chi_1 & a_{22}\chi_2 \end{bmatrix} \quad (46)$$

The measure of the set of all strings terminating on states  $q_1$  and  $q_2$  is given by  $\mu_{1,1} + \mu_{2,1} = a_{11}\chi_1 + a_{21}\chi_1$  and  $\mu_{1,2} + \mu_{2,2} = a_{12}\chi_2 + a_{22}\chi_2$  respectively. And using the diagonalization map, we obtain

$$\mathbb{D}\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}^{-T} \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix}\right) = \begin{bmatrix} a_{11} + a_{21} & 0 \\ 0 & a_{12} + a_{22} \end{bmatrix} \quad (47)$$

$$\text{implying } \mathbb{D}\left([\mathbf{I} - \Pi]^{-T} \begin{Bmatrix} 1 \\ \vdots \\ 1 \end{Bmatrix}\right) \chi = \begin{Bmatrix} a_{11}\chi_1 + a_{21}\chi_1 \\ a_{12}\chi_2 + a_{22}\chi_2 \end{Bmatrix}. \quad (48)$$

## References

- J.E. Hopcroft, R. Motwani and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed., Boston, MA: Addison-Wesley, 2001.
- F. Lin and W.M. Wonham, "Decentralized control and coordination of discrete event systems with partial observation", *Inf. Sci.*, 44, pp. 199–224, 1988a.
- F. Lin and W.M. Wonham, "On observability of discrete-event systems", *Inf. Sci.*, 44, pp. 173–198, 1988b.
- 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, "Signed real measure of regular languages for discrete-event supervisory control", *Int. J. Control*, 78, pp. 949–967, 2005.
- A. Ray, J. Fu and C.M. Lagoa, "Optimal supervisory control of finite state automata", *Int. J. Control*, 77, pp. 1083–1100, 2004.
- J.J. Rotman, *Advanced Modern Algebra*, 1st ed., Englewood Cliffs, NJ: Prentice Hall, 2002.
- W. Rudin, *Real and Complex Analysis*, 3rd ed., New York: McGraw Hill, 1988.
- K. Wong, "On the complexity of projections of discrete-event systems", Technical report 9705, Systems Control Group, Department of Electrical Engineering, University of Toronto, 1997.
- K. Wong and W. Wonham, "Hierarchical control of discrete-event systems", *Discrete Event Dynamic Systems*, 6, pp. 241–273, 1996.
- K. Wong and W. Wonham, "The computation of observers in discrete-event systems", *Discrete Event Dynamic Systems*, 14, pp. 55–107, 2004.