

# Automated behaviour recognition in mobile robots using symbolic dynamic filtering

G Mallapragada, I Chattopadhyay, and A Ray\*

Mechanical Engineering Department, Pennsylvania State University, Pennsylvania, PA, USA

The manuscript was received on 25 February 2008 and was accepted after revision for publication on 22 April 2008.

DOI: 10.1243/09596518JSCE583

**Abstract:** This paper introduces a dynamic data-driven method for signature detection in mobile robots. The core concept of the paper is built upon the principles of symbolic dynamic filtering (SDF) that can be used to extract relevant information in complex dynamical systems. The objective here is to identify the robot behaviour in real time as accurately as possible. The proposed method is validated by experimentation on a networked robotics test bed to detect and identify the type and motion profile of the robots under consideration.

**Keywords:** robotic signatures, symbolic dynamic filtering, signature detection and identification

## 1 INTRODUCTION

Automated behaviour 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 behaviour of other robots on a distributed network.

This paper defines the behaviour of a mobile robot system as statistical patterns of its evolutionary dynamics. Temporal changes in these statistical patterns occur over a slow scale with respect to the fast scale of robot dynamics. The concept of two time scales in the analysis of dynamical systems is briefly discussed below in the context of statistical pattern analysis in mobile robotic systems.

**Definition 1.** The fast scale is defined to be the time scale over which the statistical properties of robot behaviour are assumed to remain invariant, i.e. the robot has statistically stationary dynamics.

**Definition 2.** The slow scale is defined to be the time scale over which the robotic system may exhibit non-stationary dynamics.

In view of Definition 1, the variations in the internal dynamics of the robot are assumed to be negligible on the fast scale. Since stationarity is assumed to be preserved on the fast scale, it is suitable for acquisition of sensor time series data. In view of Definition 2, an observable non-stationary behaviour is associated with the gradual evolution of anomalies (i.e. deviations from the nominal behaviour) in the robotic system. In general, a long time span in the fast scale is a tiny (i.e. several orders of magnitude smaller) interval in the slow scale. A pictorial view of the two-time-scales concept is presented in Fig. 1.

It is necessary to rely on sensor time series data because accurate and computationally tractable modelling of robot dynamics in changing environments is often infeasible solely based on fundamental principles of physics. The aim of this paper is to identify robot behaviour (i.e. relevant statistical patterns) in real time. The pattern identification algorithms are validated on an experimental facility equipped with a distributed array of piezoelectric pressure sensors that monitor the environment of

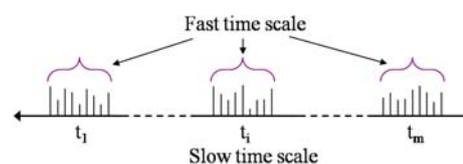


Fig. 1 Pictorial view of two time scales

\*Corresponding author: Mechanical Engineering Department, Pennsylvania State University, 329 Reber Building, University Park, PA16802, USA. email: axr2@psu.edu

robot movements. The problem is to capture the statistical patterns that represent the changing behaviour of robot motion in the slow scale based on time series data generated in the fast scale.

The technical approach presented in this paper, called symbolic dynamic filtering (SDF) [1], is built upon the concepts of symbolic dynamics [2], finite state automata [3], 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 [1, 5] 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 behaviour 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.

Causal-state splitting reconstruction (CSSR) [6] is another symbolic pattern identification method. A common feature of CSSR and SDF is that both of them belong to the class of symbolization-based hidden Markov models [7] where PFSM models are constructed from symbol sequences in each case. However, CSSR belongs to the class of sofic shift [2] and has an *a priori* unknown structure; and yields optimal pattern discovery in the sense of mutual information [8]. In contrast, SDF belongs to the more restricted class of shift of finite type [2] and has an *a priori* known structure that can be freely chosen. Although the generated patterns are sub-optimal, SDF provides an algebraic structure of the probabilistic state machine, where each state has physical significance that is invariant under a wide class of changes in statistical patterns of symbol sequences. This feature of SDF allows unambiguous comparison of different behaviours from the state probability histograms of the underlying state machines. Furthermore, SDF is computationally faster than CSSR because a significantly fewer number of floating point arithmetic operations are required.

Conventional hidden Markov models (HMM) (i.e. directly generated from time series without symbolization) have been adopted in [9] in the context of Robotic Soccer to identify the behaviour of a robot, where the states and the structure of the HMM are fixed *a priori* and the state transition and observation probabilities are identified using the Baum–Welch algorithm [7]. The SDF approach, adopted in

this paper, is computationally more efficient in the sense that, upon selection of the symbol alphabet, it is not necessary to identify the states and the state machine structure; the SDF algorithm identifies only the state transition probabilities from the ensemble of (fast scale) sensor time series data. Another remarkable feature of SDF is robustness (i.e. insensitivity to spurious noise and exogenous disturbances) due to the following reasons.

1. *Inherent ‘coarse graining’ [10] of the time series data in the process of space partitioning*: This phenomenon is a consequence of maximum entropy partitioning that automatically makes the partition segments coarser in regions of low data density and finer in regions of high data density [11, 12].
2. *Small variance in estimated parameters of the probability distribution [13]*: This phenomenon is a consequence of repeated recurrences of paths in the graph of the finite state machine with a relatively small number of states and a very large number of sample points in the (fast scale) time series data [14].

In contrast to SDF, the conventional HMM approach can be sensitive to variations in initial conditions especially if the resulting model is initially off-phase with the current behaviour. Another alternative approach is dynamic Bayesian networks [15] that have been used to identify and track targets primarily in military applications. Here, the state machines are hand-coded and the probabilities are estimated with expectation-maximization algorithms that are computationally much slower than SDF.

The major contribution of this paper is the 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 [2, 10, 11, 16]).

The paper is organized in six sections. Section 2 lays down the basic framework of the analysis and briefly reviews the theory of SDF. Section 3 formulates the pattern recognition problem in mobile robots and presents a solution to the problem under consideration. This section also provides the necessary algorithms that are implemented in this paper.

Section 4 describes the experimental facility used in this paper along with some details of both hardware and software. Section 5 discusses the experimental results. The paper is summarized and concluded in section 6 along with recommendations for future research.

## 2 SYMBOLIC DYNAMIC FILTERING (SDF)

This section reviews the theory and salient features of SDF based on the authors' previous work that has been recently reported in the literature [1, 12, 17, 18]. The core concept of SDF is built upon the fundamental principles of finite state automata, pattern recognition, and information theory, and relies on the following two basic assumptions.

1. The system behaviour is quasi-stationary at the *fast time scale* of process dynamics.
2. Observable non-stationary behaviour of the dynamical system can be associated with parametric or non-parametric changes evolving at a *slow time scale*.

Along this line, behaviour identification in robotic platforms is formulated as a two-time-scale problem. The robot motion process is assumed to have quasi-stationary dynamics on the fast scale and any observable non-stationary behaviour is associated with parametric or non-parametric changes occurring on the slow scale. In other words, the variations in the internal dynamics of the robotic system are assumed to be negligible on the fast scale, while pattern changes due to internal anomalies or exogenous environmental variations may become significant on the slow scale.

### 2.1 Space partitioning for symbol generation

This subsection describes the concept of symbol sequence generation from time series data, which is an essential ingredient of SDF. A data sequence (e.g. time series data) is converted to a symbol sequence by partitioning a compact (i.e. closed and bounded) region  $\Omega$  of the (finite-dimensional) phase space, over which the trajectory evolves, into finitely many discrete blocks. As illustrated in Fig. 2, the blocks  $\Phi_1, \Phi_2, \dots, \Phi_m$  represent a partitioning of  $\Omega$  such that it is an exhaustive and mutually exclusive set, that is

$$\bigcup_{j=1}^m \Phi_j = \Omega \quad \text{and} \quad \Phi_j \cap \Phi_k = \emptyset \quad \forall j \neq k \quad (1)$$

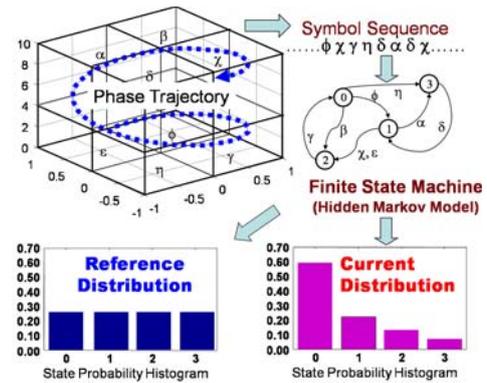


Fig. 2 Conceptual view of symbolic time series analysis

Each block  $\Phi_j$  is labelled as the symbol  $\sigma_j \in \mathcal{A}$ , where the symbol set  $\mathcal{A}$  is called the *alphabet*, consisting of  $m$  different symbols (i.e.  $m = |\mathcal{A}|$ ). As the dynamical system evolves in time, it travels through or touches various blocks in its phase space and the corresponding symbol is assigned to it, thus converting a data sequence to a symbol sequence  $\dots \sigma_{i_1} \sigma_{i_2} \dots \sigma_{i_k} \dots$ . As such, if a data point falls in a particular partitioning region it is assigned the corresponding symbol of that region and the data sequence is transformed into a symbol sequence in this manner.

Figure 2 also shows the construction of an HMM [7] from the symbol sequence as a finite state machine. The quasi-stationary histograms of the state probability distribution represent patterns that are indicative of the nominal (or reference) and anomalous behaviour of the dynamical system, as explained later.

Several partitioning techniques have been reported in the literature for symbol generation [19, 20]. The method adopted in this paper is maximum entropy partitioning [12, 18], where the time series data are partitioned such that the regions with more information are partitioned finer and those with sparse information are partitioned coarser. A uniform probability distribution of the states is a consequence of the maximum entropy partitioning. Once the symbol sequence is generated, the next step is to construct a finite state machine as described in the following subsection.

### 2.2 D-Markov machine construction

This subsection describes the construction of an HMM from a symbol sequence. A finite state machine is constructed, where the states of the machine are defined corresponding to a given

alphabet  $\mathcal{A}$  and window length  $D$ . The alphabet size  $|\mathcal{A}|$  is the total number of partitions whereas the window length  $D$  is the length of consecutive symbol words forming the states of the machine [1]. The states of the machine are chosen as words of length  $D$  from the symbol sequence, thereby making the number  $n$  of states to be less than or equal to the total permutations of the symbols within words of length  $D$ , i.e.  $n \leq |\mathcal{A}|^D$ . The choice of  $|\mathcal{A}|$  and  $D$  depends on specific experiments, noise level, partitioning, and also the available computation power. A large alphabet may be noise sensitive whereas a small alphabet could miss the details of signal dynamics. Similarly, a high value of  $D$  is sensitive to small signal distortions but would lead to a larger number of states requiring more computation power and longer data sets. For machine construction, the window of length  $D$  on the symbol sequence  $\dots \sigma_{i_1} \sigma_{i_2} \dots \sigma_{i_k} \dots$  is shifted to the right by one symbol, such that it retains the last  $(D-1)$  symbols of the previous state and appends it with the new symbol  $\sigma_{i_k}$  at the end. The symbolic permutation in the current window gives rise to a new state. The constructed machine is called  $D$ -Markov because of its Markov properties [1].

**Definition 3.** A symbolic stationary process is called  $D$ -Markov if the probability of the next symbol depends only on the previous  $D$  symbols, that is

$$P(\sigma_{i_0} | \sigma_{i-1} \dots \sigma_{i-D} \sigma_{i-D-1} \dots) = P(\sigma_{i_0} | \sigma_{i-1} \dots \sigma_{i-D})$$

The finite state machine constructed above has  $D$ -Markov properties because the probability of occurrence of symbol  $\sigma_{i_k}$  on a particular state depends only on the configuration of that state, i.e. the previous  $D$  symbols. For example, if  $\mathcal{A} = \{0, 1\}$ , i.e.  $|\mathcal{A}| = 2$  and  $D = 2$ , then the number of states is  $n \leq |\mathcal{A}|^D = 4$ ; and the possible states are 00, 01, 10, and 11, some of which may be forbidden.

Once the alphabet  $\mathcal{A}$  and word length  $D$  are assigned at the nominal condition at time epoch  $t_0$ , they are kept invariant for subsequent (slow time) epochs  $\{t_1, t_2, \dots, t_k, \dots\}$ , i.e. the machine structure is fixed although arc probabilities may vary. The states of the machine are marked with the corresponding symbolic word permutation and edges joining the states indicate occurrence of an event  $\sigma_{i_k}$ .

**Definition 4.** The probability of transitions from state  $q_j$  to state  $q_k$  belonging to the set  $Q$  of states under a transition  $\delta : Q \times \mathcal{A} \rightarrow Q$  is defined as

$$\pi_{jk} = P(\sigma \in \mathcal{A} | \delta(q_j, \sigma) \rightarrow q_k), \quad \sum_k \pi_{jk} = 1 \quad (2)$$

For a  $D$ -Markov machine under quasi-stationary conditions, the irreducible\* stochastic matrix  $\mathbf{\Pi} \equiv [\pi_{ij}]$  describes all transition probabilities between states such that it has at most  $|\mathcal{A}|^{D+1}$  non-zero entries. The left eigenvector  $\mathbf{p}$  corresponding to the unique unity eigenvalue of  $\mathbf{\Pi}$  is the state probability vector under the (fast time scale) stationary condition of the dynamical system [1]. For computation of state transition probabilities from a given symbol sequence  $\dots \sigma_{i_1} \sigma_{i_2} \dots \sigma_{i_k} \dots$  generated from the time series data collected at the slow time epoch  $t_k$ , a  $D$ -block (window of length  $D$ ) is moved by counting occurrences of symbol blocks  $\sigma_{i_1} \dots \sigma_{i_D} \sigma_{i_{D+1}}$  and  $\sigma_{i_1} \dots \sigma_{i_D}$  which are respectively denoted by  $N(\sigma_{i_1} \dots \sigma_{i_D} \sigma_{i_{D+1}})$  and  $N(\sigma_{i_1} \dots \sigma_{i_D})$ . Note that if  $N(\sigma_{i_1} \dots \sigma_{i_D}) = 0$ , then the state  $q \equiv \sigma_{i_1} \dots \sigma_{i_D} \in Q$  has a zero probability of occurrence. For  $N(\sigma_{i_1} \dots \sigma_{i_D}) \neq 0$ , the transition probabilities are then obtained by these frequency counts as follows

$$\begin{aligned} \pi_{jk} &\stackrel{\Delta}{=} P[q_k | q_j] = \frac{P(\sigma_{i_1} \dots \sigma_{i_D} \sigma_{i_{D+1}})}{P(\sigma_{i_1} \dots \sigma_{i_D})} \\ &\approx \frac{N(\sigma_{i_1} \dots \sigma_{i_D} \sigma_{i_{D+1}})}{N(\sigma_{i_1} \dots \sigma_{i_D})} \end{aligned} \quad (3)$$

where the corresponding states are denoted by  $q_j \stackrel{\Delta}{=} \sigma_{i_1} \sigma_{i_2} \dots \sigma_{i_D}$  and  $q_k \stackrel{\Delta}{=} \sigma_{i_2} \dots \sigma_{i_D} \sigma_{i_{D+1}}$ .

The time series data under the quasi-stationary condition at a (slow time) epoch  $t_0$  are set as the benchmark to generate the (irreducible) *state transition matrix*  $\mathbf{\Pi}_0$  that, in turn, is used to obtain the *state probability vector*  $\mathbf{p}^0$ . Note that  $\mathbf{p}^0$  is a uniform distribution because of maximum entropy partition [12]. In general, the elements of the vector  $\mathbf{p}^k$  are quasi-stationary state probabilities of the  $D$ -Markov machine at a slow time epoch  $t_k$ . Referring to Fig. 2,  $\mathbf{p}^0$  serves as the reference statistical pattern and  $\mathbf{p}^k$  is the statistical pattern of the slowly evolving dynamical system at the time epoch  $t_k$  by using the same partition. The pattern changes are quantified as deviations from the reference pattern (i.e. the probability distribution at the reference condition) [22]. The resulting deviations are characterized by a scalar-valued function, called *the deviation measure*  $m$ . The deviation measures are obtained as

\*A square non-negative real matrix  $\mathbf{A}$  is called irreducible [21] if, for every  $i$  and  $j$ , there exists a positive-integer  $k$  such that the  $ij$ th element of the  $k$ th power of  $\mathbf{A}$  is strictly positive. The integer  $k$  may vary with  $i$  and  $j$ .

$$m = d(\mathbf{p}, \mathbf{p}^0) \quad (4)$$

where  $d(\bullet, \bullet)$  is an appropriately defined distance function [1].

### 2.3 Stopping rule for symbol sequence generation

This subsection presents a stopping rule that is necessary to find a lower bound on the length of the symbol sequence required for convergence of the quasi-stationary stochastic matrix  $\mathbf{\Pi}$ . The stopping rule [14] is based on the properties of irreducible stochastic matrices [21]. The state transition matrix is constructed at the  $r$ th iteration (i.e. from a symbol sequence of length  $r$ ) as  $\mathbf{\Pi}(r)$  that is an  $n \times n$  irreducible stochastic matrix under stationary conditions. Similarly, the state probability vector  $\mathbf{p}(r) \equiv [p_1(r)p_2(r)\dots p_n(r)]$  is obtained as

$$p_i(r) = \frac{r_i}{\sum_{j=1}^n r_j} \quad (5)$$

where  $r_i$  is the number of D-blocks representing the  $i$ th state such that  $\sum_{j=1}^n r_j + D - 1 = r$  is the total length of the data sequence under symbolization. The stopping rule makes use of the Perron–Frobenius theorem [21] to establish a relation between the vector  $\mathbf{p}(r)$  and the matrix  $\mathbf{\Pi}(r)$ . Since the matrix  $\mathbf{\Pi}(r)$  is stochastic and irreducible, there exists a unique eigenvalue  $\lambda = 1$  and the corresponding left eigenvector  $\mathbf{p}(r)$  (normalized to unity in the sense of absolute sum). The left eigenvector  $\mathbf{p}(r)$  represents the state probability vector, provided that the matrix parameters have converged after a sufficiently large number of iterations. That is

$$\mathbf{p}(r+1) = \mathbf{p}(r)\mathbf{\Pi}(r) \Rightarrow \mathbf{p}(r) = \mathbf{p}(r)\mathbf{\Pi}(r) \quad \text{as } r \rightarrow \infty \quad (6)$$

Following equation (5), the absolute error between successive iterations is obtained such that

$$\|(\mathbf{p}(r) - \mathbf{p}(r+1))\|_{\infty} = \|\mathbf{p}(r)(\mathbf{I} - \mathbf{\Pi}(r))\|_{\infty} \leq \frac{1}{r} \quad (7)$$

where  $\|\bullet\|_{\infty}$  is the max norm of the finite-dimensional vector  $\bullet$ .

To calculate the stopping point  $r_{\text{stop}}$ , a tolerance of  $\eta$  ( $0 < \eta \ll 1$ ) is specified for the relative error such that

$$\frac{\|(\mathbf{p}(r) - \mathbf{p}(r+1))\|_{\infty}}{\|\mathbf{p}(r)\|_{\infty}} \leq \eta \quad \forall r \geq r_{\text{stop}} \quad (8)$$

The objective is to obtain the least conservative estimate for  $r_{\text{stop}}$  such that the dominant elements of the probability vector have smaller relative errors than the remaining elements. Since the minimum possible value of  $\|\mathbf{p}(r)\|_{\infty}$  for all  $r$  is  $1/n$ , where  $n$  is the dimension of  $\mathbf{p}(r)$ , the stopping point is obtained from equations (7) and (8) as

$$r_{\text{stop}} \equiv \text{floor}\left(\frac{n}{\eta}\right) \quad (9)$$

where  $\text{floor}(\bullet)$  is the integer part of the real number  $\bullet$ .

### 2.4 Summary of SDF for pattern identification

The major steps in application of SDF for pattern recognition can be summarized as follows.

1. Acquisition of time series data from appropriate sensor(s) and/or analytically derived model variables at the reference condition.
2. Signal conditioning of time series data for elimination of bias and background noise as necessary.
3. Maximum entropy partitioning of data sets at the reference condition and generation of the corresponding symbol sequence.
4. Construction of a  $D$ -Markov machine and computation of the state probability vector  $\mathbf{p}^0$  at the reference condition.
5. Acquisition of time series data at other slow time epochs  $t_k$ .
6. Generation of symbol sequences based on partitioning constructed at the reference condition and computing the state probability vector  $\mathbf{p}^k$  based on time series data at the slow time epoch  $t_k$ .
7. Computation of deviation measures  $m_k$  at each slow time epoch  $t_k$ .
8. Identification of the unknown pattern from the deviation measure at each slow time epoch  $t_k$ .

The major advantages of SDF for pattern identification are listed below.

1. Robustness to measurement noise and spurious signals [12].
2. Adaptability to low-resolution sensing due to the coarse graining in space partitions [1].
3. Capability to detect small deviations because of sensitivity to signal distortion and real-time execution on commercially available inexpensive platforms.

Chin *et al.* [23], Rao *et al.* [24], and Gupta *et al.* [25] have reported the superior capability of SDF for

detection of small anomalies with respect to other statistical pattern recognition techniques such as artificial neural networks and principal component analysis.

### 3 PATTERN IDENTIFICATION OF MOBILE ROBOTS

In contrast to the authors' earlier work on SDF-based detection of slowly evolving anomalies [1, 4], the objective of this paper is to identify the most likely pattern among a finite set of predetermined patterns from the time series data. Since behavioural 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 section 2.2. Therefore, instead of fixing the partition for the set of time series data under the nominal condition and comparing every pattern with this nominal pattern, a family of (maximum entropy) partitions [12] is generated corresponding to the patterns that need to be identified, i.e. a unique partition is obtained for each pattern from the respective time series data. The salient steps are delineated below.

1. Identification of a set of pattern classes spanning the robot behaviour under different operational conditions.
2. Generation of sets of time series data for all pattern classes either from laboratory or on site experiments, or from computer simulation. For computer simulation, an adequate dynamic model must include integration of the following two subsystem models: first a model of the robot kinematics and dynamics that may require access to manufacturer's database of the robot's internal components, and second a model of distributed dynamics of the flexible pressure-sensitive floor that has its own spatial-temporal memory. In addition, such a simulation model must be periodically calibrated with freshly conducted experimental data to ensure its credibility. The experimental procedure has been adopted in this paper in the absence of a reliable simulation model.
3. Analysis of the data sets to generate a unique partition of time series data sets belonging to each pattern class.

Let  $\xi_i(x)$  represent the  $i$ th pattern (e.g. the type and motion of a given robot) based on symbolization of the time series data set  $x$ . The set  $\Xi$  is defined as a

(non-empty finite) collection of patterns  $\xi_i$ ,  $i = 1, 2, \dots, |\Xi|$ , where  $|\Xi|$  is the cardinality of  $\Xi$ . A maximum entropy partitioning is generated from the reference time series data of each pattern. A pattern  $\xi_i$ ,  $i = 1, 2, \dots, |\Xi|$  has the reference probability distribution  $\mathbf{p}^{i0}$  that is uniform as a consequence of maximum entropy partitioning.

The analysis and supporting experimentation in this paper are formulated by constructing an identity map between the set  $\{\xi_1, \xi_2, \dots, \xi_{|\Xi|}\}$  of pattern classes and the set  $\{d_1, d_2, \dots, d_{|\Xi|}\}$  of decisions on partition selection for symbolization of time series data. The objective here is to formulate a non-negative real measure of these partitioning decisions for a given time series data.

Let  $x$  be a set of time series data that truly belongs to the pattern class  $\xi_j$ ,  $j \in \{1, \dots, |\Xi|\}$ . Let the partitioning decision of the data set  $x$  be  $d_i$ ,  $i \in \{1, \dots, |\Xi|\}$ , and the respective state probability distributions  $\mathbf{p}^{ij}(x)$ ,  $i = 1, \dots, |\Xi|$ ,  $j = 1, \dots, |\Xi|$ , of the  $D$ -Markov machines are derived from the same time series data  $x$  (whose true pattern is  $\xi_j$ ). The following definition formalizes the notion of deviation measure of a partitioning decision.

**Definition 5.** Given a time series data set  $x$  whose true pattern is  $\xi_j$ , deviation measure of the partitioning decision  $d_i$  is defined in terms of the respective state probability distribution  $\mathbf{p}^{ij}(x)$  and the reference probability distribution  $\mathbf{p}^{i0}$  as

$$m_{ij}(x) \triangleq d(\mathbf{p}^{ij}(x), \mathbf{p}^{i0}) \quad (10)$$

where  $d(\bullet, \bullet)$  is an appropriate distance function (e.g. the standard Euclidean norm of the difference between the distributions,  $\mathbf{p}^{ij}(x)$  and  $\mathbf{p}^{i0}$ ).

Due to uncertainties prevalent in the time series data  $x$ , the deviation measure  $m_{ij}(x)$  in equation (10) would not be identically equal to zero, regardless of whether or not the decision  $d_i$  is correct. Nevertheless, it is expected that  $m_{ij}(x)$  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  $\mathcal{M}_{ij}(x)$  denote the random variable associated with the deviation measure when the decision  $d_i$  that the data set  $x$  belongs to the pattern class  $\xi_i$  while  $x$  truly belongs to the  $j$ th pattern class  $\xi_j$ . Then, realization of the random variable  $\mathcal{M}_{ij}$  is the non-negative real  $m_{ij}(x)$  in equation (10). Hence, for each pattern class  $\xi_j$ , there could be decisions  $d_i$ ,  $i = 1, 2, \dots, |\Xi|$  that give rise to realizations of the random variables  $\mathcal{M}_{ij}$ ,  $i = 1, 2, \dots, |\Xi|$  as  $m_{ij}(x)$ ,  $i = 1, 2, \dots, |\Xi|$ ; and there would be a total of  $|\Xi| \times |\Xi| = |\Xi|^2$

random variables  $\mathcal{M}_{ij}$ . In the following, the probability distribution of  $\mathcal{M}_{ij}$  is denoted as  $p_{\mathcal{M}_{ij}}$ .

The *a priori* conditional probability  $P[\mathbf{x}|\xi_j, d_i]$  represents the probability of observation of the data  $\mathbf{x}$  conditioned on the true pattern  $\xi_j$  and the decision  $d_i$  of making the partition that represents the pattern  $\xi_j$ . That is

$$P[\mathbf{x}|\xi_j, d_i] = p_{\mathcal{M}_{ij}}(m_{ij}(\mathbf{x})) \quad (11)$$

where  $\mathcal{M}_{ij}$  is the random variable representing a decision  $d_i$  when the true pattern class is  $\xi_j$ ; and the argument of the distribution  $p_{\mathcal{M}_{ij}}$  is  $m_{ij}$  that is the deviation measure (see equation (10)) of the state probability vector obtained by the partition decision  $d_i$  on the data set  $\mathbf{x}$  that truly belongs to the pattern class  $\xi_j$ . The *a posteriori* probabilities are given as

$$P[\xi_j|\mathbf{x}, d_i] = \frac{P[\mathbf{x}|\xi_j, d_i]P[\xi_j]}{P[\mathbf{x}|d_i]} \quad (12)$$

Equation (12) is expressed in a different form as

$$P[\xi_j|\mathbf{x}, d_i] = \frac{P[\mathbf{x}|\xi_j, d_i]P[\xi_j]}{\sum_k P[\mathbf{x}|\xi_k, d_i]P[\xi_k]} \quad (13)$$

based on the following two facts.

1. The pattern classes  $\xi_j$  form a mutually exclusive and exhaustive set. It follows from the total probability theorem that

$$P[\mathbf{x}|d_i] = \sum_k P[\mathbf{x}|\xi_k, d_i]P[\xi_k|d_i]$$

2. The prior probability of a pattern  $\xi_j$  is independent of the process of making the decision  $d_i$ , that is

$$P[\xi_j|d_i] = P[\xi_j]$$

Substitution of equation (11) in to equation (13) yields

$$P[\xi_j|\mathbf{x}, d_i] = \frac{p_{\mathcal{M}_{ij}}(m_{ij}(\mathbf{x}))P[\xi_j]}{\sum_k p_{\mathcal{M}_{ik}}(m_{ik}(\mathbf{x}))P[\xi_k]} \quad (14)$$

Let the risk of making a decision  $d_i$  when truly the pattern class is  $\xi_j$  be specified as  $\lambda_{ij}$ . Then, the total risk of making a decision  $d_i$  becomes [4]

$$R(d_i|\mathbf{x}) = \sum_{j=1}^{|\Xi|} \lambda_{ij}P(\xi_j|\mathbf{x}, d_i) \quad (15)$$

and the decision on pattern identification is made by minimizing the risk in equation (15) as

$$d^* = \underset{i}{\operatorname{argmin}} R(d_i|\mathbf{x}) \quad (16)$$

### 3.1 Behaviour identification

The long-term objective of the work reported in this paper is to autonomously identify the behaviour patterns (e.g. type, payload, and the kind of motion) of mobile robots, in which respective partitions are generated for *a priori* determined classes of behavioural patterns.

In the following, it is shown how the pattern vectors identify the robot (e.g. Segway RMP or Pioneer 2AT) and its type of motion (e.g. random motion, circular motion, or square motion). Future work is recommended in section 6 for online identification of robot behaviour under different payloads and environmental conditions.

Let  $\mathfrak{R} = \{\rho_1, \dots, \rho_{|\mathfrak{R}|}\}$  be the set of robots and let each robot execute one or more of the different motion profiles in the set  $\Phi = \{\phi_1, \dots, \phi_{|\Phi|}\}$ . Let the number of profiles executed by robot  $\rho_i$  be  $n_i$ . Also, let the indices of the profiles executed by robot  $\rho_i$  be  $\{y_1^i, \dots, y_{n_i}^i\}$ . That is robot  $\rho_i$  executes profiles  $\{\phi_{y_1^i}, \dots, \phi_{y_{n_i}^i}\}$ . Thus, the total number of pattern classes  $|\Xi| = \sum_{ij} n_i \leq |\mathfrak{R}||\Phi|$ . The pattern identification procedure first generates a partitioning  $\xi_j \in \Xi$  of time series data belonging to each pattern class  $j$ . Algorithm 1 describes the procedure to compute the pattern vectors. Once the partition set  $\Xi$  is constructed, sets of time series data are analysed to estimate the probability densities  $p_{\mathcal{M}_{ij}}(\bullet)$  by following the procedure in lines 10 to 18 of Algorithm 1.

Given a set of time series data with an unidentified pattern,  $|\Xi|$  symbol sequences are generated corresponding to the partitions  $\xi_i \in \Xi$  by following the procedure in section 2.1. Then, a  $D$ -Markov machine of appropriate depth  $D$  is constructed based on the procedure described in section 2.2. If the correct decision is made (i.e. the correct partition is applied to the data set), then the generated probability vector  $\mathbf{p}$  should be very close to the uniform distribution, implying that the deviation measure  $m_{ij} \approx 0.0$  in equation (10). The *a priori* probabilities  $p_{\mathcal{M}_{ij}}(m_{ij})$  are computed from the densities estimated in Algorithm 1. The *a posteriori* probabilities and the Bayes risk functions are then computed from  $p_{\mathcal{M}_{ij}}(m_{ij})$  via equations (14) and (15) respectively, as shown in lines 10 and 11 of Algorithm 3. The decision  $d^*$  is chosen so as to minimize the risk in

line 12 of the Algorithm 3. Lines 13 and 14 simply convert the identified pattern vector index  $d^*$  into corresponding indices of the robot  $i$  and the movement profile  $y_\ell^i$ .

## 4 EXPERIMENTAL APPARATUS

The experimental apparatus consists of a wireless networked system incorporating mobile robots, robot simulators, and distributed sensors.

### 4.1 Distributed sensor network

A major ingredient of the experimental apparatus is the pressure sensitive floor that consists of distributed piezoelectric wires serving as arrays of distributed pressure sensors. A coil of piezoelectric wire is placed under a  $0.65 \times 0.65$  m square floor tile as shown in Fig. 3(a) such that the sensor generates an analogue voltage due to pressure applied on it.

This voltage is sensed by a Brainstem™ microcontroller using one of its 10-bit A/D channels thereby yielding sensor readings in the range of zero to 1023. A total of 144 sensors are placed in a  $9 \times 16$  grid to cover the entire laboratory environment as shown in Fig. 3(b). The sensors are grouped into four quadrants, each being connected to a stack consisting of eight networked Brainstem microcontrollers for data acquisition. The microcontrollers are, in turn, connected to two laptop computers running a Player [26] server that collects the raw sensor data and distributes to any client over the wireless network for further processing.

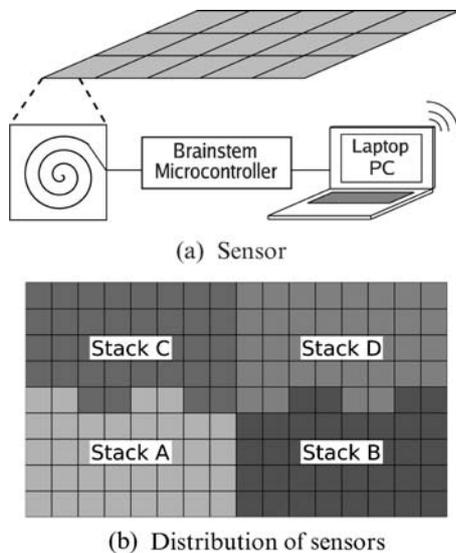


Fig. 3 Sensor layout in the laboratory environment

The functional blocks of data collection software are represented on the right-hand side of Fig. 4. Software for data acquisition from the distributed sensor network involves a specially written Player driver for collecting raw sensor data from the A/D channels of the microcontrollers. This software is run on two dedicated Player servers running on wireless laptops. Raw sensor data are then collected by a Player client software for further processing such as SDF. Using the SDF procedure, a  $D$ -Markov machine of appropriate depth  $D$  is constructed and the steady-state state probabilities,  $\mathbf{p}$ , are calculated to identify the closest pattern class.

---

#### Algorithm 1: Pattern Identification Forward Algorithm

---

```

input: time-series Data Sets
output: Sequence of Partition Vectors as matrix  $\Xi$  and
          probability densities  $p_{M_{ij}}(\bullet) \forall i, j = 1, \dots, |\Xi|$ 
1 begin
2   Let  $j = 0$ 
3   for  $i = 1$  to  $|\mathcal{R}|$  do
4     for  $k = 1$  to  $n_i$  do
5       Let robot  $\rho_i \in \mathcal{R}$  execute motion  $\varphi_{y_k} \in \Phi$ 
6       Collect the time series data  $\mathbf{x}_j$ 
7       Partition  $\mathbf{x}_j$  using Algorithm 2 to obtain pattern
          vector  $\xi_j$ 
8       Add  $\xi_j$  to the sequence  $\Xi$ 
9   increment  $j$  by one
10  for  $j = 1$  to  $|\Xi|$  do
11    for  $\ell = 1$  to  $L$  do
12      Collect time series data  $\mathbf{x}_j^\ell$ 
13      for  $i = 1$  to  $|\Xi|$  do
14        partition  $\mathbf{x}_j^\ell$  using  $\xi_i$  to obtain symbol sequence  $\mathbf{s}$ 
15        construct  $D$ -Markov machine  $G$  using  $\mathbf{s}$ 
16        compute state probability vector  $\mathbf{p}^{ij}$  for  $G$ 
17        compute the deviation measure  $m_{ij}^\ell = d(\mathbf{p}^{ij}, \mathbf{p}^{i0})$ 
18  From realizations  $\{m_{ij}^1, \dots, m_{ij}^L\}$  estimate the probability
          density for  $p_{M_{ij}}(\bullet) \forall i, j = 1, \dots, |\Xi|$ 
19 end

```

---



---

#### Algorithm 2: Maximum Entropy Partitioning

---

```

input: time-series Data  $\mathbf{x}$ , Number of Symbols  $|\mathcal{A}|$ 
output: Partition Vector  $\xi$ 
1 begin
2   sort  $\mathbf{x}$  in ascending order
3   let  $K = \text{length}(\mathbf{x})$ 
4    $\xi(1) = \mathbf{x}(1)$ ; minimum of  $\mathbf{x}$ 
5   for  $i = 2$  to  $|\mathcal{A}|$  do
6      $\xi(i) = \mathbf{x} \left( \text{floor} \left( \frac{(i-1) \times K}{|\mathcal{A}|} \right) \right)$ 
7    $\xi(|\mathcal{A}| + 1) = \mathbf{x}(K)$ ; maximum of  $\mathbf{x}$ 
8 end

```

---

### 4.2 Robot platforms

The robot hardware, consists of seven mobile robots, of which three are Pioneer 2AT robots and four are Segway RMP robots.

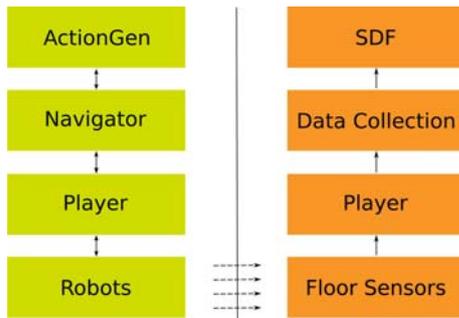


Fig. 4 Overview of the software architecture

Figure 5 shows a pair of Pioneer robots and a Segway RMP that have the following features.

1. The Segway RMP is a two-wheeled robot (with inverted pendulum dynamics) that has a zero turn radius and has an approximate weight of 70 kg.
2. The Pioneer 2AT is a four-wheeled robot that is equipped with a differential drive train system and has an approximate weight of 35 kg.

Since the Pioneer is lighter than the Segway and the Pioneer's load on the floor is more evenly distributed, their statistics are dissimilar. Furthermore, since kinematics and dynamics of the two types of robot are different, the texture of the respective pressure sensor signals is also different.

Each robot is equipped with a SICK LMS200 laser range finder for obstacle avoidance and distance measurement. The laser range finder provides depth information for a  $180^\circ$  field of view with an angular resolution of  $0.5^\circ$  and an accuracy of 1 cm ( $\pm 15$  per cent). In addition, the robots are equipped with



Fig. 5 The robot hardware: Pioneer 2AT and Segway RMP

multiple sonar sensors and infrared units serving as secondary range finders. SONY EVI-D30 pan-tilt cameras in conjunction with Sensoray 311 PC104 frame grabbers are used for object recognition and tracking. For the purpose of operation monitoring, control, and communication between the mobile robots and remote computers, a secure wireless subnet of the Penn State backbone connection is employed. In practice, a bandwidth of up to 2 Mbps is achieved. For the Pioneer AT robots, an Advantech on-board computer, powered by a Transmeta Crusoe Processor TM5400 500 MHz CPU, performs all real-time computations. It has a 320 MB memory including a 256 MB PC133 RAM and a 64 MB flash memory, and a 20 GB hard disk. The Segway RMPs are equipped with commercial laptops (Pentium M processor with a 2 GB RAM, 40 GB HDD). Each device is powered by multiple commercially available 12 V lead acid batteries; DC/DC converters are used to provide appropriate power to various devices. The actuators available on the robots are drive train motors, gripper, and camera.

The robots are operated with the functional blocks on the left-hand side of Fig. 4 while the data collection from the distributed sensors is run using the functional blocks on the right-hand side. The individual functional blocks can run on multiple machines networked either through Ethernet or WiFi. The communication between various functional blocks is through message passing via the TCP/IP protocol. The Player [26] is the lowest level functional block that has access to the robots' physical hardware, namely the sensors (e.g. laser, sonar, and camera) and actuators. The Player is usually run on the onboard computer of the robot to reduce the network-induced time delay in sending commands to the robot motion controller. The ActionGenerator block in Fig. 4 is a client to the Player server and requests access to the sensors/actuators on the robot. The ActionGenerator consists of a set of fundamental behaviours that the robot can perform. For example, the Search behaviour looks for a target, and the GoTo behaviour makes moves from point to point in a predefined map that, in the

---

**Algorithm 3:** Pattern Identification Inverse Algorithm

---

**input:** time-series Data  $\mathbf{x}$ , Partition Matrix  $\Xi$   
**output:** Identified Pattern  $\xi \in \Xi$

```

1  begin
2  for  $i = 1$  to  $|\Xi|$  do
3    partition  $\mathbf{x}$  using  $\xi_i$  to get symbol sequence  $\mathbf{s}$ 
4    construct  $D$ -Markov machine  $G$  using  $\mathbf{s}$ 
5    compute state probability vector  $\mathbf{p}^{ij*}$  for  $G$ , where  $j^*$ 
   corresponds to the unknown pattern  $\xi_{j^*}$  that is yet to be
   identified
6    compute the deviation measure  $m_{ij} = d(\mathbf{p}, \mathbf{p}^0)$ 
7    for  $j = 1$  to  $|\Xi|$  do
8      compute  $P[\mathbf{x}|d_i, \xi_j] = p_{M_{ij}}(m_{ij}(\mathbf{x}))$  (equation (11))
9    for  $j = 1$  to  $|\Xi|$  do
10     compute  $P[\xi_j|\mathbf{x}, d_i]$  using equation (14)
11    compute the Bayes risk  $R(\mathbf{x}|d_i)$  using equation (15)
12    compute  $d^* = \operatorname{argmin}_i R(\mathbf{x}|d_i)$ 
13    from sequence  $\{n_1, n_2, \dots, n_{|\mathbb{N}|}\}$  compute the cumulative
   sequence  $\{0, n_1, n_1+n_2, \dots, \sum_{i=1}^{|\mathbb{N}|} n_i\}$  to form the new
   sequence  $\{N_0, N_1, N_2, \dots, N_{|\mathbb{N}|}\}$ 
14    find  $i$  such that  $N_{i-1} < d^* \leq N_i$  and  $k = N_i - j$ 
15    Conclude that the robot  $\rho_i$  was executing  $\varphi_{y_k}$  profile
16  end
```

---

context of this paper, include motion behaviours (e.g. random motion, circular motion, and square motion).

The Navigator block implements a discrete event obstacle avoidance algorithm [27] to maintain safe operation of the robot and is capable of receiving commands from the ActionGenerator to move. However, the Navigator block overrides the command if it cannot safely navigate using the command given by ActionGenerator.

## 5 EXPERIMENTAL RESULTS AND DISCUSSION

This section provides a detailed description of the experimental procedure, an application of the SDF method to time series data of robot signature, and discussion of the experimental results. The objective here is to identify the statistical patterns of robot behaviour that may include both parametric and non-parametric uncertainties including the following.

1. Small variations in the robot mass that includes unloaded base weights of the platform itself and its payload.
2. Uncertainties in the 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.

In the presence of the above uncertainties, a complete solution of the pattern identification problem may not be possible in a deterministic setting because the pattern measure would not be identical for similar robots behaving similarly. Therefore, the problem is posed in the statistical setting, where a family of pattern measures is generated from multiple experiments conducted under identical operating conditions. The requirement is to generate a family of patterns for each class of robot behaviour that needs to be recognized. Each member of a family represents the pattern measure of a single experiment of one robot executing a particular motion profile.

Both the Segway RMP and Pioneer 2AT robots are commanded to execute three different motion trajectories, namely, random motion, circular motion, and square motion, following the software architecture described in section 4.2. Since the

spatial distribution of the pressure-sensitive coils underneath the floor (see Fig. 3) is statistically homogeneous, the decisions on the detection of robot behaviour patterns are statistically independent of the robot location (e.g. centre of the circle, centre and orientation of the square, and mean of the distribution for random motion). From this perspective, the following parameters are selected for the above three types of motion:

- (a) *Diameter of the circular motion:* 4 m;
- (b) *Edge length of the square motion:* 3 m;
- (c) *Distribution of the random motion:* Uniform in the range of 1 to 7 m in the  $x$ -direction and 1 to 4 m in the  $y$ -direction.

For the two types of robot, a total of six different behaviour patterns are defined. The data sets for each of the six pattern classes are collected and processed to create the respective partitions  $\zeta_i \in \Xi$ , where  $|\Xi| = 6$ . To simplify the exposition of results, the patterns are labelled as follows:

- (a) Segway random = 1, Segway circle = 2, Segway square = 3;
- (b) Pioneer random = 4, Pioneer circle = 5, Pioneer square = 6.

Both robots were made to execute each of the three different types of trajectory on the pressure-sensitive floor of the laboratory environment for about an hour. The data from the 144 sensors in the  $(9 \times 16)$  grid floor were collected at the sampling rate of approximately 10 Hz. The 144 readings were stacked row-wise to create a one-dimensional array of length 144. This configuration facilitates identification of the type of robot and its motion profile but not its exact location. Thus, a total of approximately 1440 points per second were collected for an hour to create a sufficiently long one-dimensional time series data set. This procedure was repeated to collect six data sets for two different robots executing each of the three different motion behaviours. It has been assumed that, during the execution of each motion, the statistical behaviour of the robot is stationary and it does not switch behaviours in between.

The voltage generated by the piezoelectric pressure sensors, embedded under the floor, is converted to a digital signal by using a 10-bit A/D converter over the range of zero to 5 V to generate readings in the range of zero to 1023. 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.

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

1. *Partitioning method*: Hilbert-transform-based analytical signal space partitioning [18].
2. *D-Markov machine parameters*: Alphabet size  $|\mathcal{A}|=8$  and depth  $D=1$ .
3. *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 behavioural patterns with only eight 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 [12]. Increasing the value of  $D$  beyond one 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.

The results of partitioning the data sets using Algorithm 1 are presented in Fig. 6. The abscissa shows segment numbers of the six partitions and the ordinate represents the respective segment boundaries in terms of (pressure) sensor readings as outputs of 10-bit analog-to-digital converters. For a clear exposition of differences in the partitions, the two bottom plates in Fig. 6 present separate displays for the partition segments of one to four and five to eight, respectively, for different ranges of sensor readings. It is seen in Fig. 6 that the segment lengths are non-uniform due to maximum entropy partitioning.

### 5.1 Generation of statistical patterns

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

1. Partition the  $l$ th data set for pattern  $j$  using the reference partition  $i$  to generate a symbol sequence.
2. 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  $\mathbf{p}$ .
3. Compute the realization  $m_{ij}^\ell$ .

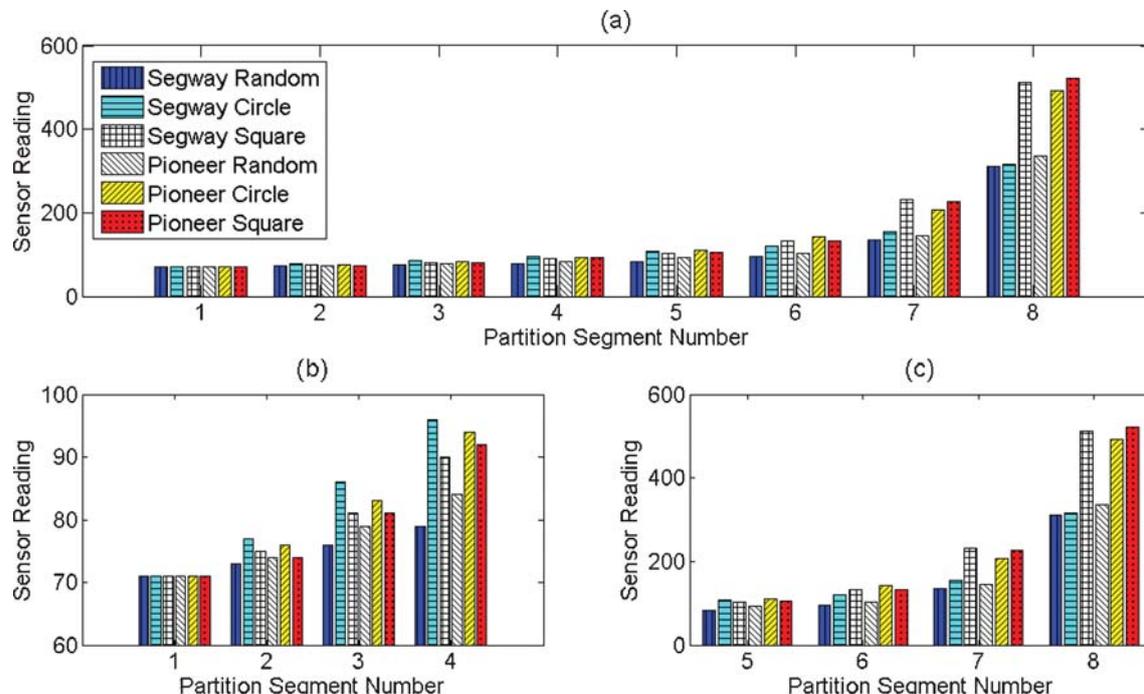


Fig. 6 Partitions created by various patterns

Thus, an ensemble consisting of  $L = 60$  realizations  $\{m_{ij}^1, \dots, m_{ij}^L\}$  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 a lognormal distribution of  $\mathcal{M}_{ij}$ , as opposed to other distributions (e.g. normal or Weibull), is stated below.

1. 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.
2. For a sample data set using the correct partition, 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 modelled by a lognormal distribution.
3. Since the random variable  $\ln(\mathcal{M}_{ij})$  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

$$p_{\mathcal{M}_{ij}}(x) = \frac{1}{\sqrt{2\pi}\sigma_{ij}x} \exp\left(-\frac{(\ln(x) - \mu_{ij})^2}{2\sigma_{ij}^2}\right) \mathcal{U}(x) \quad (17)$$

where  $\mathcal{U}(\bullet)$  is the standard Heaviside unit step function; and  $\mu$  and  $\sigma$  are respectively the mean and standard deviation of the Gaussian distributed random variable  $\ln(\mathcal{M}_{ij})$ . The two parameters (i.e. mean  $\mu_{ij}$  and the variance  $\sigma_{ij}^2$  of  $\mathcal{M}_{ij}$ ) for lognormal distribution were identified from each of the 36 sets,  $\{m_{ij}^1, \dots, m_{ij}^L\}$ . Each lognormal distribution satisfied the 10 per cent significance level which suffices for the conventional standard of 5 per cent significance level. Figure 7 shows the histograms of  $\mathcal{M}_{ij}$  for four typical cases (out of a total of 36) for different values of  $i$  and  $j$ . The goodness of fit of these histograms evinces that the lognormal distribution is an adequate approximation of the statistics of  $\mathcal{M}_{ij}$ .

### 5.2 Identification of robot type and motion profile

The problem at hand is to identify the type of robot and its motion profile on the pressure-sensitive floor in the laboratory environment. Based on the acquired information of statistical patterns, a

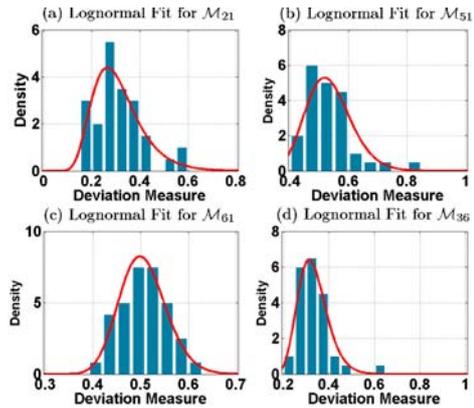


Fig. 7 PDF plots of  $\mathcal{M}_{ij}$  and their lognormal fit

solution to the above identification problem was obtained through usage of Algorithm 3 from online time series data from the unidentified robot (i.e. either Segway or Pioneer in the present experimentation). The time series data were partitioned using each of the six predefined partitioning formats shown in Fig. 6 to generate respective symbol sequences. A  $D$ -Markov machine (with state cardinality  $|\mathcal{A}|^D = 8$  for  $|\mathcal{A}| = 8$  and  $D = 1$ ) was constructed for each generated symbol sequence. The state probability vectors were computed for each of the constructed state machines; typical examples of state probability vectors for different motion profiles of Segway and Pioneer robots are shown in Fig. 8. Following equation (10), the deviation measure  $m_{ij}(x)$  was computed for each probability vector  $\mathbf{p}^{ij}(x)$ , where  $i \in \{1, \dots, 6\}$  for a given set of time series data  $x$  and the unknown pattern  $\xi_j$  is yet to be identified.

The pattern of robot type and motion was identified based on the probabilistic Bayesian method in section 3. The following assumptions were made in the absence of any specific information on  $P[\xi_j]$  in equation (14) and  $\lambda_{ij}$  in equation (15).

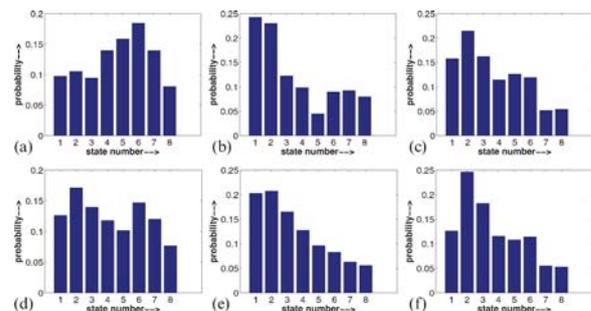


Fig. 8 State probability vectors for pattern identification (a) Segway random; (b) Segway circle; (c) Segway square; (d) Pioneer random; (e) Pioneer circle; (f) Pioneer square

**Table 1** Deviation measures  $m_{ij}(\mathbf{x})$  (see equation (10))

Decision $d_i$	$m_{ij}(\mathbf{x})$
Segway random	0.14831
Segway circular	0.12076
Segway square	0.10643
Pioneer random	0.44743
Pioneer circular	0.075726
Pioneer square	0.038984

1. Uniform probability of the prior probabilities of occurrence of the pattern classes  $\xi_j$ 's, that is

$$P[\xi_j] = \frac{1}{|\Xi|} \quad \forall j \in \{1, \dots, |\Xi|\}$$

2. Uniform non-zero 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  $\lambda_{ij}$  values and  $P[\xi_j]$  values, risk minimization in equation (16) is equivalent to having the maximum likelihood estimate of the pattern class as

$$\operatorname{argmax}_i (P[\mathbf{x}|d_i, \xi_i]) = \operatorname{argmax}_i (p_{\mathcal{M}_{ii}}(m_{ij})(\mathbf{x})) \quad (18)$$

The pertinent results for a given time series data  $\mathbf{x}$  are summarized in Tables 1, 2, 3, 4, and 5. Table 1 lists the values of the deviation measures  $m_{ij}$  for an unidentified set of time series data  $\mathbf{x}$  which belongs to the class of Pioneer Square (i.e.  $\xi_6$ ). The matrix elements in Table 2 list the values of  $p_{\mathcal{M}_{ij}}(m_{ij})$  for  $i = 1, 2, \dots, 6$  and  $j = 1, 2, \dots, 6$ . The values in Table 3 show the *a posteriori* probabilities given by equation (14). Assuming the prior probabilities for all  $\xi_j$  are equal, equation (14) reduces to

$$P[\xi_j|\mathbf{x}, d_i] = \frac{p_{\mathcal{M}_{ij}}(m_{ij}(\mathbf{x}))}{\sum_k p_{\mathcal{M}_{ik}}(m_{ij}(\mathbf{x}))} \quad (19)$$

Table 4 shows the risks  $\lambda_{ij} = (1 - \delta_{ij})$  of making decision  $d_i$  when the true hypothesis is  $\xi_j$ . With this choice of the risk parameters, the total risk of making the decision  $d_i$  given by equation (15) becomes

$$R(d_i|\mathbf{x}) = \sum_{j \neq i} P(\xi_j|\mathbf{x}, d_i) \quad (20)$$

Table 5 shows the computed values of the total risk of making decision  $d_i$ . The decision  $d^*$  as given in

**Table 2** Probabilities  $p_{\mathcal{M}_{ij}}(m_{ij}(\mathbf{x}))$  (see equation (17))

Decision $d_i \setminus$ True pattern $\xi_j$	Segway random	Segway circular	Segway square	Pioneer random	Pioneer circular	Pioneer square
Segway random	0.0030	0.0001	0.0120	0.0000	0.2816	0.1223
Segway circular	0.0883	0.0000	0.0186	0.0000	0.2422	0.1909
Segway square	0.1724	0.0000	0.0049	0.0000	0.2829	0.2013
Pioneer random	0.5007	0.0080	0.5416	0.0479	0.4663	0.6150
Pioneer circular	0.0053	0.0000	0.1495	0.0000	0.1652	0.3338
Pioneer square	0.0000	0.0000	0.0008	0.0000	0.0165	0.3966

**Table 3** *A Posteriori* Probabilities  $P[\xi_j|\mathbf{x}, d_i]$  (see equation (19))

Decision $d_i \setminus$ True pattern $\xi_j$	Segway random	Segway circular	Segway square	Pioneer random	Pioneer circular	Pioneer square
Segway random	0.0071	0.0003	0.0287	0.0000	0.6720	0.2919
Segway circular	0.1635	0.0000	0.0344	0.0000	0.4485	0.3535
Segway square	0.2607	0.0000	0.0074	0.0000	0.4277	0.3043
Pioneer random	0.2297	0.0037	0.2485	0.0220	0.2140	0.2822
Pioneer circular	0.0081	0.0000	0.2286	0.0000	0.2527	0.5105
Pioneer square	0.0000	0.0000	0.0019	0.0000	0.0400	0.9581

**Table 4** The risks  $\lambda_{ij}$  of making decision  $d_i$  when the true pattern is  $\xi_j$ 

Decision $d_i \setminus$ True pattern $\xi_j$	Segway random	Segway circular	Segway square	Pioneer random	Pioneer circular	Pioneer square
Segway random	0.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Segway circular	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000
Segway square	1.0000	1.0000	0.0000	1.0000	1.0000	1.0000
Pioneer random	1.0000	1.0000	1.0000	0.0000	1.0000	1.0000
Pioneer circular	1.0000	1.0000	1.0000	1.0000	0.0000	1.0000
Pioneer square	1.0000	1.0000	1.0000	1.0000	1.0000	0.0000

**Table 5** Total risk of making decision  $d_i$  (see equation (20))

Decision $d_i$	Total risk $R(d_i \mathbf{x})$
Segway random	0.9929
Segway circular	1.0000
Segway square	0.9926
Pioneer random	0.9780
Pioneer circular	0.7473
Pioneer square	<b>0.0419</b> (see equation (16))

equation (16), is the one that minimizes the total risk  $R(d_i|\mathbf{x})$ . The maximum likelihood estimate is simply the maximum of the diagonal elements of the matrix  $[p_{\mathcal{M}_{ij}}(m_{ij})]$  given in Table 2. In this example the maximum corresponds to Pioneer Square, shown in bold in Table 5, which confirms the decision  $d^*$  obtained by minimizing the total risk in equation (20).

Figure 9 shows the results of identification of pattern types for 60 sets of time series data for each pattern class. The numbers on the abscissa of Fig. 9 indicate the data corresponding to a particular pattern class. For all six patterns, the algorithms successfully identified the patterns for more than 80 per cent 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 section 5.1).

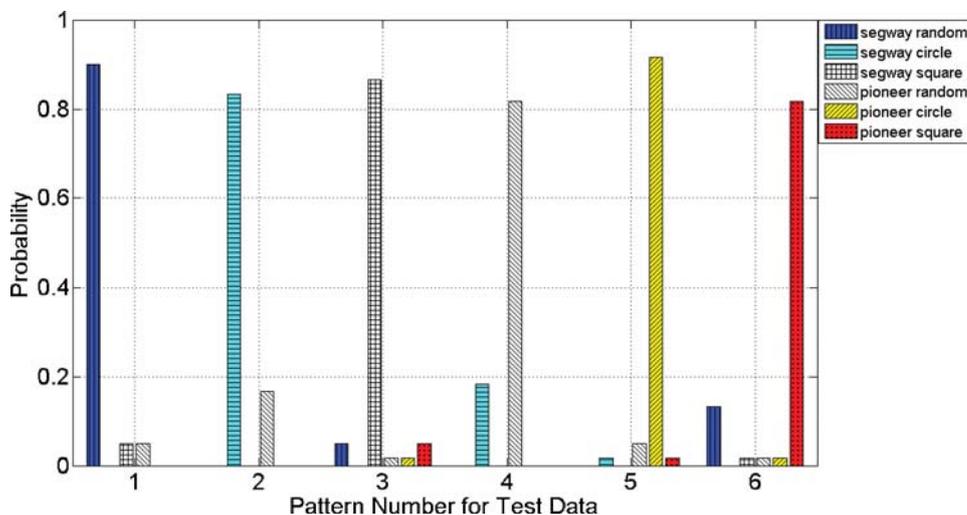
The performance for correct pattern identification was also tested both in the absence and presence of faults. For healthy robots, robustness of performance was tested by perturbing the parameters  $\lambda_{ij}$  values and  $P[\xi_j]$  values by 10 per cent from their nominal values; the algorithms successfully identified the patterns for about 80 per cent of the cases studied for all pattern classes.

Two cases were encountered for fault occurrences during the course of experiments. In the first case, a pin in the wheel of the Pioneer was broken which made it difficult for the robot to turn. This event significantly affected the robot's motion characteristics and the performance for correct pattern identification dropped below 35 per cent. In the second case of the Segway moving in a circle, a malfunction in the robot control software made the robot repeatedly run toward the walls of the laboratory and the underlying obstacle avoidance algorithm had to change the robot's motion behaviour to avoid collisions. This phenomenon caused the performance of pattern identification to drop below 50 per cent.

## 6 SUMMARY, CONCLUSIONS, AND FUTURE WORK

This paper presents an online dynamic data-driven method for identification of behaviour patterns in autonomous agents such as mobile robots. The proposed method utilizes SDF [1] to model the statistical behaviour patterns of mobile robots. These identified models are then used to detect the pattern class of robot behaviour (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. The proposed pattern identification method has the following distinct features compared to standard statistical pattern recognition [4].

1. Fully automated model identification in the symbol space via coarse graining of the phase space. This feature allows usage of relatively

**Fig. 9** Performance of the robot behaviour identification algorithm for different patterns

low-precision and inexpensive commercially available sensors.

2. Robustness to parametric and non-parametric uncertainties due to, for example, phase distortion and imprecise initial conditions.
3. Insensitivity to environmental and spurious disturbances due to the inherent noise suppression capability of SDF [1, 12].

The pattern identification method has been tested to closely identify the correct behaviour pattern among six different pattern classes with two different kinds of robot, each performing three types of movement in the laboratory environment. However, further theoretical and experimental research is needed before its implementation beyond the laboratory environment. Topics of future research include the following.

1. *Comparison of SDF with other existing pattern recognition tools for robot behaviour identification under different operating (e.g. change of payload) and environmental (e.g. change of terrain) conditions*: Performance evaluation of SDF has been reported for other applications such as fatigue damage detection [25] and non-linear active electronic systems [24]. Similar investigation is needed for robot behaviour identification.
2. *Behaviour identification of multiple robots simultaneously operating in a dynamic environment*: This scenario poses a challenging problem because the time series data are intermingled and signal separation becomes very difficult, especially in real time. A possible solution to circumvent this problem is spatial localization of data by noting the initial positions and keeping track of the robot motion throughout the operation, thus generating individual time series data for each robot.
3. *Augmentation of data-driven analysis with model-based information*: There is a strong spatial correlation among the time series data generated from neighbouring slabs in the pressure-sensitive floor due to inherent mechanical coupling that exists for each sensor from its neighbours and also due to electrical coupling of the sensors having a common ground at the microcontroller end. Due to the capacitive nature of piezoelectric transducers, the sensor readings in response to a step change in the load decay slowly, which creates temporal correlations as well. Analysis in the frequency domain may also be ineffective because the generated data sets suffer from a very wide frequency band with near-uniform power throughout the band. Therefore, information on the spatial-temporal correlations needs to be

captured for more accurate estimates of the probability histogram estimates.

## ACKNOWLEDGEMENTS

The authors thankfully acknowledge technical assistance of their colleague, Dr. Eric Keller, for design and construction of the laboratory facility.

This work has been supported in part by the Army Research Laboratory and the Army Research Office under grant W911NF-07-1-0376 and by the Office of Naval Research under grant N00014-08-1-380.

## REFERENCES

- 1 **Ray, A.** Symbolic dynamic analysis of complex systems for anomaly detection. *Signal Process.*, 2004, **84**(7), 1115–1130.
- 2 **Lind, D.** and **Marcus, M.** *An introduction to symbolic dynamics and coding*, 1995 (Cambridge University Press, Cambridge, UK).
- 3 **Hopcroft, H., Motwani, R., and Ullman, J.** *Introduction to automata theory, languages, and computation*, second edition, 2001 (Addison Wesley, Boston, MA).
- 4 **Duda, R., Hart, P., and Stork, D.** *Pattern classification*, 2001 (Wiley-Interscience, New York, NY, USA).
- 5 **Crutchfield, J. P.** and **Young, K.** Inferring statistical complexity. *Phys. Rev. Lett.*, 1989, **63**, 105–108.
- 6 **Shalizi, C., Shalizi, K., and Crutchfield, J.** An algorithm for pattern discovery in time series. Technical report, Santa Fe Institute, October 2002.
- 7 **Rabiner, L.** A tutorial on hidden Markov models and selected applications in speech processing. *Proc. IEEE*, 1989, **77**(2), 257–286.
- 8 **Cover, T. M.** and **Thomas, J. A.** *Elements of information theory*, 1991 (Wiley-Interscience, New York, NY, USA).
- 9 **Han, K.** and **Veloso, M.** Automated robot behavior recognition applied to robotic soccer. In Proceedings of IJCAI-99 Workshop on *Team behaviors and plan recognition*, 1999.
- 10 **Beck, C.** and **Schlogl, F.** *Thermodynamics of chaotic systems: an introduction*, 1993 (Cambridge University Press, Cambridge, UK).
- 11 **Badii, R.** and **Politi, A.** *Complexity hierarchical structures and scaling in physics*, 1997 (Cambridge University Press, Cambridge, UK).
- 12 **Rajagopalan, V.** and **Ray, A.** Symbolic time series analysis via wavelet-based partitioning. *Signal Process.*, 2006, **86**(11), 3309–3320.
- 13 **Brunk, H.** *An introduction to mathematical statistics*, third edition, 1995 (Xerox Publishing, Lexington, MA).
- 14 **Ray, A.** Signed real measure of regular languages for discrete-event supervisory control. *Int. J. Control*, 2005, **78**(12), 949–967.

- 15 Denis, N. and Jones, E. Spatio-temporal pattern detection using dynamic Bayesian networks. In Proceedings of the 42nd IEEE Conference on *Decision and control*, Maui, HI, USA, vol.5, December 2003, pp. 4533–4538.
- 16 Ruelle, D. *Thermodynamic formalism*, 2004 (Cambridge University Press, Cambridge, UK).
- 17 Gupta, S. and Ray, A. Pattern identification using lattice spin systems: a thermodynamic approach. *Appl. Phys. Lett.*, 2007, **19**, 194105-1–194105-3.
- 18 Subbu, A. and Ray, A. Space partitioning via Hilbert transform for symbolic time series analysis. *Appl. Phys. Lett.*, 2008, **92**(8), 084107-1–084107-3.
- 19 Buhl, M. and Kennel, M. Statistically relaxing to generating partitions for observed time-series data. *Phys. Rev. E*, 2005, **71**, 1–14.
- 20 Kantz, H. and Schreiber, T. *Nonlinear time series analysis*, second edition, 2004 (Cambridge University Press, Cambridge, UK).
- 21 Bapat, R. B. and Raghavan, T. E. S. *Nonnegative matrices and applications*, 1997 (Cambridge University Press, Cambridge, UK).
- 22 Rao, C. Diversity: its measurement, decomposition, apportionment and analysis. *Sankhya: Indian J. Stat., Series A, Part 1*, 1982, **44**, 1–22.
- 23 Chin, S., Ray, A., and Rajagopalan, V. Symbolic time series analysis for anomaly detection: a comparative evaluation. *Signal Process.*, 2005, **85**(9), 1859–1868.
- 24 Rao, C., Ray, A., Sarkar, S., and Yasar, M. Review and comparative evaluation of symbolic dynamic filtering for detection of anomaly patterns. *Signal, Image Video Process.*, 2008, DOI 10.1007/s11760-008-0061-8.
- 25 Gupta, S., Ray, A., and Keller, E. Symbolic time series analysis of ultrasonic data for early detection of fatigue damage. *Mech. Syst. Signal Process.*, 2007, **2**, 866–884.
- 26 Gerkey, B., Vaughan, R., and Howard, A. The player/stage project: tools for multi-robot and distributed sensor systems. In Proceedings of the International Conference on *Advanced robotics (ICAR 2003)*, Coimbra, Portugal, 30 June–3 July 2003, pp. 317–323.
- 27 Mallapragada, G., Chattopadhyay, I., and Ray, A. Autonomous navigation of mobile robots using optimal control of finite state automata. In Proceedings of the Conference on *Decision and control*, San Diego, CA, December 2006, pp. 2400–2405 (IEEE).