2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Behavior Recognition in Mobile Robots Using Symbolic Dynamic Filtering and Language Measure★

Goutham Mallapragada **Asok Ray**
grm150@psu.edu axr2@psu.edu
Mechanical Engineering Department
The Pennsylvania State University
University Park, PA 16802, USA

***Abstract*—This paper addresses dynamic data-driven signature detection in mobile robots. The core concept of the paper is built upon the principles of Symbolic Dynamic Filtering (*SDF*) that has been recently reported in literature for extraction of relevant information (i.e., features) in complex dynamical systems. The objective here is to identify the robot behavior in real time as accurately as possible. Two different approaches to classifier design are presented in the paper; the first one is Bayesian and the second is based on measures of formal languages. The proposed methods have been experimentally validated on a networked robotic test-bed to detect and identify the type and motion profile of the robots under consideration.**

I. Introduction

Automated behavior recognition of robots is critical for multi-agent coordination and has become increasingly important with technological advancements in information processing and sensor networking. In such missions, a robotic platform may be required to make real-time decisions based on the collective behavior of other robots on a distributed network. This paper defines the behavior of a mobile robot system as statistical patterns of its evolutionary dynamics. Temporal changes in these statistical patterns occur over a slow time scale with respect to the fast time scale of robot dynamics [1].

The technical approach for feature selection presented in this paper, called the Symbolic Dynamic Filtering (*SDF*) [2], is built upon the concepts of Symbolic Dynamics [3], Finite State Automata and Pattern Recognition [4]. Partitioning of the space of robot dynamics yields an alphabet to obtain symbol sequences from time-series data. Then, the tools of computational mechanics are used to identify statistical patterns of these symbol sequences through construction of a probabilistic finite state machine (PFSM) for each symbol sequence. Transition probability matrices of the PFSM, obtained from the symbol sequences, capture the evolving pattern of the robot behavior in the slow scale. The statistical patterns (i.e., state probability histograms) that are derived from the respective state transition matrices are compared with an appropriate metric to discover how close a particular pattern is to a set of reference patterns. Two different methods for classifier design are presented in the paper. The first approach is based on the classical Bayesian classifier which is design to minimize the Bayes' risk function. The second approach for classifier design proposed in the paper is based on the recently reported measures of formal languages [5]. It is a new approach for classifier design that is faster than the classical Bayesian approach.

The major contribution of this paper is formulation of a dynamic data-driven method for signature detection in mobile robots, and its experimental validation on robotic agents in real-time. The novel part of the signature detection algorithm is pattern generation and identification in mobile robots by space partitioning of the time-series data. , where the theory of partitioning has been successfully developed and widely reported in the Physics and Applied Mathematics literature (e.g., see citations in [3]).

★This work has been supported in part the U.S. Army Research Office under Grant No. W911NF-07-1-0376. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

II. Pattern Identification of Mobile Robots

The objective of this paper is to identify the most likely pattern among a finite set of pre-determined patterns from the time-series data. Since Behavioral patterns of robots may vary due to, for example, variations in payload, type of drive-system, type of motion, and faults in the robot, the patterns are constructed as the state probability vectors of the D-Markov machine, described in [2].

Let $\Xi = \{\xi_i,\ i = 1, 2, \ldots, |\Xi|\}$ be defined as a (nonempty finite) collection of patterns, where $|\Xi|$ is the cardinality of Ξ. The reference time-series data $\boldsymbol{x}_i$ for each pattern $\xi_i \in \Xi$ is combined to generate a single time-series data $\boldsymbol{x}$. This combined time-series data $\boldsymbol{x}$ is then partitioned using the maximum entropy partitioning technique to generate the unique partition vector ζ. Using this partition vector ζ the reference time-series data $\boldsymbol{x}_i$ for each pattern is partitioned and a D-Markov machine M_i of appropriate depth is constructed. The steady state probability vector of each M_i is chosen as the reference probability distribution $\tilde{\boldsymbol{p}}_i$ for each pattern class.

We present two methods to construct a classifier based on the feature vector $\boldsymbol{p}$ which is the state probability vector corresponding to the time-series data $\boldsymbol{x}$. The first method is based on the Bayesian analysis and the second method is based on measures of formal languages.

A. Bayesian classifier

The Bayesian analysis in this paper is formulated by constructing an identity map between the set $\{\xi_1, \xi_2, \cdots, \xi_{|\Xi|}\}$ of pattern classes and the set $\{d_1, d_2, \cdots, d_{|\Xi|}\}$ of decisions on selection of reference probability distribution to compute the distance from. The objective here is to formulate a nonnegative real measure of these decisions for a given time series data.

Let $\boldsymbol{x}$ be a set of time series data that truly belongs to the pattern class ξ_j for some $j \in \{1, \cdots, |\Xi|\}$. Let the state probability distribution of the respective D-Markov machine be $\boldsymbol{p}_i(\boldsymbol{x}),\ i \in \{1, \cdots, |\Xi|\}$. The following definition formalizes the notion of deviation measure of a decision $d_i, i \in \{1, \cdots, |\Xi|\}$.

Definition 1: Given a time series data set $\boldsymbol{x}$ whose true pattern is ξ_j, *deviation measure* of the decision d_i is defined in terms of the respective state probability distribution $\boldsymbol{p}_j(\boldsymbol{x})$ and the reference probability distribution $\tilde{\boldsymbol{p}}_i$ as:

$$m_{ij}(\boldsymbol{x}) \triangleq d\left(\boldsymbol{p}_j(\boldsymbol{x}), \tilde{\boldsymbol{p}}_i\right) \quad (1)$$

where $d(\bullet, \bullet)$ is a distance function, e.g., the standard Euclidean norm of the difference between the distributions, $\boldsymbol{p}_j(\boldsymbol{x})$ and $\tilde{\boldsymbol{p}}_i$.

Due to uncertainties prevalent in the time-series data $\boldsymbol{x}$, the deviation measure $m_{ij}(\boldsymbol{x})$ in Eq. (1) would not be identically equal to

0, regardless of whether the decision d_i is correct or not. Nevertheless, it is expected that $m_{ij}(\boldsymbol{x})$ would be relatively small if d_i is the correct decision, i.e., $i = j$. This fact motivates the deviation measure to be treated as a random variable.

Let $\mathcal{M}_{ij}(\boldsymbol{x})$ denote the random variable associated with the deviation measure when the decision d_i that the data set $\boldsymbol{x}$ belongs to the pattern class ξ_i while $\boldsymbol{x}$ truly belongs to the j^{th} pattern class ξ_j. Then, realization of the random variable $\mathcal{M}_{ij}$ is the non-negative real $m_{ij}(\boldsymbol{x})$ in Eq. (1). Hence, for each pattern class ξ_j, there could be decisions $d_i, i = 1, 2, \cdots, |\Xi|$ that give rise to realizations of the random variables $\mathcal{M}_{ij},\ i = 1, 2, \cdots, |\Xi|$ as $m_{ij}(\boldsymbol{x}),\ i = 1, 2, \cdots, |\Xi|$; and there would be a total of $|\Xi| \times |\Xi| = |\Xi|^2$ random variables $\mathcal{M}_{ij}$. In the sequel, the probability distribution of $\mathcal{M}_{ij}$ is denoted as $p_{\mathcal{M}_{ij}}$.

The *a priori* conditional probability $P[\boldsymbol{x}|\xi_j, d_i]$ represents the probability of observation of the data $\boldsymbol{x}$ conditioned on the true pattern ξ_j and the decision d_i of choosing the reference probability vector $\tilde{\boldsymbol{p}}_i$ that represents the pattern ξ_i. That is,

$$P[\boldsymbol{x}|\xi_j, d_i] = p_{\mathcal{M}_{ij}}\big(m_{ij}(\boldsymbol{x})\big) \tag{2}$$

where $\mathcal{M}_{ij}$ is the random variable representing a decision d_i when the true pattern class is ξ_j; and the argument of the distribution $p_{\mathcal{M}_{ij}}$ is m_{ij} that is the deviation measure (see Eq. (1)) of the state probability vector of the data set $\boldsymbol{x}$ that truly belongs to the pattern class ξ_j obtained by choosing the reference state probability vector $\tilde{\boldsymbol{p}}_i$ based on decision d_i. The *a posteriori* probabilities are given as:

$$P[\xi_j|\boldsymbol{x}, d_i] = \frac{P[\boldsymbol{x}|\xi_j, d_i]P[\xi_j|d_i]}{P[\boldsymbol{x}|d_i]} \tag{3}$$

Equation (3) is expressed in a different form as:

$$P[\xi_j|\boldsymbol{x}, d_i] = \frac{P[\boldsymbol{x}|\xi_j, d_i]P[\xi_j]}{\sum_k P[\boldsymbol{x}|\xi_k, d_i]P[\xi_k]} \tag{4}$$

based on the following two assumptions.

- The pattern classes ξ_j's form a mutually exclusive and exhaustive set. It follows from the total probability theorem that
$$P[\boldsymbol{x}|d_i] = \sum_k P[\boldsymbol{x}|\xi_k, d_i]P[\xi_k|d_i]$$
- The prior probability of a pattern ξ_j is independent of the process of making the decision d_i, i.e.,
$$P[\xi_j|d_i] = P[\xi_j]$$

Substitution of Eq. (2) in Eq. (4) yields:

$$P[\xi_j|\boldsymbol{x}, d_i] = \frac{p_{\mathcal{M}_{ij}}\big(m_{ij}(\boldsymbol{x})\big)P[\xi_j]}{\sum_k p_{\mathcal{M}_{ik}}\big(m_{ij}(\boldsymbol{x})\big)P[\xi_k]} \tag{5}$$

Let the risk of making a decision d_i when truly the pattern class is ξ_j be specified as λ_{ij}. Then, the total risk of making a decision d_i becomes [4]:

$$R(d_i|\boldsymbol{x}) = \sum_{j=1}^{|\Xi|} \lambda_{ij} P(\xi_j|\boldsymbol{x}, d_i) \tag{6}$$

and the decision on pattern identification is made by minimizing the risk in Eq. (6) as:

$$d^\star = \underset{i}{\operatorname{argmin}}\ R(d_i|\boldsymbol{x}) \tag{7}$$

B. Language Measure classifier

This section presents an alternative method of classifier design that is significantly faster than the Bayesian classifier presented in Section II-A. The rationale is that the language measure classifier does not require estimation of probability distributions of the random variables $\mathcal{M}_{ij}$. The probabilistic finite state machine (*PFSM*) constructed from the symbol sequence acts as a language generator. A signed measure of the language [5], [6] is obtained by assigning a characteristic weight to each of the states of the constructed *PFSM*. In this setting, the task of classifier design amounts to choosing the characteristic weight vector for each pattern class; and the decision is based on maximizing the measure generated by different pattern classes. The concept is formally presented below.

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

The time-series data is modeled as a probabilistic finite state automaton ($PFSA$), which is a $DFSA$ augmented by the *characteristic function* and *event generation probabilities* [5], [6].

Definition 2: The characteristic function $\chi : Q \rightarrow [-1,\ 1]$ assigns a signed real weight to each state $q_i, i = 1, 2, \ldots, n$ such that relatively more positive weights are assigned to relatively more desirable states.

Definition 3: The event generation probabilities are specified as $\tilde{\pi} : \Sigma^\star \times Q \rightarrow [0,\ 1]$ such that $\forall q_j \in Q, \forall \sigma_k \in \Sigma, \forall s \in \Sigma^\star$,

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

The $\tilde{\boldsymbol{\Pi}}$-matrix is defined as: $\tilde{\boldsymbol{\Pi}}_{ij} = \tilde{\pi}(q_i, \sigma_j), q_i \in Q, \sigma_j \in \Sigma$.

Remark 1: The $\tilde{\boldsymbol{\Pi}}$-matrix is analogous to the morph matrix of a Markov chain in the sense that an element $\tilde{\pi}_{ij}$ represents the probability of the j^{th} event occurring at the i^{th} state.

Definition 4: The probabilistic state transition map of the probabilistic finite state automaton ($PFSA$) is defined as a function $\pi : Q \times Q \rightarrow [0,\ 1)$ such that

$$\pi(q_j,\ q_k) = \begin{cases} 0 \quad if\ \{\sigma \in \Sigma :\ \delta(q_j,\ \sigma) = q_k\} = \emptyset \\ \sum\limits_{\sigma \in \Sigma:\, \delta(q_j,\, \sigma) = q_k} \tilde{\pi}(\sigma,\ q_j) \triangleq\ \pi_{jk} \quad otherwise \end{cases} \tag{8}$$

The $\boldsymbol{\Pi}$-matrix, defined as $\boldsymbol{\Pi}_{ij} = \pi(q_i, q_j), q_i, q_j \in Q$.

Remark 2: The $\boldsymbol{\Pi}$-matrix is analogous to the state transition probability matrix of a Markov chain in the sense that an element π_{jk} is analogous to the transition probability from state q_j to state q_k. The language generated by a $PFSA$ is defined to be a probabilistic regular language.

The regular language generated by the $PFSA$ under consideration is a sublanguage of the Kleene closure Σ^* of the alphabet Σ.

Definition 5: The formal measure of the generated language of a $PFSA$ with respect to a defined characteristic weight vector [6] is defined as:

$$\boldsymbol{\nu}(\theta) = \theta\ [I - (1 - \theta)\boldsymbol{\Pi}]^{-1}\boldsymbol{\chi} \text{ with } \theta \in (0, 1) \tag{9}$$

Proposition 1: The limiting measure vector $\boldsymbol{\nu}(0) \triangleq \lim_{\theta \rightarrow 0^+} \boldsymbol{\nu}(\theta)$ exists and $||\boldsymbol{\nu}(0)||_\infty \leq 1$ (For proof see [6]).

Proposition 2: If the state transition matrix is irreducible, the renormalized measure vector is given by the expression $\boldsymbol{\nu}(0) = \nu \boldsymbol{e}$ where $\boldsymbol{e} \triangleq [1\ 1\ \dots\ 1]^T$. Then, the scalar renormalized measure ν is denoted as [6]

$$\nu = \boldsymbol{p}^T \boldsymbol{\chi} \tag{10}$$

The language-measure-based classifier is designed by computing a characteristic weight vector $\boldsymbol{\chi}$ for each pattern class $\xi_i \in \Xi$ from the reference probability vector $\{\tilde{\boldsymbol{p}}_i, i = 1, \dots, |\Xi|\}$. Given the reference probability vectors $\tilde{\boldsymbol{p}}_i$, let

$$\mathcal{H}_i \triangleq \left[\tilde{\boldsymbol{p}}_1 \dots \tilde{\boldsymbol{p}}_{i-1}\ \tilde{\boldsymbol{p}}_{i+1} \dots \tilde{\boldsymbol{p}}_{|\Xi|}\right] \tag{11}$$

Corresponding to each pattern ξ_i, the characteristic weight vector $\boldsymbol{\chi}_i$ is assigned as

$$\boldsymbol{\chi}_i = \left[I - \mathcal{H}_i(\mathcal{H}_i^T \mathcal{H})^{-1} \mathcal{H}^T\right] \tilde{\boldsymbol{p}}_i \tag{12}$$

Let $\boldsymbol{p}$ be the stationary probability vector of the constructed D-Markov machine for a data set $\boldsymbol{x}$ that is symbolized by using a given partition vector ζ. Then, the decision for pattern classification is made by maximizing the renormalized measure in Eq. (10) as

$$d^\star = \underset{i}{\operatorname{argmax}}\ \boldsymbol{p}^T \boldsymbol{\chi}_i \tag{13}$$

Computation of the characteristic weight in Eq. (12) relies on the concept of parity space [7], [8]. The projection matrix in Eq. (12) can be derived as

$$V_i^T V_i = \left[I - \mathcal{H}_i(\mathcal{H}_i^T \mathcal{H})^{-1} \mathcal{H}^T\right] \tag{14}$$

where V_i is known as the parity matrix for the pattern class ξ_i and the corresponding parity vector is $V_i \boldsymbol{p}$ for a data set that generates the stationary probability vector $\boldsymbol{p}$. The magnitude of this parity vector determines whether $\boldsymbol{p}$ belongs to the pattern class ξ_i as seen below.

$$\varrho_i = \|V_i \boldsymbol{p}\| = \begin{cases} \text{large, if } \boldsymbol{p} = \tilde{\boldsymbol{p}}_i \\ \text{small, if } \boldsymbol{p} \neq \tilde{\boldsymbol{p}}_i \end{cases} \tag{15}$$

Let $\theta_i \triangleq V_i \tilde{\boldsymbol{p}}_i$ and $\phi_i \triangleq V_i \boldsymbol{p}$ be the parity vectors of the reference probability vectors and the unknown probability vector respectively. Then, the following inner product is computed as

$$\langle \theta_i, \phi_i \rangle = \theta_i^T \phi_i = \tilde{\boldsymbol{p}}_i^T V_i^T V_i \boldsymbol{p} \tag{16}$$

By precomputation, it is possible to assign $\boldsymbol{\chi}_i = V_i^T V_i \tilde{\boldsymbol{p}}_i$. Then, Eq. (16) becomes

$$\langle \theta_i, \phi_i \rangle = \tilde{\boldsymbol{p}}_i^T V_i^T V_i \boldsymbol{p} = \boldsymbol{\chi}_i^T \boldsymbol{p} = \boldsymbol{p}^T \boldsymbol{\chi}_i \tag{17}$$

Equivalently, the the inner product $\langle \theta_i, \phi_i \rangle$ is the same as the renormalized measure defined in Eq. (10).

C. Algorithms for Behavior Identification

Let $\mathcal{R} = \{\rho_1, \dots, \rho_{|\mathcal{R}|}\}$ be the set of robots and let each robot execute one or more of the different motion profiles in the set $\Phi = \{\varphi_1, \dots, \varphi_{|\Phi|}\}$. Let the number of profiles executed by robot ρ_i be n_i. Also, let the indices of the profiles executed by robot ρ_i be $\{y_1^i, \dots, y_{n_i}^i\}$. That is robot ρ_i executes profiles $\{\varphi_{y_1^i}, \dots, \varphi_{y_{n_i}^i}\}$. Thus, the total number of pattern classes $|\Xi| = \sum_{ij} n_i \leq |\mathcal{R}||\Phi|$. The pattern identification procedure first generates a partitioning ζ and obtains a reference state probability vector $\tilde{\boldsymbol{p}}_j, i = 1, \dots, |\Xi|$ of time-series data belonging to each pattern class ξ_j. Algorithm 1 describes the procedure to compute the pattern vectors. Once the pattern set $\{\tilde{\boldsymbol{p}}_i\}$ is constructed, time-series data sets are analyzed to estimate the probability densities $p_{\mathcal{M}_{ij}}(\bullet)$ for the Bayesian classifier by following the procedure in Algorithm 2. Algorithm 3 gives the corresponding forward algorithm for the Language Measure classifier

Given a set of time-series data with an unidentified pattern, a symbol sequence is generated using the partitions ζ. Then, a D-Markov machine of appropriate depth D is constructed based on the procedure described in [2]. If the correct decision is made (i.e., the distance is computed from the correct reference probability vector), then the generated probability vector $\boldsymbol{p}$ should be very close to the chosen reference probability vector, implying that the deviation measure $m_{ij} \approx 0.0$ in equation (1). The *a priori* probabilities $p_{\mathcal{M}_{ij}}(m_{ij})$ are computed from the densities estimated in Algorithm 2. The *a posteriori* probabilities and the Bayes risk functions are then computed from $p_{\mathcal{M}_{ij}}(m_{ij})$ via Eqs. (5) and (6) respectively, as shown in lines 10-11 of Algorithm 5. The decision $d^\star$ is chosen so as to minimize the risk in line 12 of the Algorithm 5. Lines 13-14 simply convert the identified pattern vector index $d^\star$ into corresponding indices of the robot i and the movement profile y_ℓ^i. Algorithm 6 provides the analogous inverse or identification algorithm for the Language Measure classifier.

Algorithm 1 Pattern Identification Forward Algorithm

Input: time-series data sets
Output: Sequence of reference probability vectors $\tilde{\boldsymbol{p}}_i, i = 1, \dots, |\Xi|$
  Let $j = 0$;
  $\boldsymbol{x} = \emptyset$;
  for $i = 1$ to $|\mathcal{R}|$ **do**
    for $k = 1$ to n_i **do**
      Let robot $\rho_i \in \mathcal{R}$ execute motion $\varphi_{y_k^i} \in \Phi$;
      Collect the time-series data $\boldsymbol{x}_j$;
      $\boldsymbol{x} = \boldsymbol{x} \bigcup \boldsymbol{x}_j$;
      $j = j + 1$;
    end for
  end for
  Partition $\boldsymbol{x}$ using Alg. 4 to obtain partition vector ζ.
  for $i = 1$ to $|\Xi|$ **do**
    partition $\boldsymbol{x}_i$ using ζ to obtain the reference probability vector $\tilde{\boldsymbol{p}}_i$.
  end for

Algorithm 2 Forward Algorithm for Bayesian Classifier

Input: time-series data sets
Output: probability densities $p_{\mathcal{M}_{ij}}(\bullet)\ \forall\ i, j = 1, \cdots, |\Xi|$
  for $j = 1$ to $|\Xi|$ **do**
    for $\ell = 1$ to L **do**
      Collect time-series data $\boldsymbol{x}_j^\ell$
      for $i = 1$ to $|\Xi|$ **do**
        partition $\boldsymbol{x}_j^\ell$ using ζ to obtain symbol sequence $\boldsymbol{s}$
        construct D-Markov machine G using $\boldsymbol{s}$
        compute state probability vector $\boldsymbol{p}_j$ for G
        compute the deviation measure $m_{ij}^\ell = d(\boldsymbol{p}_j, \tilde{\boldsymbol{p}}_i)$
      end for
    end for
  end for
  From realizations $\left\{m_{ij}^1, \cdots, m_{ij}^L\right\}$ estimate the probability density for $p_{\mathcal{M}_{ij}}(\bullet)\ \forall\ i, j = 1, \cdots, |\Xi|$

III. Experimental Results and Discussion

This section provides a detailed description of the experimental procedure, an application of the *SDF* method to time-series data

Algorithm 3 Forward Algorithm for Language Measure Classifier

Input: time-series data sets
Output: characteristic weight vectors $\{\boldsymbol{\chi}_i\}$
  for $i = 1$ to $|\Xi|$ **do**
    compute $\mathcal{H}_i = \left[\tilde{\boldsymbol{p}}_1 \cdots \tilde{\boldsymbol{p}}_{i-1}\tilde{\boldsymbol{p}}_{i+1} \cdots \tilde{\boldsymbol{p}}_{|\Xi|}\right]$
    compute $\boldsymbol{\chi}_i = \left[I - \mathcal{H}_i(\mathcal{H}_i^T\mathcal{H})^{-1}\mathcal{H}^T\right]\tilde{\boldsymbol{p}}_i$
  end for

Algorithm 4 Maximum Entropy Partitioning

Input: time-series Data $\boldsymbol{x}$,Number of Symbols $|\mathcal{A}|$
Output: Partition Vector ζ
  sort $\boldsymbol{x}$ in ascending order
  let K=length($\boldsymbol{x}$)
  $\zeta(1) = \boldsymbol{x}(1)$; minimum of $\boldsymbol{x}$
  for i=2 to $|\mathcal{A}|$ **do**
    $\zeta(i) = \boldsymbol{x}\left(\mathit{floor}\left(\frac{(i-1)*K}{|\mathcal{A}|}\right)\right)$
  end for
  $\zeta(|\mathcal{A}|+1) = \boldsymbol{x}(K)$; maximum of $\boldsymbol{x}$

Algorithm 5 Inverse Algorithm for Bayesian Classifier

Input: time-series Data $\boldsymbol{x}$, Reference probability vectors $\{\tilde{\boldsymbol{p}}_i\}$, density estimates $\{p_{\mathcal{M}_{ij}}\}$
Output: Identified Pattern i.e decision $d^\star$
  partition $\boldsymbol{x}$ using ζ to get symbol sequence $\boldsymbol{s}$
  construct D-Markov machine G using $\boldsymbol{s}$
  compute state probability vector $\boldsymbol{p}_{j^\star}$ for G, where $j^\star$ corresponds to the unknown pattern $\xi_{j^\star}$ that is yet to be identified
  for $i = 1$ to $|\Xi|$ **do**
    compute the deviation measure $m_{ij} = d(\boldsymbol{p}_{j^\star}, \tilde{\boldsymbol{p}}_i)$
    for $j = 1$ to $|\Xi|$ **do**
      compute $P[\boldsymbol{x}|d_i, \xi_j] = p_{\mathcal{M}_{ij}}(m_{ij}(\boldsymbol{x}))$ (equation 2)
    end for
    for $j = 1$ to $|\Xi|$ **do**
      compute $P[\xi_j|\boldsymbol{x}, d_i]$ using Eq. (5)
    end for
    compute the Bayes risk $R(\boldsymbol{x}|d_i)$ using Eq. (6)
  end for
  compute $d^\star = \mathrm{argmin}_i\, R(\boldsymbol{x}|d_i)$
  from sequence $\{n_1, n_2, \ldots, n_{|\mathcal{R}|}\}$ compute the cumulative sequence $\{0, n_1, n_1 + n_2, \ldots, \sum_{i=1}^{i=|\mathcal{R}|} n_i\}$ to form the new sequence $\{N_0, N_1, N_2, \ldots, N_{|\mathcal{R}|}\}$
  find i such that $N_{i-1} < d^\star \leq N_i$ and $k = N_i - j$
  Conclude that the robot ρ_i was executing $\varphi_{y_k^i}$ profile

of robot signature, and discussion of the experimental results. The objective here is to identify the statistical patterns of robot behavior that may include both parametric and nonparametric uncertainties such as:

1) Small variations in the robot mass that includes unloaded base weights of the platform itself and its payload.
2) Uncertainties in friction coefficients for robot traction.
3) Fluctuations in the robot motion due to small delays in commands due to communication delays, computational delays especially if the processor is heavily loaded.
4) Sensor uncertainties due to random noise in the A/D channels of the microprocessor.

The Segway RMP and Pioneer 2AT robots are commanded to execute three different motion trajectories, namely, *random motion*,

Algorithm 6 Inverse Algorithm for language measure Classifier

Input: time-series Data $\boldsymbol{x}$, characteristic weight vectors $\{\boldsymbol{\chi}_i\}$
Output: Identified Pattern i.e decision $d^\star$
  partition $\boldsymbol{x}$ using ζ to get symbol sequence $\boldsymbol{s}$
  construct D-Markov machine G using $\boldsymbol{s}$
  compute state probability vector $\boldsymbol{p}_{j^\star}$ for G, where $j^\star$ corresponds to the unknown pattern $\xi_{j^\star}$ that is yet to be identified
  for $i = 1$ to $|\Xi|$ **do**
    compute the language measure $\nu_i = (\boldsymbol{p}_{j^\star})^T\boldsymbol{\chi}_i$
  end for
  compute $d^\star = \mathrm{argmax}_i\, \nu_i$
  from sequence $\{n_1, n_2, \ldots, n_{|\mathcal{R}|}\}$ compute the cumulative sequence $\{0, n_1, n_1 + n_2, \ldots, \sum_{i=1}^{i=|\mathcal{R}|} n_i\}$ to form the new sequence $\{N_0, N_1, N_2, \ldots, N_{|\mathcal{R}|}\}$
  find i such that $N_{i-1} < d^\star \leq N_i$ and $k = N_i - j$
  Conclude that the robot ρ_i was executing $\varphi_{y_k^i}$ profile

circular motion, and *square motion*.Both robots were made to execute each of the three different types of trajectories on a pressure sensitive floor consisting of 144 piezoelectric sensors in the laboratory environment for about an hour. Details of the laboratory setup can be found in [1]. This procedure was repeated to collect six data sets for two different robots executing each of the three different motion behaviors. It has been assumed that, during the execution of each motion, the statistical behavior of the robot is stationary and it does not switch behaviors in between.

Since the robot movements influence those sensors that surround its location, only a few sensors generate significantly higher readings than the remaining sensors. A simple background subtraction was used to eliminate the readings from sensors away from the robot's location. Since the spatial distribution of the pressure-sensitive coils underneath the floor is statistically homogenous, the decisions on detection of robot behavior patterns are statistically independent of the robot location (e.g., center of the circle, center and orientation of the square, and mean of the distribution for random motion). The parameters selected for the above three type of motion are presented below. For the two types of robots, a total of six different

Motion Type	**Parameter**	**Value**
Circular	Diameter	4m
Square	Edge length	3m
Random	Uniform distribution	range in x-dir 1 to 7 range in y-dir 1 to 4

behavior patterns are defined. The data sets for each of the six pattern classes are collected and processed to create the respective reference probability vectors $\tilde{\boldsymbol{p}}_i$, $i \in 1, \ldots, |\Xi|$, where $|\Xi| = 6$.

For all cases considered in this paper, the following options have been used in the *SDF* procedure for construction of D-Markov machines.

- *Partitioning Method*: Hilbert-transform-based analytical signal space partitioning ($ASSP$) [9];
- *D-Markov machine parameters*: Alphabet size $|\mathcal{A}|$=8 and depth $D = 1$;
- *Distance function for computation of deviation measure*: Standard Euclidean norm of the difference between the pair of patterns.

The above combination of the parameters $|\mathcal{A}|$ and D was adequate to successfully recognize all six behavioral patterns with only 8 states and was computationally very fast in the sense that the code execution time was orders of magnitude smaller than the process response time. Further increase of the alphabet size $|\mathcal{A}|$ did not provide any noticeable improvement in the results because a finer partitioning did not generate any significant new information as discussed in detail by Rajagopalan and Ray [10]. Increasing the value of D beyond 1 was also found to be ineffective, which increases the number of states of the finite state machine, many of them having near zero or zero probabilities and requires a larger data set for computational convergence of the state probability vectors.

A. Generation of statistical patterns for Bayesian classifier

A set of L=60 experiments were conducted to generate an ensemble of realizations for each of these random variables. To compute a realization m_{ij}^{ℓ}, where $\ell \in \{1 \ldots L\}$, the following procedure is adopted:

- Partition the ℓ^{th} data set for pattern j using partition vector ζ.
- Construct a *D*-Markov machine (of state cardinality less than or equal to $|\mathcal{A}|^D$) for each generated symbol sequence and compute the state probability vector $\boldsymbol{p}_j$.
- Compute the realization m_{ij}^{ℓ} as the distance between $\boldsymbol{p}_j$ and reference probability vector $\tilde{\boldsymbol{p}}_i$.

Thus, an ensemble consisting of L=60 realizations $\{m_{ij}^1, \ldots, m_{ij}^L\}$ was created for each random variable $\mathcal{M}_{ij}$. A two-parameter *lognormal* distribution was hypothesized for each random variable $\mathcal{M}_{ij}$. The rationale for selecting *lognormal* distribution of $\mathcal{M}_{ij}$, as opposed to other distributions (e.g., normal or Weibull), is stated below.

- The fact that the *lognormal* distribution is one directional on the position axis is consistent with the deviation measure which cannot be negative since it is a distance function.
- For a sample data set using the correct reference probability vector, the probability of deviation measure being extremely close to zero is less but higher for a certain range and gradually decreases as the deviation measure increases. This is easily modeled by a *lognormal* distribution.
- Since the random variable $\ell n(\mathcal{M}_{ij})$ is Gaussian, many standard statistical tools are available for statistical data analysis.

The probability density function of the random variable $\mathcal{M}_{ij}$ is defined as:

$$p_{\mathcal{M}_{ij}}(x) = \frac{1}{\sqrt{2\pi}\ \sigma_{ij}\ x}\ exp\left(\frac{-(ln(x)-\mu_{ij})^2}{2\sigma_{ij}^2}\right)\mathcal{U}(x) \qquad (18)$$

where $\mathcal{U}(\bullet)$ is the standard Heaviside unit step function; and μ and σ are respectively the mean and standard deviation of the Gaussian distributed random variable $ln(\mathcal{M}_{ij})$. The two parameters (i.e., mean μ_{ij} and the variance σ_{ij}^2 of $\mathcal{M}_{ij}$) for *lognormal* distribution were identified from each of the 36 sets, $\{m_{ij}^1, \ldots, m_{ij}^L\}$. Each *lognormal* distribution satisfied the 10% significance level which suffices for the conventional standard of 5% significance level. The goodness of fit of these histograms evinces that the *lognormal* distribution is an adequate approximation of the statistics of $\mathcal{M}_{ij}$.

B. Identification of Robot Type and Motion Profile

The problem at hand is to identify the type of the robot and its motion profile on the pressure-sensitive floor in the laboratory environment. Based on the acquired information of statistical patterns, a solution to the above identification problem was obtained through usage of Algorithms 5 and 6 from online time-series data of the unidentified robot (i.e., either Segway or Pioneer in the present experimentation). The time-series data was partitioned using a partition ζ obtained from Algorithm 1 to generate a symbol sequence. A *D*-Markov machine (with state cardinality $|\mathcal{A}|^D = 8$ for $|\mathcal{A}| = 8$ and $D = 1$) was constructed for the generated symbol sequence. The state probability vector was computed for the constructed state machine. Following Eq. (1), the deviation measure $m_{ij}(\boldsymbol{x})$ was computed for the probability vector $\boldsymbol{p}_i(\boldsymbol{x})$, and each reference vector $\tilde{\boldsymbol{p}}_i$ where $i \in \{1, \cdots, 6\}$ for a given set of time-series data $\boldsymbol{x}$ and the unknown pattern ξ_j is yet to be identified.

The pattern of robot type and motion was identified based on the probabilistic Bayesian method in Section II. The following assumptions were made in the absence of any specific information on $P[\xi_j]$ in Eq. (5) and λ_{ij} in Eq. (6).

- Uniform probability of the prior probabilities of occurrence of the pattern classes ξ_j's, i.e., $P[\xi_j] = \frac{1}{|\Xi|}\ \forall j \in \{1, \cdots, |\Xi|\}$.
- Uniform nonzero risk for all wrong decisions and zero risk for correct decisions, i.e., $\lambda_{ij} = 1-\delta_{ij}$ where $\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$ is the Kronecker-delta function.

With the above choices of λ_{ij}'s and $P[\xi_j]$'s, risk minimization in Eq. (7) is equivalent to having the maximum likelihood estimate of the pattern class as

$$\underset{i}{\text{argmax}}\left(P[\boldsymbol{x}|d_i,\xi_i]\right) = \underset{i}{\text{argmax}}\left(p_{\mathcal{M}_{ii}}(m_{ij}(\boldsymbol{x}))\right) \qquad (19)$$

TABLE I
DEVIATION MEASURES $m_{ij}(\boldsymbol{x})$ AND TOTAL RISK d_i

Decision d_i	$m_{ij}(\boldsymbol{x})$	**Total Risk** $R(d_i\|\boldsymbol{x})$
Segway Rand.	0.14831	0.9929
Segway Circ.	0.12076	1.0000
Segway Sq.	0.10643	0.9926
Pioneer Rand.	0.44743	0.9780
Pioneer Circ.	0.075726	0.7473
Pioneer Sq.	0.038984	**0.0419** (*see Eq.* (7))

The pertinent results for a given time series data $\boldsymbol{x}$ are summarized in Table I that lists the values of the deviation measures m_{ij} in the second column for an unidentified set of time-series data $\boldsymbol{x}$ which belongs to the class of *Pioneer Square* (i.e., ξ_6). Assuming the prior probabilities for all ξ_j are equal, Eq. (5) reduces to

$$P[\xi_j|\boldsymbol{x}, d_i] = \frac{p_{\mathcal{M}_{ij}}(m_{ij}(\boldsymbol{x}))}{\sum_k p_{\mathcal{M}_{ik}}(m_{ij}(\boldsymbol{x}))} \qquad (20)$$

The risk of of making decision d_i when the true hypothesis is ξ_j is chosen as $\lambda_{ij} = (1-\delta_{ij})$. With this choice of the risk parameters, the total risk of making the decision d_i given by Eq. (6) becomes

$$R(d_i|\boldsymbol{x}) = \sum_{j\neq i} P(\xi_j|\boldsymbol{x}, d_i) \qquad (21)$$

Table I shows the computed values of the total risk of making decision d_i in the third column. The decision $d^\star$ as given in Eq. (7), is the one that minimizes the total risk $R(d_i|\boldsymbol{x})$. The maximum likelihood estimate is simply the maximum of the diagonal elements of the matrix $\left[p_{\mathcal{M}_{ij}}(m_{ij})\right]$. In this example the maximum corresponds to *Pioneer Square*, shown in bold in Table I, which confirms the decision $d^\star$ obtained by minimizing the total risk in Eq. (21).

Figure 1 shows the results of identification of pattern types with the Bayesian classifier for 60 sets of time-series data for each pattern class. The numbers on the abscissa of Fig. 1 indicate the data corresponding to a particular pattern class. For all six patterns, the algorithms successfully identified the patterns for more than 60% of the cases studied for all pattern classes. From the statistical perspectives, it is expected that the success rate would improve if the number of samples in the goodness of fit analysis is increased (see Subsection III-A).

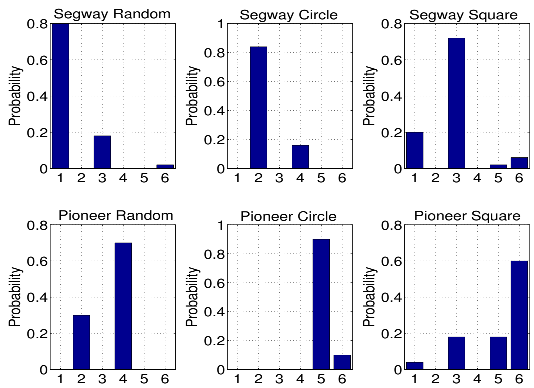

Fig. 1. Performance of robot behavior identification for different patterns using the Bayesian classifier;

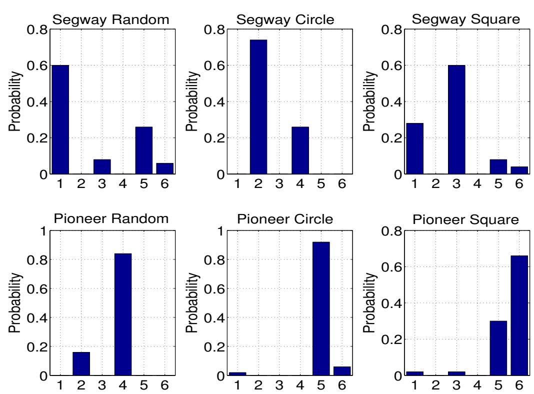

Fig. 2. Performance of robot behavior identification for different patterns using the language-measure-based classifier;

Figure 2 shows the results of identification of pattern types with the language-measure classifier for the same 60 data sets that were used for the Bayesian classifier. It shows that the algorithm successfully identified the correct pattern for more than 60% of the cases studied. The performance of the language-measure-based classifier is qualitatively similar to the Bayesian classifier in terms of error due to misclassification. However, the language-measure-based classifier is much simpler in terms of computational complexity as seen form the algorithms presented in Section II-C. The main computational advantage accrues from the fact that the language-measure-based classifier does not require estimation of the probability density functions for the random variables $\{p_{\mathcal{M}_{ij}}\}$. The following list compares the computation time, needed for feature extraction, classifier construction, and pattern identification, in the two classifiers.

Classifier	**Feature extraction**	**Classifier construction**	**Pattern identification**
Bayesian	407.09 sec	16.2677 sec	1.6076 sec
Language measure	407.09 sec	12.0406 sec	0.3861 sec

The amount of time required during the feature extraction is relatively large due to the volume of data that the algorithm processes in both classifiers. However, construction of the language measure classifier and subsequent pattern identification take less time than those for the Bayesian classifier.

IV. Summary and Conclusions

This paper presents an online dynamic data-driven method for identification of behavior patterns in autonomous agents such as mobile robots. The proposed method utilizes symbolic dynamic filtering (*SDF*) [2] to model the statistical behavior patterns of mobile robots. These identified models are then used to detect the pattern class of robot behavior (e.g., the type of robot and the kind of robot motion) in real-time based on the time-series data collected from an array of sensors. Two pattern identification methods are proposed and they have the following distinct features compared to standard statistical pattern recognition [4].

- *Fully automated model identification in the symbol space via coarse graining of the phase space*. This feature allows usage of relatively low-precision and inexpensive commercially available sensors.
- *Robustness to parametric and non-parametric uncertainties due to, for example, phase distortion and imprecise initial conditions*.
- *Insensitivity to environmental and spurious disturbances due to the inherent noise suppression capability of symbolic dynamic filtering* (*SDF*) [2], [10].

Further the language-measure-based classifier is seen to be computationally less expensive than the Bayesian classifier while they have comparable performance.

References

[1] G. Mallapragada, I. Chattopadhyay, and A. Ray, "Automated behavior recognition in mobile robots using symbolic dynamic filtering," *Proceedings of the I Mech E Part I: Journal of Systems & Control Engineering*, vol. 222 (I6), pp. 409–424, September 2008.

[2] A. Ray, "Symbolic dynamic analysis of complex systems for anomaly detection," *Signal Processing*, vol. 84, no. 7, pp. 1115–1130, 2004.

[3] D. Lind and M. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, Cambridge, UK, 1995.

[4] R. Duda, P. Hart, and D. Stork, *Pattern Classification, 2nd ed.* Wiley-Interscience, New York, NY, USA, 2001.

[5] A. Ray, "Signed real measure of regular languages for discrete-event supervisory control," *Int. J. Control*, vol. 78, no. 12, pp. 949–967, 2005.

[6] I. Chattopadhyay and A. Ray, "Renormalized measure of regular languages," *Int. J. Control*, vol. 79, no. 9, pp. 1107–1117, 2006.

[7] J. Potter and M. Suman, "Thresholdless redundancy management with arrays of skewed instruments," *Integrity in Electronic Flight Control Systems, AGRADOGRAPPH-224*, vol. 15, pp. 1–25, 1977.

[8] A. Ray and R. Luck, "An introduction to sensor signal validation in redundant measurement systems," *IEEE Control System Magazine*, vol. 11, no. 1, pp. 44–49, 1991.

[9] A. Subbu and A. Ray, "Space partitioning via Hilbert transform for symbolic time series analysis," *Applied Physics Letters*, vol. 92, no. 08, pp. 084107–1 to 084107–3, 2008.

[10] V. Rajagopalan and A. Ray, "Symbolic time series analysis via wavelet-based partitioning,," *Signal Processing*, vol. 86, no. 11, pp. 3309–3320, 2006.