Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Applied Mathematical Modelling 29 (2005) 597–613

APPLIED MATHEMATICAL MODELLING

www.elsevier.com/locate/apm

On-line identification of language measure parameters for discrete-event supervisory control ☆

Xi Wang, Asok Ray *, Amol M Khatkhate

Department of Mechanical Engineering, The Pennsylvania State University, 137 Reber Building, University Park, PA 16802, USA

Received 1 April 2004; received in revised form 20 October 2004; accepted 26 October 2004
Available online 20 December 2004

Abstract

The discrete-event dynamic behavior of physical plants is often represented by regular languages that can be realized as deterministic finite state automata (DFSA). The concept and construction of signed real measures of regular languages have been recently reported in literature. Major applications of the language measure are: quantitative evaluation of the discrete-event dynamic behavior of unsupervised and supervised plants; and analysis and synthesis of optimal supervisory control algorithms in the discrete-event setting. This paper formulates and experimentally validates an on-line procedure for identification of the language measure parameters based on a DFSA model of the physical plant. The recursive algorithm of this identification procedure relies on observed simulation and/or experimental data. Efficacy of the parameter identification procedure is demonstrated on the test bed of a mobile robotic system, whose dynamic behavior is modelled as a DFSA for discrete-event supervisory control.

Keywords: Discrete-event supervisory control; Formal languages; Parameter identification; Performance measure

☆ This work has been supported in part by Army Research Office (ARO) under Grant No. DAAD19-01-1-0646; and NASA Glenn Research Center under Grant No. NNC04GA49G.
* Corresponding author. Tel.: +1 814 865 6377; fax: +1 814 863 4848.
E-mail addresses: xxw117@psu.edu (X. Wang), axr2@psu.edu (A. Ray), amk303@psu.edu (A. M Khatkhate).

0307-904X/$ - see front matter
doi:10.1016/j.apm.2004.10.013

1. Introduction

The discrete-event dynamic behavior of physical plants is often modelled as regular languages that can be realized by deterministic finite state automata (DFSA) [3]. The (regular) sublanguage of a controlled plant could be different under different supervisors that are constrained to satisfy their respective specifications [7]. Such a partially ordered set of sublanguages requires a quantitative measure for total ordering of their respective performance. Two techniques for language measure computation have been recently reported. While the first technique [10] leads to a system of linear equations whose (closed form) solution yields the language measure vector, the second technique [8] is a recursive procedure with finitely many iterations. A sufficient condition for finiteness of the signed real measure has been established in both cases.

The language measure serves as a common quantitative tool to compare the performance of different supervisors and is assigned an event cost $\widetilde{\mathbf{\Pi}}$-matrix and a state characteristic $\mathbf{X}$-vector [10,8]. Event costs (i.e., elements of the $\widetilde{\mathbf{\Pi}}$-matrix) are analogous to the state-based conditional probabilities of the respective events; therefore, they are physical phenomena dependent on the plant behavior. On the other hand, the $\mathbf{X}$-vector is chosen solely relying on the designer's perception of the individual state's impact on the system performance. In the performance evaluation of both the unsupervised and supervised plant behavior, one of the critical parameters is the event cost matrix $\widetilde{\mathbf{\Pi}}$ that, in turn, determines the state transition cost matrix $\mathbf{\Pi}$. For example, Fu et al [2] have proposed state-based optimal control policies by selectively disabling controllable events to maximize the language measure of the controlled plant language. Furthermore, since the plant behavior is often slowly time-varying, there is a need for on-line parameter identification to generate up-to-date values of the $\widetilde{\mathbf{\Pi}}$-matrix within allowable bounds of errors.

This paper complements the earlier work [10] that reported theoretical formulation of a language measure without specifically stating how to obtain the underlying (e.g., event cost matrix $\widetilde{\mathbf{\Pi}}$-matrix) parameters. The main contribution of the present paper is development of a recursive procedure to identify the $\widetilde{\mathbf{\Pi}}$-matrix parameters of the language measure based on real-time experimentation and/or simulation. The recursive parameter estimation scheme permits on-line synthesis of optimal supervisory control [2] of discrete-event systems based on the language measure. The parameter identification procedure is supported by a stopping rule that determines the number of experiments to be conducted for a given bound of allowable estimation error.

The paper is organized in five sections including the present section. The language measure is briefly reviewed in Section 2 including introduction of the notations. Section 3 presents the main result—formulation of the algorithm for $\widetilde{\mathbf{\Pi}}$-matrix parameter identification and the associated stopping rules. Section 4 validates the algorithm through learning by simulation as well as by experimentation on the test bed of a mobile robotic system. The paper is concluded in Section 5 with a brief discussion on robustness of the identification procedure.

2. Brief review of language measure

This section first introduces the signed real measure of regular languages, which is reported in [10,8].

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

Let Σ^* be the Kleene closure of Σ, i.e., the set of all finite-length strings made of the events belonging to Σ as well as the empty string ϵ that is viewed as the identity of the monoid Σ^* under the operation of string concatenation, i.e., $\epsilon s = s = s\epsilon$. The extension $\hat{\delta} : Q \times \Sigma^* \to Q$ is defined recursively in the usual sense [3] [7]. For discrete event supervisory control [7], the event alphabet Σ is partitioned into sets, Σ_c and Σ_{uc} of controllable and uncontrollable events, respectively, where each event in Σ_c and no event in Σ_{uc} can be disabled by the supervisor.

Definition 2.1. The language $L(G_i)$ generated by a DFSA G initialized at the state $q_i \in Q$ is defined as

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

Definition 2.2. The language $L_m(G_i)$ marked by a DFSA G_i initialized at the state $q_i \in Q$ is defined as

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

The language $L(G_i)$ is partitioned as the non-marked and the marked languages, $L^o(G_i) \equiv L(G_i) - L_m(G_i)$ 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 further partitioned into Q_m^+ and Q_m^-, where Q_m^+ contains all *good* marked states that are desired to be terminated on and Q_m^- contains all *bad* marked states that one may not want to terminate on, although it may not always be possible to avoid the bad states while attempting to reach the good states. 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 $\mu : 2^{\Sigma^*} \to R \equiv (-\infty, \infty)$ 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 2.3. 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,

$$L(q_i, q_j) \equiv \{s \in L(G_i) : \hat{\delta}(q_i, s) = q_j\} \tag{3}$$

Definition 2.4. The characteristic function that assigns a signed real weight to state-partitioned sublanguages $L(q_i, q_j)$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, n$ is defined as: $\chi: Q \to [-1, 1]$ such that

$$\chi_j \equiv \chi(q_j) \in \begin{cases} [-1,0), & \text{if } q_j \in Q_m^- \\ \{0\}, & \text{if } q_j \notin Q_m \\ (0,1], & \text{if } q_j \in Q_m^+ \end{cases} \tag{4}$$

Definition 2.5. The event cost is conditioned on a DFSA state at which the event is generated, and is defined as $\tilde{\pi} : \Sigma^* \times Q \to [0,1]$ such that $\forall q_j \in Q\ \forall \sigma_k \in \Sigma$, $\forall s \in \Sigma^*$,

(1) $\tilde{\pi}[\sigma_k, q_j] \equiv \tilde{\pi}_{jk} \in [0,1)$; $\sum_k \tilde{\pi}_{jk} < 1$;
(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 is defined as

$$\widetilde{\mathbf{\Pi}} = \begin{bmatrix} \tilde{\pi}_{11} & \tilde{\pi}_{12} & \dots & \tilde{\pi}_{1l} \\ \tilde{\pi}_{21} & \tilde{\pi}_{22} & \dots & \tilde{\pi}_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{\pi}_{n1} & \tilde{\pi}_{n2} & \dots & \tilde{\pi}_{nl} \end{bmatrix} \tag{5}$$

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

An application of the induction principle to part (3) in Definition 2.5 shows $\tilde{\pi}[st, q_j] = \tilde{\pi}[s, q_j]\tilde{\pi}[t, \hat{\delta}(q_j, s)]$. The condition $\sum_k \tilde{\pi}_{jk} < 1$ provides a sufficient condition for the existence of the real signed measure as discussed in [9] along with additional comments on the physical interpretation of the event cost.

The next task is to formulate a measure of sublanguages of the plant language $L(G_i)$ in terms of the signed characteristic function χ and the non-negative event cost $\tilde{\pi}$.

Definition 2.6. 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

$$\mu(\{s\}) \equiv \tilde{\pi}(s, q_i)\chi(q_j) \quad \forall s \in L(q_i, q_j). \tag{6}$$

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

$$\mu\big(L(q_i, q_j)\big) \equiv \sum_{s \in L(q_i, q_j)} \mu(\{s\}) \tag{7}$$

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

$$\mu_i \equiv \mu(L(G_i)) = \sum_j \mu\big(L(q_i, q_j)\big) \tag{8}$$

Definition 2.7. The state transition cost, $\pi: Q \times Q \to [0,1)$, of the DFSA G_i is defined, for all q_i, $q_j \in Q$, as follows:

$$\pi_{ij} = \begin{cases} \sum_{\sigma \in \Sigma} \tilde{\pi}[\sigma, q_i], & \text{if } \delta(q_i, \sigma) = q_j \\ 0 & \text{if } \{\delta(q_i, \sigma) = q_j\} = \emptyset. \end{cases} \tag{9}$$

Consequently, the $n \times n$ state transition cost $\mathbf{\Pi}$-matrix is defined as

$$\mathbf{\Pi} = \begin{bmatrix} \pi_{11} & \pi_{12} & \dots & \pi_{1n} \\ \pi_{21} & \pi_{22} & \dots & \pi_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{n1} & \pi_{n2} & \dots & \pi_{nn} \end{bmatrix} \tag{10}$$

Wang and Ray [10] have shown that the measure $\mu_i \equiv \mu(L(G_i))$ 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 $\chi_i \equiv \chi(q_i)$. Equivalently, in vector notation:

$$\boldsymbol{\mu} = \mathbf{\Pi}\boldsymbol{\mu} + \mathbf{X} \tag{11}$$

where the measure vector $\boldsymbol{\mu} \equiv [\mu_1 \mu_2 \dots \mu_n]^{\mathrm{T}}$ and the characteristic vector $\mathbf{X} \equiv [\chi_1 \chi_2 \dots \chi_n]^{\mathrm{T}}$. Following Definitions 2.4 and 2.5, there exists $\theta \in (0,1)$ such that the induced infinity norm $\|\mathbf{\Pi}\|_\infty \equiv \max_i \sum_j \pi_{ij} = 1 - \theta$. The matrix operator $[\mathbf{I} - \mathbf{\Pi}]$ is invertible implying that the inverse $[I - \mathbf{\Pi}]^{-1}$ is a bounded linear operator [5] with its induced infinity norm $\|[\mathbf{I} - \mathbf{\Pi}]^{-1}\|_\infty \leqslant \theta^{-1}$. Therefore, the language measure vector can be expressed as

$$\boldsymbol{\mu} = [\mathbf{I} - \mathbf{\Pi}]^{-1}\mathbf{X} \tag{12}$$

where $\boldsymbol{\mu} \in \mathbf{R}^n$ and computational complexity [9] of the measure $\boldsymbol{\mu}$ is $O(n^3)$.

3. Estimation of language measure parameters

This section presents a recursive algorithm for identification of the language measure parameters (i.e., elements of the event cost matrix $\widetilde{\mathbf{\Pi}}$) (see Definition 2.5) which, in turn, allows computation of the state transition cost matrix $\mathbf{\Pi}$ (see Definition 2.7) and the language measure $\boldsymbol{\mu}$-vector in Eq. (12). It is assumed that the underlying physical process evolves at two different time scales. In the fast-time scale, i.e., over a short time period, the system is assumed to be an ergodic, discrete Markov process. In the slowly-varying time scale, i.e., over a long period, the system (possibly) behaves as a non-stationary stochastic process. For such a slowly-varying non-stationary process, it might be necessary to redesign the supervisory control policy in real time. In that case, the $\widetilde{\mathbf{\Pi}}$-matrix parameters should be updated at selected slow-time epochs.

3.1. A recursive parameter estimation scheme

Let p_{ij} be the transition probability of the event σ_j at the state q_i, i.e.,

$$p_{ij} = \begin{cases} P[\sigma_j \mid q_i], & \text{if } \exists q \in Q, \text{ s.t. } q = \delta(q_i, \sigma_j) \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

and its estimate be denoted by the parameter $\hat{p}_{ij}$ that is to be identified from the ensemble of simulation and/or experimental data.

Let a strictly increasing sequence of time epochs of consecutive event occurrence be denoted as:

$$\mathscr{T} \equiv \{t_k : k \in \mathbf{N}_0\} \tag{14}$$

where $\mathbf{N}_0$ is the set of non-negative integers. Let the indicator $\psi : \mathbf{N}_0 \times \mathscr{I}_Q \times \mathscr{I}_\Sigma \to \{0,1\}$ represent the incident of occurrence of an event. For example, if the DFSA was in state q_i at time epoch t_{k-1}, then

$$\psi_{ij}(k) = \begin{cases} 1, & \text{if } \sigma_j \text{ occurs at the time epoch } t_k \in \mathscr{T} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

Consequently, the number of occurrences of any event in the alphabet Σ is represented by $\Psi : \mathbf{N}_0 \times \mathscr{I}_Q \to \{0,1\}$. For example, if the DFSA was in state q_i at the time epoch t_{k-1}, then

$$\Psi_i(k) = \sum_{j \in \mathscr{I}_\Sigma} \psi_{ij}(k) \tag{16}$$

Let $n : \mathbf{N}_0 \times \mathscr{I}_Q \times \mathscr{I}_\Sigma \to \mathbf{N}_0$ represent the cumulative number of occurrences of an event at a state up to a given time epoch. That is, $n_{ij}(k)$ denotes the number of occurrences of the event σ_j at the state q_i up to the time epoch $t_k \in \mathscr{T}$. Similarly, let $N : \mathbf{N}_0 \times \mathscr{I}_Q \to \mathbf{N}_0$ represent the cumulative number of occurrences of any event in the alphabet Σ at a state up to a given time epoch. Consequently,

$$N_i(k) = \sum_{j \in \mathscr{I}_\Sigma} n_{ij}(k) \tag{17}$$

A frequency estimator, $\hat{p}_{ij}(k)$, for probability $p_{ij}(k)$ of the event σ_j occurring at the state q_i at the time epoch t_k, is obtained as

$$\begin{aligned} \hat{p}_{ij}(k) &= \frac{n_{ij}(k)}{N_i(k)} \\ \lim_{k\to\infty} \hat{p}_{ij}(k) &= p_{ij} \end{aligned} \tag{18}$$

Convergence of the above limit is justified because the occurrence of an event at a given state of a stationary Markov chain can be treated as an independent and identically distributed random variable.

A recursive algorithm of learning p_{ij} is formulated as a stochastic approximation scheme, starting at the time epoch t_0 with the initial conditions: $\hat{p}_{ij}(0) = 0$ and $n_{ij}(0) = 0$ for all $i \in \mathscr{I}_Q, j \in \mathscr{I}_\Sigma$; and $\Psi_i(0) = 0$ for all $i \in \mathscr{I}_Q$. Starting at $k = 0$, the recursive algorithm runs for $\{t_k: k \geqslant 1\}$. For example, upon occurrence of an event σ_j at a state q_i, the algorithm is recursively incremented as:

$$\begin{aligned} n_{ij}(k) &= n_{ij}(k-1) + \psi_{ij}(k) \\ N_i(k) &= N_i(k-1) + \Psi_i(k) \end{aligned} \tag{19}$$

Next it is demonstrated how the estimates of the language parameters (i.e., the elements of event cost matrix $\widetilde{\Pi}$) are determined from the probability estimates. The set of unmodelled events at state q_i, denoted by $\Sigma_i^u \forall i \in \mathscr{I}_Q$, accounts for the row-sum inequality: $\sum_j \tilde{\pi}_{ij} < 1$ (see Definition 2.5). Then, $P[\Sigma_i^u] = \theta_i \in (0,1)$ and $\sum_i \tilde{\pi}_{ij} = 1 - \theta_i$. An estimate of the (i,j)th element of the $\widetilde{\mathbf{\Pi}}$-matrix, denoted by $\hat{\tilde{\pi}}_{ij}$, is approximated as

$$\hat{\tilde{\pi}}_{ij}(k) = \hat{p}_{ij}(k)(1-\theta_i) \quad \forall j \in \mathscr{I}_\Sigma \tag{20}$$

Additional experiments on a more detailed automaton model would be necessary to identify the parameters $\theta_i \forall i \in \mathscr{I}_Q$. Given that $\theta_i \ll 1$, the problem of conducting additional experimentation can be circumvented by the following approximation:

A single parameter $\theta \approx \theta_i \forall i \in \mathscr{I}_Q$, $i \in \mathscr{I}_Q$, such that $0 < \theta \ll 1$, could be selected for convenience of implementation. From the numerical perspective, this option is meaningful because it sets an upper bound on the language measure based on the fact that the sup-norm $\|\mu\|_\infty \leqslant \theta^{-1}$. Note that each row sum in the $\widetilde{\mathbf{\Pi}}$-matrix being strictly less than 1, i.e., $\sum_j \tilde{\pi}_{ij} < 1$, is a sufficient condition for finiteness of the language measure.

Theoretically, $\tilde{\pi}_{ij}$ is the asymptotic value of the estimated probabilities $\hat{\tilde{\pi}}_{ij}(k)$ as if the event σ_j occurs infinitely many times at the state q_i. However, dealing with finite amount of data, the objective is to obtain a *good* estimate $\hat{p}_{ij}$ of p_{ij} from independent Bernoulli trials of generating events. Critical issues in dealing with finite amount of data are: (i) how much data are needed; and (ii) when to stop if adequate data are available. Section 3.2 addresses these issues.

3.2. Stopping rules for recursive learning

A stopping rule is necessary to find a lower bound on the number of experiments to be conducted for identification of the $\widetilde{\mathbf{\Pi}}$-matrix parameters. This section presents two stopping rules that are discussed below.

The first stopping rule is based on an inference approximation having a specified absolute error bound ε with a probability λ. The objective is to achieve a trade-off between the number of experimental observations and the estimation accuracy.

A bound on the required number of samples is estimated using the Gaussian structure for binomial distribution that is an approximation of the sum of a large number of independent and identically distributed (i.i.d.) Bernoulli trials of $\hat{\tilde{\pi}}_{ij}(t)$. The central limit theorem yields $\hat{\tilde{\pi}}_{ij} \sim \mathscr{N}\left(\tilde{\pi}_{ij}, \frac{\tilde{\pi}_{ij}(1-\tilde{\pi}_{ij})}{N}\right)$, where $\mathscr{N}$ indicates normal (or Gaussian) distribution with $E[\hat{\tilde{\pi}}_{ij}] \approx \tilde{\pi}_{ij}$ and $\mathrm{Var}[\hat{\tilde{\pi}}_{ij}] \equiv \sigma^2 \approx \frac{\tilde{\pi}_{ij}(1-\tilde{\pi}_{ij})}{N}$, provided that the number of samples N is sufficiently large. Let $\Delta = \hat{\tilde{\pi}}_{ij} - \tilde{\pi}_{ij}$, then $\frac{\Delta}{\sigma} \sim \mathscr{N}(0,1)$. Given $0 < \varepsilon \ll 1$ and $0 < \lambda \ll 1$, the problem is to find a bound N_b on the number N of experiments such that $P\{|\Delta| \geqslant \varepsilon\} \leqslant \lambda$. Equivalently

$$P\left\{\frac{|\Delta|}{\sigma} \geqslant \frac{\varepsilon}{\sigma}\right\} \leqslant \lambda \tag{21}$$

that yields a bound N_b on N as

$$N_\mathrm{b} \geqslant \left(\frac{\xi^{-1}(\lambda)}{\varepsilon}\right)^2 \tilde{\pi}_{ij}(1-\tilde{\pi}_{ij}) \tag{22}$$

where $\xi(x) \equiv 1 - \sqrt{\frac{2}{\pi}} \int_0^x \mathrm{e}^{-\frac{t^2}{2}} \mathrm{d}t$. Since the parameter $\tilde{\pi}_{ij}$ is unknown, one may use the fact that $\tilde{\pi}_{ij}(1-\tilde{\pi}_{ij}) \leqslant 0.25$ for every $\tilde{\pi}_{ij} \in [0,1]$ to (conservatively) obtain a bound on N only in terms of the specified parameters ε and λ as

$$N_b \geqslant \left(\frac{\xi^{-1}(\lambda)}{2\varepsilon}\right)^2 \tag{23}$$

The above estimate of the bound on the required number of samples is less conservative than that obtained from the Chernoff bound and is significantly less conservative than that obtained from Chebyshev bound that does not require the assumption of any specific distribution of Δ except for finiteness of the rth ($r = 2$) moment.

The second stopping rule, which is an alternative to the first stopping rule, is based on the properties of irreducible stochastic matrices. Following Eq. (18) and the state transition function δ of the DFSA, the state transition matrix is constructed at the kth iteration as $\mathbf{P}(k)$ that is an irreducible $n \times n$ stochastic matrix under stationary conditions. Similarly, the state probability vector $\mathbf{p}(k) \equiv [p_1(k)p_2(k)\ldots p_n(k)]$ is obtained by following Eq. (18):

$$p_i(k) = \frac{N_i(k)}{\sum_{j\in\mathscr{I}_Q} N_j(k)} \tag{24}$$

The stopping rule makes use of the Perron–Frobenius theorem [6] to establish a relation between the state probability vector $\mathbf{p}(k)$ and the irreducible stochastic matrix $\mathbf{P}(k)$. There is a unique unity eigenvalue of $\mathbf{P}(k)$ and the corresponding left eigenvector $\mathbf{p}(k)$ (normalized to unity in the sense of absolute sum) representing the state probability vector, provided that the matrix parameters have converged after a sufficiently large number of iterations. That is,

$$\|\mathbf{p}(k)(\mathbf{I} - \mathbf{P}(k))\|_\infty \leqslant \frac{1}{k} \to 0 \text{ as } k \to \infty$$

Equivalently,

$$\|(\mathbf{p}(k) - \mathbf{p}(k+1))\|_\infty \leqslant \frac{1}{k} \to 0 \text{ as } k \to \infty \tag{25}$$

Taking the expected value of $\|\mathbf{p}(k)\|_\infty$ to be $\frac{1}{n}$, a threshold of $\frac{\eta}{n}$ is specified, where n is the number of states and $0 < \eta \ll 1$ is a constant. A lower bound on the required number of samples is determined from Eq. (25) as

$$N_{\text{stop}} \equiv \text{Integer}\left(\frac{n}{\eta}\right) \tag{26}$$

based on the number of states n and the specified tolerance η.

4. Language parameter identification

This section presents on-line identification of the $\widetilde{\mathbf{\Pi}}$-matrix in a behavior-based mobile robotic system, Pioneer 2 AT, which is controlled by a discrete-event supervisor. The scenario for the robot movement is briefly delineated below.

(1) The robot explores an unknown environment by random search;
(2) The robot approaches a recognizable object upon detection by the camera;

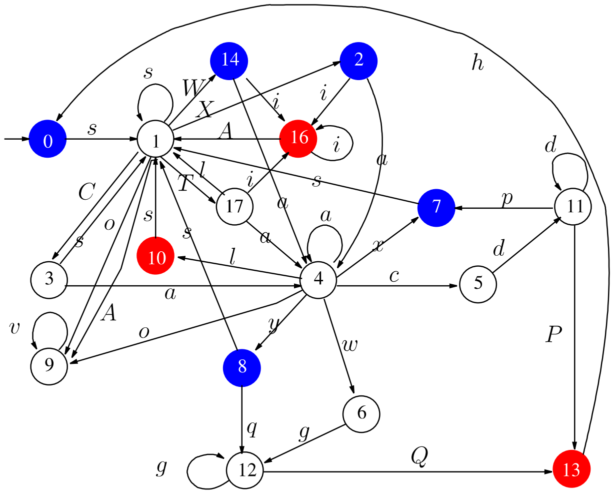

Fig. 1. DFSA plant model G for the experiment scenario.

(3) The robot grabs the object and searches for a recognizable goal;
(4) The robot approaches the destination upon detecting the object and releases the object at the destination;
(5) The robot keeps on repeating the same procedure unless commanded to stop.

Fig. 1 shows the deterministic finite state automaton (DFSA) model of the plant (i.e., the mobile robotic system). The corresponding alphabet Σ of discrete events is listed in Table 1. The set Q of states and the assigned state characteristic vector are listed in Table 2. The state transition cost matrix $\mathbf{\Pi}$ (which is derived from the event cost matrix $\widetilde{\mathbf{\Pi}}$) is listed in Table 3. The task is to identify the $\widetilde{\mathbf{\Pi}}$-matrix for the regular language generated by the DFSA plant model. The next subsection presents Matlab simulation of a typical scenario of the robotic system operation to demonstrate how the on-line identification of the $\widetilde{\mathbf{\Pi}}$-matrix works.

4.1. Parameter identification by robotic system simulation

This subsection presents a graphical design and development environment for hybrid system simulation using the Matlab simulink and stateflow, which is event-driven and is based on the theory of finite state machines. In this setting, the robot dynamics remains unaffected when the discrete-event supervisor changes its internal states; in other words, the continuous dynamics and discrete state space are decoupled.

The test facility allows on-line synthesis and verification of DES control policies using the language measure. The simulation block diagram in Fig. 2 shows the top level functions, in which the major blocks represent linear state space model of robot motion dynamics, obstacle avoidance using a simple sonar model, object detection by a simplified laser scanner model, and object recognition by a camera model.

Table 1
Event alphabet Σ for the Pioneer 2 AT robot

Event	Description	Controllable
a	Approach the object	√
A	Avoid obstacle successfully	
c	Reach goal with an object	
C	Find an object but gripper full	
d	Drop an object	√
g	Grab an object	√
h	Return to home	√
i	Ignore the current observed target	√
l	Lost the target	
o	Obstacle ahead	
p	Drop an object successfully	
P	Fail to drop an object	
q	Grab an object successfully	
Q	Fail to grab an object	
T	Find goal with an target	
s	Search recognizable target	√
v	Avoid obstacle	√
w	Reach target without an object	
W	Find object 1	
x	Lost the goal	
X	Find target 2	
y	Lost an object	

Table 2
State set Q of the plant model G and its **X**-vector

State	Description	χ
0	Robot ready for mission	0.0
1	Searching for target	0.0
2	Found object 2	0.1
3	Fount an object but gripper is full	0.0
4	Approaching target	0.0
5	Ready to drop object at destination	0.0
6	Ready to grab object	0.0
7	Drop an object successfully	0.8
8	Grab an object successfully	0.4
9	Avoiding obstacle	0.0
10	Lost target	−0.2
11	Dropping an object at destination	0.0
12	Grabbing an object	0.0
13	Failed grabbing or dropping	−1
14	Found object 1	0.3
17	Found the destination	0.0
16	Ignoring an object	−0.1

The simulation is initiated with robot's random walk. The robot dynamic model was obtained by system identification of experimental data collected from the continuous motion of the robot

Table 3
Π-matrix (17 × 17) for the discrete-event model *G*

π_{ij}	q_0	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	q_{13}	q_{14}	q_{17}	q_{16}
q_0	0	.9500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q_1	0	.4087	.0254	0	0	0	0	0	0	.4087	0	0	0	0	.0288	.0785	0
q_2	0	0	0	0	.4396	0	0	0	0	0	0	0	0	0	0	0	.5104
q_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q_4	0	0	0	0	0	.3760	0.3760	0	0	0	.1979	0	0	0	0	0	0
q_5	0	0	0	0	0	0	0	0	0	0	0	.95	0	0	0	0	0
q_6	0	0	0	0	0	0	0	0	0	0	0	0	.95	0	0	0	0
q_7	0	.9500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q_8	0	.9500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q_9	0	.4750	0	0	0	0	0	0	0	.4750	0	0	0	0	0	0	0
q_{10}	0	.9500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q_{11}	0	0	0	0	0	0	0	.95	0	0	0	0	0	0	0	0	0
q_{12}	0	0	0	0	0	0	0	0	.95	0	0	0	0	0	0	0	0
q_{13}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
q_{14}	0	0	0	0	.4495	0	0	0	0	0	0	0	0	0	0	0	.5005
q_{17}	0	0	0	0	.5200	0	0	0	0	0	0	0	0	0	0	0	.4300
q_{16}	0	.3484	0	0	0	0	0	0	0	0	0	0	0	0	0	0	.6016

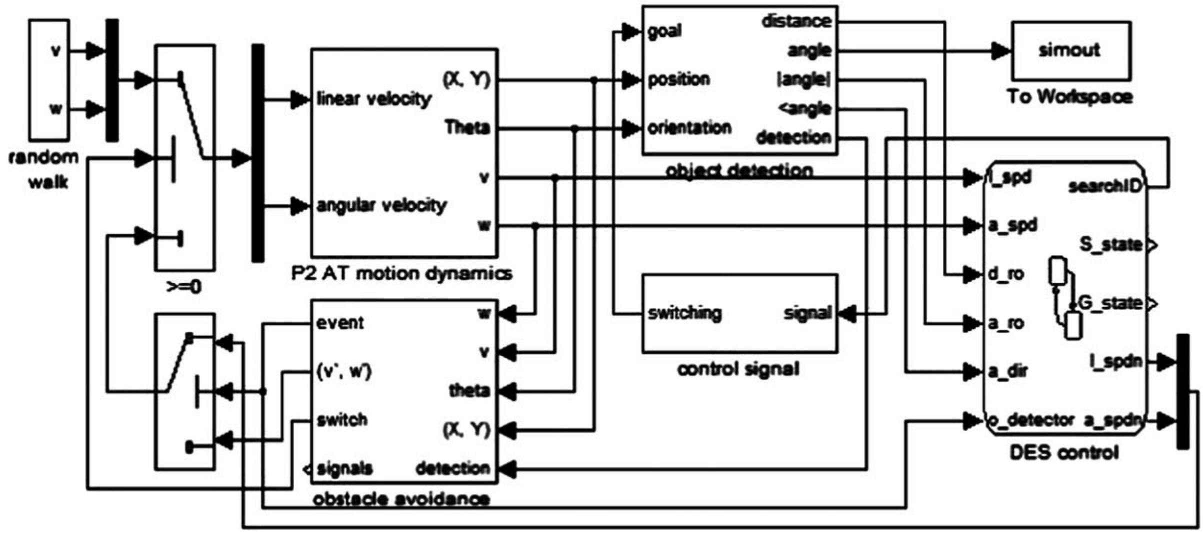

Fig. 2. Matlab robot simulation block diagram.

motion. The experiments for system identification of the robot dynamics were conducted with pseudo random inputs in both time and speed under the following constraints: (i) $2\ \text{s} \leqslant t_d \leqslant 10\ \text{s}$; (ii) $-300\ \text{mm/s} \leqslant v \leqslant 300\ \text{mm/s}$; (3) $-80\ \text{deg/s} \leqslant \omega \leqslant 80\ \text{deg/s}$, where t_d is the time duration in which the robot runs at constant speed; v and ω are the linear and angular velocities of the robot, respectively. The subspace technique for system identification of multivariable systems was applied using the `n4sid` routine in Matlab `IDENT` toolbox. (A multivariable autoregressive with external input (ARX) model was found to be inaccurate to capture the nonlinear dynamics.) The resulting state space model, having the reference and measured velocity vector $[v\omega]^T$ as the input and output, is given below.

$$x_{n+1} = Ax_n + Bu_n \tag{27}$$

$$y_n = Cx_n \tag{28}$$

$$A = \begin{bmatrix} 0.8579 & 0.0958 & -0.3488 & -0.0118 \\ -0.0012 & 0.6882 & -0.1532 & 0.0540 \\ 0.1415 & 0.3104 & 0.6644 & -0.5791 \\ -0.0893 & 0.2350 & 0.3603 & 0.6374 \end{bmatrix}$$

$$B = \begin{bmatrix} -0.3081 & -0.0037 \\ -0.0032 & 0.6524 \\ 0.4281 & -0.2781 \\ -0.1809 & -0.4719 \end{bmatrix}$$

$$C = \begin{bmatrix} -0.4808 & 0.2368 & -0.5129 & -0.2716 \\ 0.0138 & 0.4664 & 0.0547 & 0.2910 \end{bmatrix}$$

Fig. 3 shows that the results of the random walk experiment on the robotic system are in good agreement with those predicted by the model. Note that, in Fig. 3, the variations of the model predictions from the actual measured data are largely due to randomly varying environmental disturbances (e.g., floor friction).

Fig. 4 shows a typical trajectory of the supervised robot in a single mission cycle and Fig. 5 shows the DES control structure of the robotic system. A mission cycle is defined as the completion of the first four steps of the scenario. Four positions of the robot are shown: mission start point in the top left corner; the place where the robot grabs an object; the place where the robot drops the object; and the place where the robot begins its journey to return home. Note that when the robot sequentially detects an object, a goal, and its home, then its motion is further regulated by its supervisor to avoid potential collisions and to improve its performance.

Fig. 6 presents the results of on-line identification pertinent to some of the non-zero elements of the $\widetilde{\mathbf{\Pi}}$-matrix. It is seen that the transition probabilities indeed converge, possibly at different rates, in this structured environment. Environment disturbances have not yet been simulated. For example, since the lighting condition in the actual scene may vary over time and space, it is possible that robot loses the detected object during approaching. The next section investigates

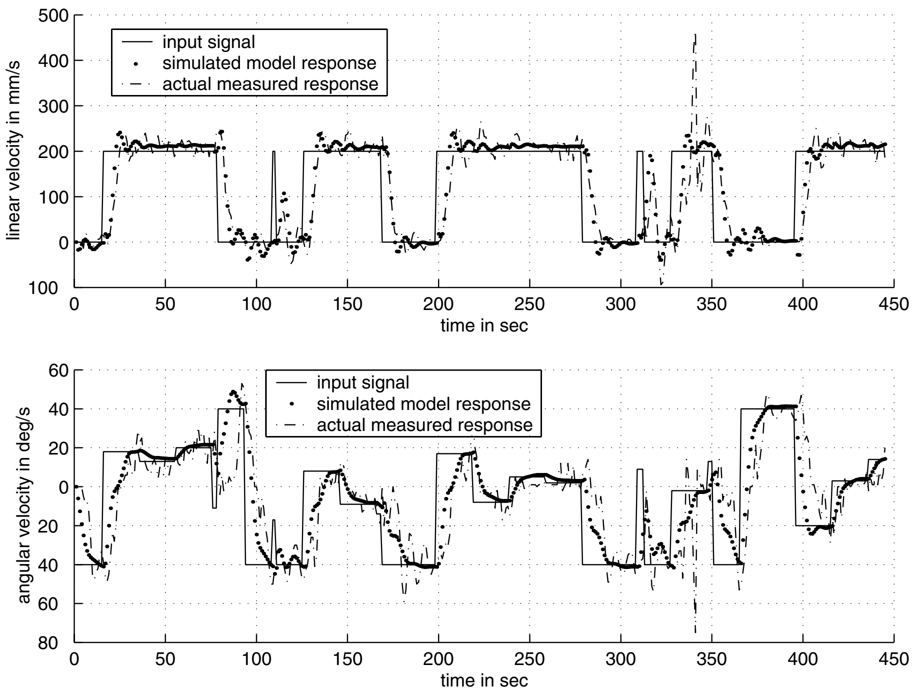

Fig. 3. System identification of the mobile robot.

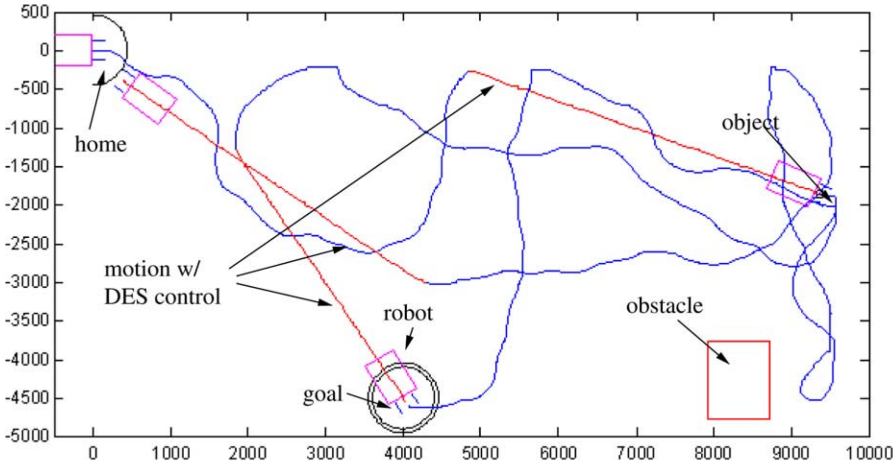

Fig. 4. A typical trajectory of the simulated robot.

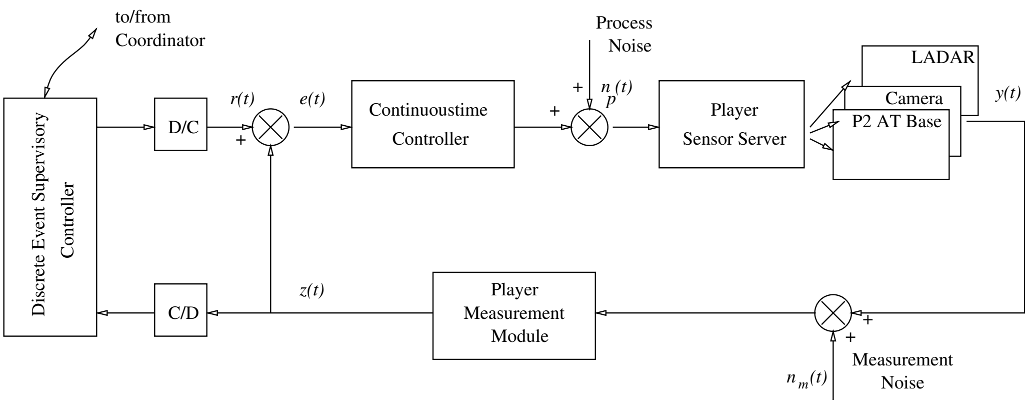

Fig. 5. DES control structure of the robotic system.

the impact of noise and disturbances effects on the convergence of $\widetilde{\mathbf{\Pi}}$-matrix learning based on real world experiments.

4.2. On-line parameter identification by robotic experiments

This section describes experiments on the robotic test bed for language measure identification. Fig. 5 shows the supervisory control system of a Pioneer 2 AT robot. A network device server, called *Player*, has been used to manage the robot's sensors and actuators. The main Pioneer 2 AT sensors include a Sony EVI D30 camera, a SICK LMS200 laser scanner, and Polaroid ultrasonic sonar sensors. The camera recognizes different targets, including the object, destination, and home base. The laser scanner is used to measure its distance from the target. The sonar and laser scanners together provide robust obstacle avoidance. The continuous controller, provided in the

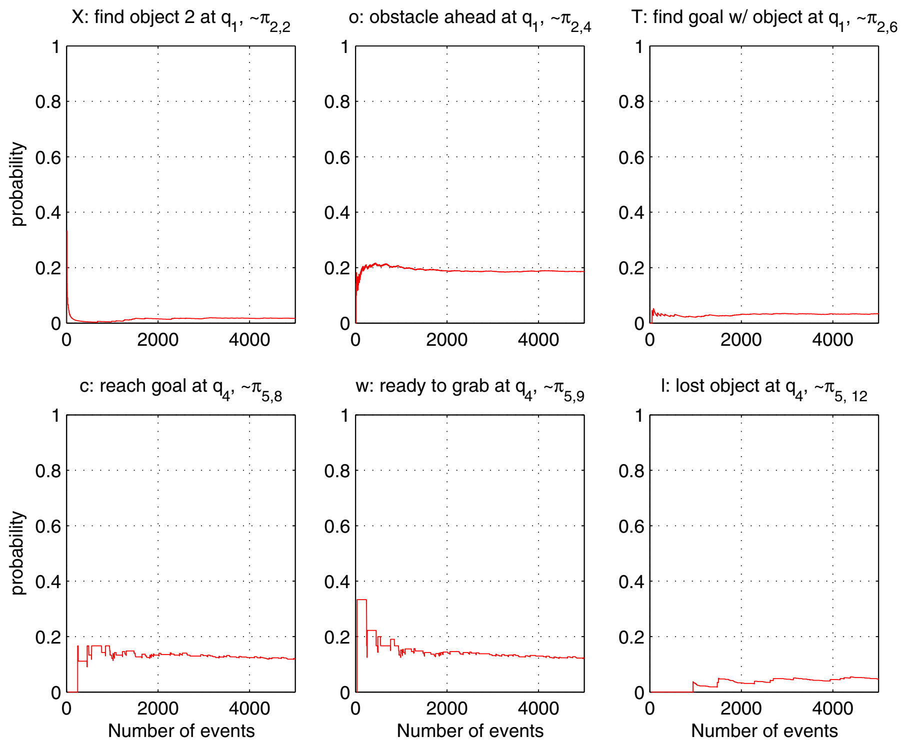

Fig. 6. Selected non-zero $\widetilde{\Pi}$ elements in simulation.

Pioneer 2 AT software system, sends desired speed signals to the robot through the Player while the sensor readings are relayed by Player's measurement module. Conceptually, the continuous-to-discrete (C/D) interface acts as the discrete-event generator and the discrete-to-continuous (D/C) interface converts controllable discrete events sent by robotic DES controller into continuous signals. The details of the DES control system that are not presented here to space limitations will be reported in a forthcoming publication.

Convergence of some of the non-zero elements of the $\widetilde{\Pi}$-matrix is shown in Fig. 7. The convergence rate in the experimental identification is often slower than that obtained in simulation due to the presence of environmental noise (e.g., floor friction and light intensity). Moreover, these elements may occasionally converge to different values because of the presence of spurious biased disturbances. Due to the noisy nature of the actual sensor readings, some of the events are generated more often than those in the simulation. While the simulation efforts prove the concept, it is the real-time experimental data that provides the information to the DES control system.

The characteristic vector $\mathbf{X}$ needs to be specified, as listed in Table 2, for the purpose of performance evaluation in terms of the language measure μ. A value of 0.4 is assigned to state 8 where the robot grabs an object successfully. The state, where the robot successfully drops an object at the destination and also represents the end of the current mission, is assigned a value of 0.8. The two colored objects are given different levels of importance; if the robot discovers a pink object, it receives a credit of 0.3, whereas a green object is assigned a smaller value of 0.1. Loss of an object and ignoring an object are assigned penalties of -0.2 and -0.1, respectively.

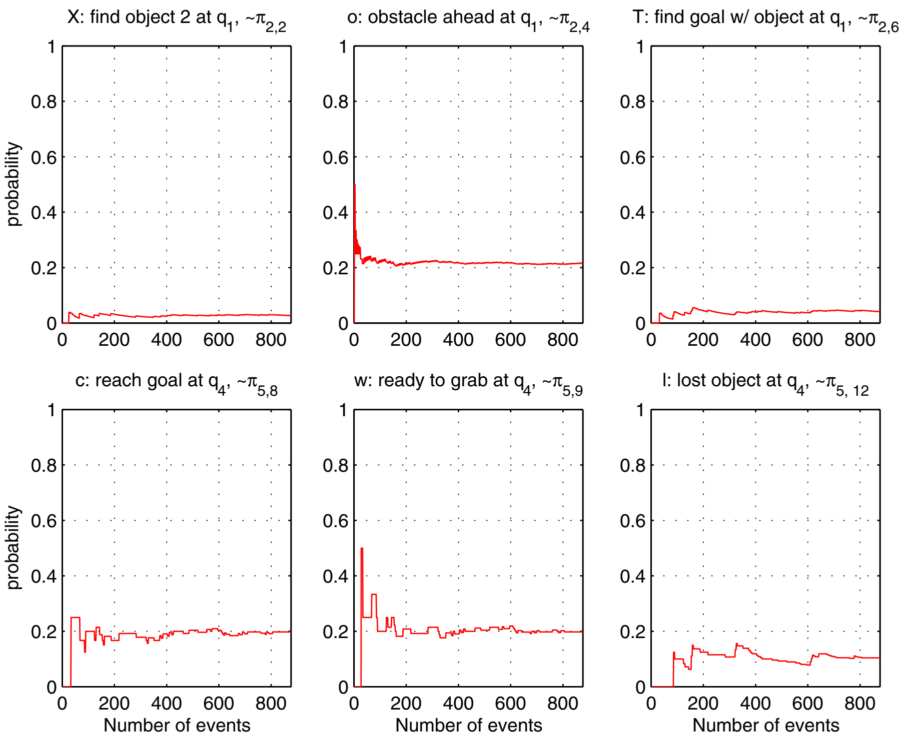

Fig. 7. Selected non-zero $\tilde{\Pi}$ elements in experiment.

5. Summary and conclusions

This paper presents a recursive procedure for on-line identification of the event cost matrix, called the $\tilde{\Pi}$-matrix, that is critical for analysis and synthesis of language-measure-based supervisory control systems. Given a bound parameter on identification error within a specified confidence level, the elements of the $\tilde{\Pi}$-matrix are shown to converge for a stationary process based on a stopping rule. It is demonstrated by both simulation studies and experimentation on a mobile robotic system that elements of the $\tilde{\Pi}$-matrix converge in a structured environment. For slowly varying non-stationary processes, the $\tilde{\Pi}$-matrix should be periodically updated for on-line synthesis of optimal control policies [2]. This is recommended as future research.

The identification procedure of the $\tilde{\Pi}$-matrix parameters is dependent on the event generation mechanism. Different event generators may cause the elements of $\tilde{\Pi}$-matrix converge to different transition probabilities. Ongoing efforts in this direction include formulation of event generation mechanisms using particle filters [1,4] to estimate conditional probability density distributions propagating over time.

Acknowledgement

The authors would like to thank Mr. Peter Lee for his assistance on robotic experiments.

References

[1] A. Doucet, S. Godsill, C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, Statist. Comput. 10 (3) (2000) 197–208.
[2] J. Fu, A. Ray, C.M. Lagoa, Unconstrained optimal control of regular languages, Automatica 40 (4) (2004) 639–648.
[3] J.E. Hopcroft, R. Motwani, J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, second ed., Addison-Wesley, Reading, MA, 2001.
[4] J.S. Liu, Monte Carlo Strategies in Scientific Computing, Springer-Verlag, New York, 2001.
[5] A.W. Naylor, G.R. Sell, Linear Operator Theory in Engineering and Science, Springer-Verlag, New York, 1982.
[6] R.J. Plemmons, A. Berman, Nonnegative Matrices in the Mathematical Science, Academic Press, New York, 1979.
[7] P.J. Ramadge, W.M. Wonham, Supervisory control of a class of discrete event processes, SIAM J. Contr. Optim. 25 (1) (1987) 206–230.
[8] A. Ray, S. Phoha, Signed real measure of regular languages for discrete-event automata, Int. J. Contr. 76 (18) (2003) 1800–1808.
[9] A. Surana, A. Ray, Signed real measure of regular languages, Demonst. Math. 37 (2) (2004) 485–503.
[10] X. Wang, A. Ray, A language measure for performance evaluation of discrete event supervisory control systems, J. Appl. Math. Model. 28 (9) (2004) 817–833.