In: Pattern Recognition: Theory and Application
Editor: Erwin A. Zoeller, pp. 17-71
ISBN 978-1-60021-717-3

Chapter 2

SYMBOLIC DYNAMIC FILTERING FOR DATA-DRIVEN PATTERN RECOGNITION*

Shalabh Gupta † ***and Asok Ray*** ‡
Mechanical Engineering Department, The Pennsylvania State University
University Park, PA 16802, USA

Abstract

Gradual development of anomalies (i.e., deviations from the nominal condition) may alter the quasi-static behavior patterns of human-engineered complex systems. This chapter presents a recently reported technique of pattern recognition, called Symbolic Dynamic Filtering (*SDF*), for early detection and prognosis of such changes in behavior patterns due to slowly evolving anomalies that may be benign or malignant. The underlying concept of *SDF* is built upon the principles of *Symbolic Dynamics*, *Information Theory*, and *Statistical Signal Processing*, where time series data from selected sensor(s) in the *fast* time scale of the process dynamics are analyzed at discrete epochs in the *slow* time scale of anomaly evolution. The key idea here is early detection and identification of (possible) changes in quasi-static statistical patterns of the dynamical system behavior. An important feature of this pattern recognition technique is extraction of the relevant statistics by conversion of the time series data into a symbol sequence by appropriate coarse-graining of the imbedded information. As an alternative to the currently practiced method of phase-space partitioning in the time domain, a new concept of partitioning is introduced for symbol generation, based on wavelet analysis of the time series data. This chapter also discusses various aspects of the wavelet-based partitioning tool, such as selection of the wavelet basis, noise mitigation, and robustness to spurious disturbances. The partitioning scheme is built upon the principle of *maximum entropy* such that the regions of the data space with more information are partitioned finer and those with sparse information are partitioned coarser. The algorithms of *SDF* are constructed to solve two problems: (i) *Forward problem of Pattern Recognition* for (offline) characterization of the anomalous behavior, relative to the nominal behavior; and (ii) *Inverse problem of Pattern*

*This work has been supported in part by the U.S. Army Research laboratory and the U.S. Army Research Office under Grant No. W911NF-07-1-0376 and by NASA under Cooperative Agreement No. NNX07AK49A.
†E-mail address: szg107@psu.edu
‡E-mail address: axr2@psu.edu

Identification for (online) estimation of parametric or non-parametric changes based on the knowledge assimilated in the forward problem and the observed time series data of quasi-stationary process response.

The concept of *SDF* has been experimentally validated on two laboratory apparatuses for identification of anomalous patterns. The first apparatus is an active nonlinear electronic system with a slowly varying dissipation parameter and the second apparatus is a special-purpose computer-controlled fatigue test machine that is instrumented with ultrasonic flaw detectors and an optical travelling microscope. Time series data of observed variables have been used to experimentally validate the *SDF* algorithm.

1. Introduction

In diverse fields of science and engineering, the underlying physical process is modelled as a finite-dimensional dynamical system in the setting of an initial value problem as:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\eta}(t), t);\ \mathbf{x}(t_0) = \mathbf{x}_0 \tag{1}$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t)) + \boldsymbol{\upsilon}(t) \tag{2}$$

where $t \in [t_0, \infty)$ denotes the time of process evolution; $\mathbf{f}$ describes the time evolution of the state trajectory; $\mathbf{g}$ represents the measurement model; $\mathbf{x} \in \mathbb{R}^n$ is the state vector; $\mathbf{u} \in \mathbb{R}^m$ is the input excitation vector; $\mathbf{y} \in \mathbb{R}^p$ is the measurement vector of sensor outputs; $\boldsymbol{\eta} \in \mathbb{R}^\ell$ is the (possibly non-additive) process noise vector; and $\boldsymbol{\upsilon} \in \mathbb{R}^q$ is the measurement noise vector.

Parameter identification and robust solutions of such models are often very difficult to achieve due to uncertain, nonlinear and nonstationary dynamics [1]. For example, no existing model can capture the dynamical behavior of fatigue damage at the grain level based on the basic fundamentals of molecular physics [2]. Furthermore, in real-time applications, the analysis becomes computationally very expensive for a high-dimensional model. In general, these models could be very sensitive to the initial and boundary conditions and also on certain system parameters. Small deviations in critical parameters may produce large variations in the evolution of the system response for (apparently) identical operating conditions. Therefore, sole reliance on model-based analysis for pattern recognition is infeasible because of the difficulty in achieving requisite accuracy with available computational resources. As such, the problem is simplified using observation-based estimation of the underlying mathematical structure of the system and its relevant parameters. Typically, a map of the dynamical process (in discrete time) is described as:

$$\mathbf{x}_{k+1} = \boldsymbol{\varphi}_k(\mathbf{x}_k, \mathbf{u}_k, \eta_k) \tag{3}$$

$$\mathbf{y}_k = \boldsymbol{\gamma}(\mathbf{x}_k, \mathbf{u}_k) + \boldsymbol{\upsilon}_k \tag{4}$$

where k is the discrete time index; $\boldsymbol{\varphi}$ describes the time evolution of the state trajectory; $\boldsymbol{\gamma}$ represents the measurement model; $\mathbf{x}$ is the state vector in the phase space; $\mathbf{u}$ is the input excitation vector; $\mathbf{y}$ is the measurement vector; $\boldsymbol{\eta}$ is the (possibly non-additive) process

noise; and υ is the measurement noise. Evolution of the dynamical process generates time series data of system outputs $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k, \ldots$ starting from an initial point $\mathbf{x}_0$. Since $\mathbf{x}$ is usually hidden and φ is generally unknown especially for anomalous systems, the problem needs to be investigated by alternative means of extraction of relevant information from the time series data set $\mathbb{Y} = \{\mathbf{y}_0, \mathbf{y}_1, ..., \mathbf{y}_k, \ldots\}$ of selected observable outputs (e.g., sensor data), as shown in Fig. 1.

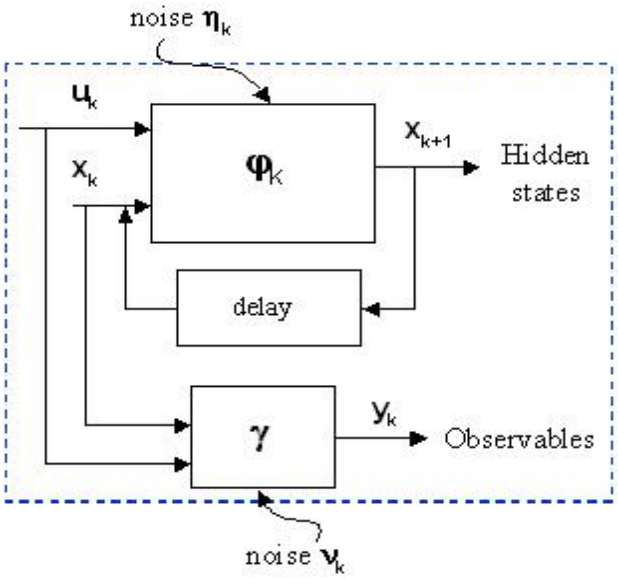

Figure 1. Measurement of the physical process by a set of observable parameters.

In view of the above discussion, the analysis of time series data from available sensors is needed for real-time pattern recognition. While there exist many reported techniques (e.g., particle filtering [3][4]) for combined model-based and data-driven pattern recognition, the real-time execution of such tools is an open research issue. This chapter addresses the problem of real-time information extraction and presents data-driven pattern recognition for early detection and prognosis of changes in behavior patterns due to slowly evolving anomalies in dynamical systems. (Note: Anomaly in a dynamical system can be defined as a deviation of its behavior pattern from the nominal pattern that is viewed as the desired healthy behavior.)

1.1. Background and Motivation

The response of a complex dynamical system can change due to the development and progression of small anomalies that gradually evolve over a long period of service life. Subsequent growth of anomalies from the nominal pattern can directly affect the performance and reliability of the system and introduce undesirable behavioral characteristics.

Anomalies can be associated with either parametric or non-parametric changes in the dynamics of the complex system. Parametric changes are usually related to the degradation in the precision of a single or multiple parameters that are used to construct the model of

the system. For example, a change in the stiffness parameter of the diaphragm of a flexible mechanical coupling between two shafts can lead to misalignments and cause the whirling. This phenomenon may increase the machine vibrations and eventually lead to failures of the bearing or the coupling. However, the change in the stiffness parameter is a consequence of the fatigue damage gradually evolving over a long period of operation. The associated changes in the system dynamics can be directly related to changes in the parameters that form an integral part of the system model. Therefore, the time series data of the observed process variables can be directly used to estimate the model parameters.

The other possible changes that can occur in a system are termed as non-parametric changes that are difficult to precisely measure, identify and model, and a direct relation of their effects on the performance variables may be unknown. However, these non-parametric changes may affect the responses of observable variables. As such, non-parametric changes can also be detected from the time series data of certain observable process variables. Often, external sensors are necessary to detect changes in such systems. The exact interpretation and quantification of these changes might not be feasible because of the lack of knowledge of the underlying physics. For example, small growth of fatigue damage in the crack initiation region can be represented as a non-parametric change, and time series data of sensors (e.g., ultrasonic flaw detector) can be used to detect the growth of small microstructural changes in the crack initiation regime [2].

The observed pattern changes are often indicatives of hidden damage that may degrade safety and reliability of machine operations. Accurate prediction and quantification of hidden damage could be infeasible due to lack of relevant information or inadequacy of analytical tools that extract such information. This problem is often circumvented by conservative enforcement of large safety factors, which could prohibitively increase the life cycle cost of operating machinery. A possible solution to reduction of overly conservative safety factors is to have frequent inspection that also turns out to be expensive and time-consuming if maintenance actions are taken based on fixed usage intervals. From these perspectives, it is logical to have on-line identification of anomalous patterns, which would allow continual re-evaluation and extension of service life and enhance inherent protection against unforeseen early failures. This information will also reduce the frequency of inspections, i.e., increase the mean time between major maintenance actions. Furthermore, early detection of anomalies and identification of incipient fault patterns are essential for prognosis of forthcoming failures to avert colossal loss of expensive equipment and human life [5].

The discussion above evinces the need for developing capabilities of pattern recognition and anomaly detection for prognosis and estimation of impending failures (e.g., the onset of wide-spread fatigue crack damage in mechanical structures) for reliable and safe operation of human-engineered systems as well as for enhanced availability of their service life. Furthermore, since modelling of the physical process could be inaccurate and infeasible for real-time execution, the information derived from relevant observed variables (i.e., sensor time series data) is often necessary to detect the resulting parametric or non-parametric changes. As such, a data-driven pattern recognition methodology has been presented in this chapter for anomaly detection and prognosis of forthcoming failures. This chapter presents an information-based technique, called Symbolic Dynamic Filtering (*SDF*) [6], for pattern recognition and online identification of anomaly patterns.

1.2. Methodology

The anomaly detection methodology is formulated to achieve the following objectives:

- *Information-based identification of anomaly progression patterns* - The possible sources of information can include time series data of appropriate sensors mounted on the critical components at different spacial locations of the complex dynamical system. The other possible sources include outputs of the analytical models that are sensitive to small changes in the system dynamics;
- *Real-time execution* - The analytical tools must be computationally efficient and have the capability of real-time execution on commercially available inexpensive platforms;
- *Capability of small change detection* - The pattern recognition methodology for anomaly detection must be sensitive to small changes and have the capability of providing early warnings of incipient faults. The methodology must also be capable of estimating fault precursors to formulate a decision and control policy for damage mitigation and life extension;
- *Robustness to measurement noise and disturbances* - The pattern recognition tool must be robust to noise and disturbances and must have low probability of false alarms.

The theme of pattern recognition and anomaly detection, formulated in this chapter, is built upon the concepts of *Symbolic Dynamics* [7], *Finite State Automata* [8], *Information Theory* and *Statistical Signal Processing* [9] as a means to qualitatively describe the dynamical behavior in terms of symbol sequences [1] [10]. The chapter presents symbolic dynamic filtering (SDF) [11] [7] [1] to analyze time series data of sensors and/or observable variables for detection and identification of gradually evolving anomalies in complex dynamical systems.

The core concept of SDF is based on appropriate phase-space partitioning of the dynamical system to yield an alphabet to obtain symbol sequences from time series data [12] [13] [14]. The time series data of appropriate sensors are processed and subsequently converted from the domain of real numbers into the domain of (discrete) symbols [7] [1]. The resulting symbol sequence is a transform of the original time series sequence such that the loss of information is minimized in the sense of *maximized entropy*. The chapter has adopted wavelet-based partitioning approach for symbol sequence generation [6] [15]. Wavelet based partitioning approach is robust and is particularly effective with noisy data [15].

Subsequently, tools of Computational Mechanics [6] [16] [17] are used to identify statistical patterns in these symbolic sequences through construction of a (probabilistic) finite-state machine [6] [8]. Transition probability matrices of the finite state machines, obtained from the symbol sequences, capture the pattern of the system behavior by means of information compression. For anomaly detection, it suffices that a detectable change in the pattern represents a deviation of the nominal pattern from an anomalous one. The state

probability vectors, which are derived from the respective state transition matrices under the nominal and an anomalous condition, yield a statistical pattern of the anomaly.

Symbolic dynamic filtering (SDF) for anomaly detection is an information-theoretic pattern recognition tool that is built upon a fixed-structure, fixed-order Markov chain, called the *D-Markov machine* [6]. Recent literature [2] [18] has reported experimental validation of SDF-based pattern recognition by comparison with other existing techniques such as Principal Component Analysis (PCA) and Artificial Neural Networks (ANN); SDF has been shown to yield superior performance in terms of early detection of anomalies, robustness to noise [15], and real-time execution in different applications such as electronic circuits [18], mechanical vibration systems [19], and fatigue damage in polycrystalline alloys [2][20].

1.3. Organization

This chapter is organized in eleven sections and three appendices. Section 1. provides a background and motivation for data-driven pattern recognition for anomaly detection. Section 2. briefly introduces the notion of nonlinear time series analysis and presents the two-time-scale problem formulation for anomaly detection using symbolic dynamic filtering. Section 3. provides a brief overview of symbolic dynamics and encoding of time series data. Section 4. presents the wavelet-based partitioning technique for symbol sequence generation. Section 5. describes the maximum entropy approach for partitioning the data. presents two ensemble approaches for statistical pattern representation. Section 6. describes the construction of a finite state machine for pattern generation from the symbol sequences. Section 7. presents the notion of anomaly measure to quantify the changing patterns of anomalous behavior of the dynamical system form the information-theoretic perspectives. Section 8. provides a brief summary of the anomaly detection procedure along with different advantages of SDF. Section 9. provides an overview of the forward and inverse problems. Section 10. presents experimental results on a nonlinear active electronic circuit and fatigue damage test apparatus to demonstrate efficacy of the SDF-based pattern recognition and anomaly detection technique. Section 11. summarizes and concludes the chapter with recommendations for future research. Appendix A. explains the physical significance of different information-theoretic quantities. Appendix B. presents a comparison of two concepts of finite sate machines that are used for construction of hidden Markov models from symbol sequences. Appendix C. introduces the concept of shift spaces.

2. Problem Formulation

This section presents the problem formulation for pattern recognition and anomaly detection based on symbolic dynamic filtering (SDF) in complex dynamical systems. The underlying concepts and essential features of SDF [6] are presented in the next section.

2.1. Non-linear Time Series Analysis for Pattern Recognition

This section presents nonlinear time series analysis ($NTSA$) that is needed to extract relevant physical information on the dynamical system from the observed data. $NTSA$ tech-

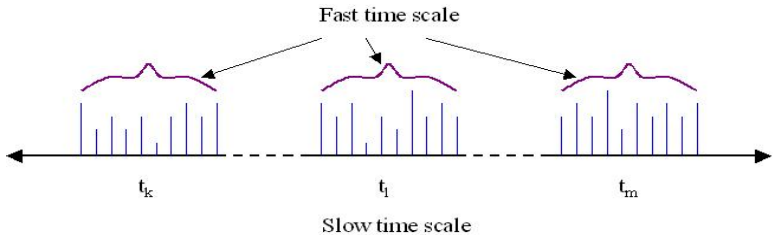

Figure 2. Pictorial view of the two time scales: 1) *slow time scale* where anomalies evolve and 2) *fast time scale* where data acquisition is done.

niques are usually executed in the following steps [12]:

1. *Signal Separation*: The (deterministic) time-dependent signal $\{y(n) : n \in \mathbb{N}\}$, where $\mathbb{N}$ is the set of positive integers, is separated from noise, using time-frequency and other types of analysis.

2. *Phase Space Reconstruction*: Based on the Takens' Embedding theorem [21], time lagged or delayed variables are used to construct the state vector $\mathbf{x}(n)$ in a phase space of dimension d_E (which is diffeomorphically equivalent to the attractor of the original dynamical system) as follows:

$$\mathbf{x}(n) = [y(n), y(n+T), \cdots, y(n+(d_E-1)T)] \tag{5}$$

   where the time lag T is determined using *mutual information*; and one of the ways to determine d_E is the *false nearest neighbors test* [12].

3. *Signal Classification*: Signal classification and system identification in nonlinear chaotic systems require a set of invariants for each subsystem of interest followed by comparison of observations with those in the library of invariants. The invariants are properties of the attractor and could be independent of any particular trajectory. These invariants can be divided into two classes: *fractal dimensions* and *Lyapunov exponents*. Fractal dimensions characterize geometrical complexity of dynamics (e.g., spatial distribution of points along a system orbit); and Lyapunov exponents describe the dynamical complexity (e.g., stretching and folding of an orbit in the phase space) [22].

4. *Modeling and Prediction*: This step involves determination of the parameters of the assumed model of the dynamics, which is consistent with the invariant classifiers (e.g., Lyapunov exponents, and fractal dimensions).

The first three steps show how chaotic systems may be separated from stochastic ones and, at the same time, provide estimates of the degrees of freedom and the complexity of the underlying dynamical system. Based on this information, Step 4 formulates a dynamic model that can be used for prediction of anomalies and incipient faults. The functional form often used in this step, includes orthogonal polynomials and radial basis functions.

This chapter has adopted an alternative class of discrete models [6] inspired from *Automata Theory* [8], which is built upon the principles of *Symbolic Dynamics* [7].

Anomaly detection using SDF is formulated as a two-time-scale problem as explained below.

- The *fast time scale* is related to the response time of process dynamics. Over the span of a given time series data sequence, the behavioral statistics of the system are assumed to remain invariant, i.e., the process is assumed to have statistically stationary dynamics at the fast time scale. In other words, statistical variations in the internal dynamics of the system are assumed to be negligible on the fast time scale.

- The *slow time scale* is related to the time span over which the process may exhibit non-stationary dynamics due to (possible) evolution of anomalies. Thus, an observable non-stationary behavior can be associated with anomalies evolving at a slow time scale.

A pictorial view of the two time scales is presented in Figure 2. In general, a long time span in the fast time scale is a tiny (i.e., several orders of magnitude smaller) interval in the slow time scale. For example, fatigue damage evolves on a slow time scale, possibly in the order of months or years, in machinery structures that are operated in the fast time scale approximately in the order of seconds or minutes. Hence, the behavior pattern of fatigue damage is essentially invariant on the fast time scale. Nevertheless, the notion of fast and slow time scales is dependent on the specific application, loading conditions and operating environment. As such, from the perspective of anomaly detection, sensor data acquisition is done on the fast time scale at different slow time epochs separated by uniform or non-uniform intervals on the slow time scale.

2.2. Procedure for Anomaly Detection

The SDF-based anomaly detection requires the following steps:

- *Time series data acquisition on the fast time scale from appropriate sensors or from the response of process variables* - Collection of data sets is done at different slow time epochs. As stated in the previous subsection, the choice of time scales is dependent on the application and requires an approximate *a priori* knowledge about the time period of evolution of anomalies.

- *Transformation of time series data from the continuous domain to the symbolic domain* - This is done by partitioning the data into finitely many discrete regions to generate symbol sequences at different slow time epochs [7] [1] (details in Section 3.). The chapter has presented a wavelet-based partitioning scheme for symbol sequence generation (details in Section 4.).

- *Construction of a finite state machine* - The machine is constructed from the symbol sequence generated at the nominal condition (details in Section 6.)

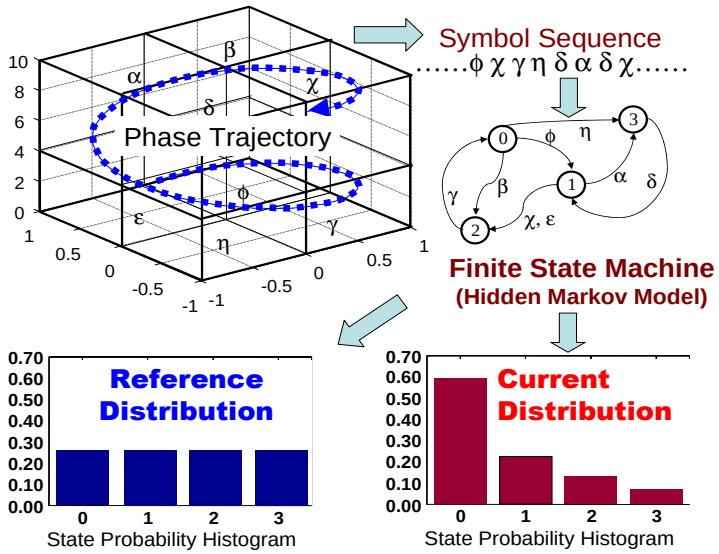

Figure 3. Conceptual view of symbolic dynamic filtering.

- *Calculation of the pattern vectors at different slow time epochs* - The elements of these pattern vectors consist of the visiting frequencies of the finite state machine states (details in Section 6.)

- *Identification of behavioral changes*- Anomaly detection is based on the information derived from the evolution of the pattern vector at different slow time epochs with respect to the one at the nominal condition (details in Section 7.)

The anomaly detection problem is separated into two parts [6]: (i) *forward problem of Pattern Recognition* for (offline) characterization of the anomalous behavior, relative to the nominal behavior; and (ii) *inverse problem of Pattern identification* for (online) estimation of parametric or non-parametric changes based on the knowledge assimilated in the forward problem and the observed time series data of quasi-stationary process response. The inverse problem could be ill-posed or have no unique solution. That is, it may not always be possible to identify a unique anomaly pattern based on the observed behavior of the dynamical system. Nevertheless, the feasible range of parameter variation estimates can be narrowed down from the intersection of the information generated from inverse images of the responses under several stimuli. The algorithms of SDF can be implemented to solve both these problems.

Often dynamical systems are either self-excited (e.g., vibrating pedestals of rotating machinery) or they can be stimulated with *a priori* known exogenous inputs to recognize (possible) anomaly patterns from the observed stationary response (e.g., ultrasonic excitation for fatigue crack detection). In both cases, it is envisioned that complex dynamical systems will acquire the ability of *self-diagnostics* through usage of the proposed anomaly detection technique that is analogous to the diagnostic procedure employed in medical practice. The latter case of self-excited excitation is similar to the notion of injecting *med-*

ication or *inoculation* on a nominally healthy patient in the sense that a dynamical system would be excited with known stimuli (chosen in the forward problem) in the idle cycles for self diagnosis and process monitoring. The inferred information on health status can then be used for the purpose of damage mitigating or life-extending control [23].

3. Symbolic Dynamics and Encoding

This section provides a brief review of the concept of *Symbolic Dynamics* and its usage for encoding nonlinear system dynamics from observed time series data. Upon collection of a time series data set at a slow time epoch, the next step is transformation from the domain of real numbers to the domain to discrete symbols. As discussed in Section 1., sole usage of a dynamic model may not always be feasible due to unknown parametric and non-parametric uncertainties and noise. A convenient way of learning the dynamical behavior is to rely on the additional information provided by (sensor-based) time series data [24][12].

A tool for behavior description of nonlinear dynamical systems is based on the concept of formal languages for transitions from smooth dynamics to a discrete symbolic description [1]. The phase space of the dynamical system is partitioned into a finite number of cells, so as to obtain a coordinate grid of the space. A compact (i.e., closed and bounded) region $\Omega \in \mathbb{R}^n$, within which the (stationary) motion under the specific exogenous stimulus is circumscribed, is identified. Encoding of Ω is accomplished by introducing a partition $\mathbb{B} \equiv \{B_0, \cdots, B_{m-1}\}$ consisting of m mutually exclusive and exhaustive cells such that

$$B_j \cap B_k = \emptyset \ \ \forall j \neq k \quad \text{and} \quad \bigcup_{j=0}^{m-1} B_j = \Omega \tag{6}$$

The dynamical system describes an orbit by the time series data as: $\mathbb{O} \equiv \{x_0, x_1 \cdots, x_k \cdots\}, x_i \in \Omega$, which passes through or touches the cells of the partition $\mathbb{B}$. Let us denote the cell visited by the trajectory at a time instant as a random variable S that takes a symbol value $s \in \mathcal{A}$. The set $\mathcal{A}$ of m distinct symbols that label the partition elements is called the *symbol alphabet* (Note: $2 \leq |\mathcal{A}| < \infty$). As the system evolves in time, the trajectory travels through various blocks in its phase space and the corresponding symbol $s \in \mathcal{A}$ is assigned to it, thus converting the time series data sequence to a symbol sequence. Each initial state $x_0 \in \Omega$ generates a sequence of symbols defined by a mapping from the phase space into the symbol space as:

$$x_0 \longrightarrow s_{i0} s_{i1} s_{i2} \cdots s_{ik} \cdots \tag{7}$$

The mapping in Eq. (7) is called *Symbolic Dynamics* as it attributes a legal (i.e., physically admissible) symbol sequence to the system dynamics starting from an initial state. (Note: A symbol alphabet $\mathcal{A}$ is called a generating partition of the phase space Ω if every legal symbol sequence uniquely determines a specific initial condition x_0, i.e., every symbolic orbit uniquely identifies one continuous space orbit.) In general, a dynamical system would only generate a subset of all possible sequences of symbols as there could be some illegal (i.e., physically inadmissible) sequences. Figure 3 pictorially elucidates the concepts of partitioning a finite region of the phase space and mapping from the partitioned space

into the symbol alphabet. This represents a spatial and temporal discretization of the system dynamics defined by the trajectories. Figure 3 also shows conversion of the symbol sequence into a finite-state machine as explained in later sections.

Symbolic dynamics can be viewed as coarse graining of the phase space, which is subjected to (possible) loss of information resulting from granular imprecision of partitioning boxes, measurement noise and errors, and sensitivity to initial conditions. However, the essential robust features (e.g., periodicity and chaotic behavior of an orbit) are expected to be preserved in the symbol sequences through an appropriate partitioning of the phase space [1]. Although the theory of phase-space partitioning is well developed for one-dimensional mappings, very few results are known for two and higher dimensional systems [10].

4. Wavelet-Based Partitioning

A crucial step in SDF is extraction of relevant information, imbedded in the measured time series series data, to generate symbol sequences. Symbol generation requires partitioning of the data space to obtain the symbol sequences [11] [24]. Various partitioning techniques have been suggested in literature for symbol generation, which include variance-based [25], entropy-based [26], and hierarchial clustering [27] methods. A survey of clustering techniques is provided in [28]. In addition to these methods, another scheme of partitioning, based on *symbolic false nearest neighbors* ($SFNN$), was reported by Kennel and Buhl [14]. The objective of $SFNN$ partitioning is to ensure that points that are close to each other in the symbol space are also close to each other in the phase space. Partitions that yield a smaller proportion of *symbolic false nearest neighbors* are considered optimal. However, this partitioning method may become cumbersome and extremely computation-intensive if the dimension of the phase space is large. Moreover, if the time series data is noise-corrupted, then the symbolic false neighbors would rapidly grow in number and require a large symbol alphabet to capture the pertinent information on the system dynamics. Therefore, symbolic sequences as representations of the system dynamics should be generated by alternative methods because phase-space partitioning might prove to be a difficult task in the case of high dimensions and presence of noise. The wavelet transform [29] largely alleviates these shortcomings and is particularly effective with noisy data from high-dimensional dynamical systems. As such, this chapter has presented a wavelet-based partitioning approach [6] [15] for construction of symbol sequences from the time series data.

4.1. Wavelet Analysis of Time Series Data

This section presents generation of wavelet coefficients from measured time series data, and their arrangement for symbol generation. Specifically, issues of wavelet basis and scale range selection are also addressed. A wavelet is a function $\psi \in \mathbf{L^2}(\mathbb{R})$ with a zero average:

$$\int_{-\infty}^{\infty} \psi(t)dt = 0. \tag{8}$$

which is normalized such that $||\psi||_2 = 1$. The wavelet transform of a function $f(t) \in \mathbb{H}$ is given by

$$F_{\alpha,\beta} = \frac{1}{\sqrt{\alpha}} \int_{-\infty}^{\infty} f(t)\psi^*(\frac{t-\beta}{\alpha})dt, \tag{9}$$

where $\alpha > 0$ is the scale, β is the time shift, and $\mathbb{H}$ is a Hilbert space. Wavelet analysis alleviates the difficulties associated with Short-Time Fourier Transform via adaptive usage of long windows for retrieving low frequency information and short windows for high frequency information [29]. The ability to perform flexible localized analysis is one of the striking features of wavelet transform. In addition to this, wavelet preprocessing helps in noise mitigation.

In multi-resolution analysis (MRA) of wavelet transform, a continuous signal $f \in \mathbb{H}$, where $\mathbb{H}$ is a Hilbert space, is decomposed as a linear combination of time translations of scaled versions of a suitably chosen scaling function $\phi(t)$ and the derived wavelet function $\psi(t)$. Let the sequence $\{\phi_{j,k}\}$ belong to another Hilbert space $\mathbb{M}$ with a countable measure, where the scale $s = 2^j$ and time translation $\tau = 2^{-j}k$. If the sequence $\{\phi_{j,k}\}$ is a frame for the Hilbert space $\mathbb{H}$ with a frame representation operator $\mathbb{L}$, then there are positive real scalars *A* and *B* such that:

$$A||f||^2_{\mathbb{H}} \leq ||\mathbb{L}f||^2_{\mathbb{M}} \leq B||f||^2_{\mathbb{H}} \ \forall f \in \mathbb{H}, \tag{10}$$

where $\mathbb{L}f = \{< f, \phi_{j,k} >\}$ and $||\mathbb{L}f||_{\mathbb{M}}$ is an appropriate norm, e.g., $||\mathbb{L}f||_{\mathbb{M}} = \sqrt{\sum_j \sum_k | < f, \phi_{j,k} > |^2}$ is a candidate norm; and $< x, y >$ is the inner product of x and y, both belonging to $\mathbb{H}$ [15]. The above relationship is a norm equivalence and represents the degree of coherence of the signal f with respect to the frame set of scaling functions; it may be interpreted as enforcing an approximate energy transfer between the domains $\mathbb{H}$ and $\mathbb{L}(\mathbb{H})$. A measure of coherence between the signal and wavelet may be obtained from the cross-correlation between them that is defined as

$$\Gamma_{f,\psi_\alpha} = \frac{< f, \psi_\alpha >}{||f||_2 ||\psi_\alpha||_2} \tag{11}$$

where ψ_α is the suitably scaled wavelet and $< f, \psi_\alpha >$ is the inner product between the vectors f and ψ_α.

In other words, for all signals $f \in \mathbb{H}$, a scaled amount of energy is distributed in the coefficient domain where the scale factor lies between *A* and *B* [29]. However, the energy distribution is dependent on the signal's degree of coherence with the underlying frame $\{\phi_{j,k}\}$. For a signal f, which is coherent with respect to the frame $\{\phi_{j,k}\}$, norm equivalence in the frame representation necessarily implies that a few coefficients contain most of the signal energy and hence have relatively large magnitudes. Similarly, pure noise signal w being incoherent with respect to the set $\{\phi_{j,k}\}$, must have a frame representation in which the noise energy is spread out over a large number of coefficients. Consequently, these coefficients have a relatively small magnitude [30].

Let $\tilde{f}$ be a noise corrupted version of the original signal f expressed as:

$$\tilde{f} = f + \sigma\, w, \tag{12}$$

where w is additive white gaussian noise with zero mean and unit variance and σ is the noise level. Then, the inner product of $\tilde{f}$ and $\phi_{j,k}$ is obtained as:

$$< \tilde{f}, \phi_{j,k} >= \underbrace{< f, \phi_{j,k} >}_{signal\ part} + \sigma \underbrace{< w, \phi_{j,k} >}_{noise\ part}. \tag{13}$$

The noise part in Eq. (13) may further be reduced if the scales over which coefficients are obtained are properly chosen.

Wavelet preprocessing of time series data for SDF consists of three steps namely

1. Selection of appropriate wavelet basis
2. Selection of scales
3. Generation of wavelet coefficients for the chosen scales

4.1.1. Selecting a Wavelet Basis

Choice of wavelet primarily depends on the signal being analyzed. However, there are few properties that may be considered while selecting a wavelet basis. These are described below:

- *Time-Frequency Localization:*
  A wavelet transform derives its strength from its potential ability to localize the energy of the signal in the time-scale plane. In turn, a wavelet transform's localizing ability is directly inherited from the analyzing wavelet. If the analyzing wavelet is not well localized either in time and/or frequency, then the wavelet transform will exhibit the same non-locality. But localization in one domain comes necessarily at the cost of another. The uncertainty principle [30] determines the time and frequency localization of the wavelet. As extreme cases 'haar' wavelet is well localized in time but not in frequency while it is vice-versa with 'sinc' wavelet. Figure 4 depicts the 'haar' wavelet and its fourier transform magnitude while the 'sinc' wavelet and its fourier transform magnitude are shown in Figure 5. 'Gaussian' wavelets [31] provide better localization when both domains are taken into consideration.

- *Vanishing moments:*
  A wavelet ψ has p vanishing moments if

$$\int_{-\infty}^{\infty} t^k \psi(t) dt = 0 \ \ for\ 0 \ \leq \ k \ < \ p \tag{14}$$

  This means that ψ is orthogonal to any polynomial of degree $p-1$. If a function f is smooth and ψ has enough vanishing moments, then the wavelet coefficients $< f, \psi_{j,k} >$ are small at fine scales [29]. This property is quite beneficial in capturing faults/anomalies which often manifest as higher order terms in a power series expansion of a signal.

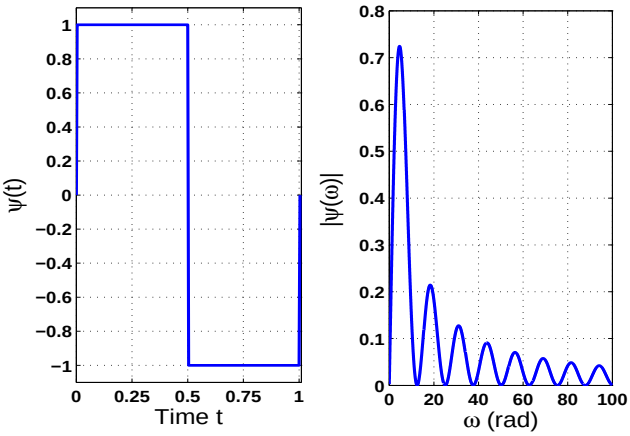

Figure 4. Haar Wavelet and its Fourier Transform Magnitude

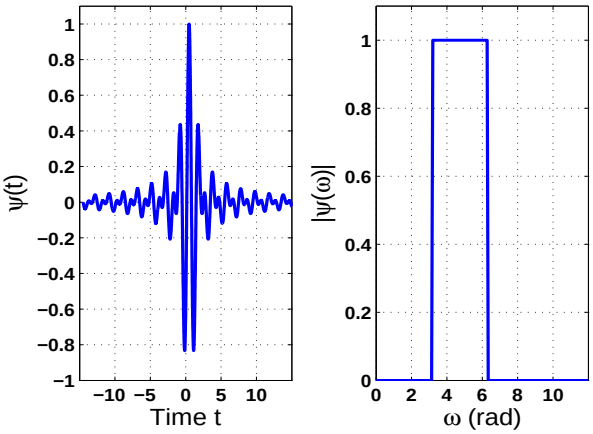

Figure 5. Sinc Wavelet and its Fourier Transform Magnitude

- *Support:*
  The support size and number of vanishing moments are a priori independent in general. However for orthogonal wavelets, p vanishing moments imply that the support is at least of size $2p-1$. When choosing a wavelet, there is a trade-off between the number of vanishing moments and support [29]. From the perspective of fault/anomaly detection while the increase in vanishing moments enhances detection, the increase in support leads to increased computation. Daubechies wavelets are optimal in the sense that they have minimal support for a given number of vanishing moments.

Many wavelets may satisfy one or more of these desirable properties. In such a case, it is advantageous to choose a wavelet basis that is coherent with the signal. A signal is coherent with a set of functions if its inner product representation with respect to that set is succinct in the sense that relatively few coefficients in the representation domain have large magnitude [30]. Accordingly noise may be viewed as lack of coherence with respect to the

set of functions.

4.1.2. Choice of Wavelet Scales

For every wavelet, there exists a certain frequency called the center frequency F_c that has the maximum modulus in the Fourier transform of the wavelet [32]. The pseudo-frequency f_p of the wavelet at a particular scale α is given by the following formula [33]:

$$f_p = \frac{F_c}{\alpha \, \Delta t}, \tag{15}$$

where Δt is the sampling interval. Figure 6 depicts the center frequency associated with the Daubechies wavelet 'db4' [29] [31]. The Power Spectral Density (PSD) of the signal provides the information about the frequency content of the signal. This information along with Eq. (15) can be used for scale selection. The procedure of selecting the scales is summarized below:

- Identification of the frequencies of interest through PSD analysis of time series data
- Substitution of the above frequencies in place of f_p in Eq. (15) to obtain the respective scales in terms of the known parameters F_c and Δt

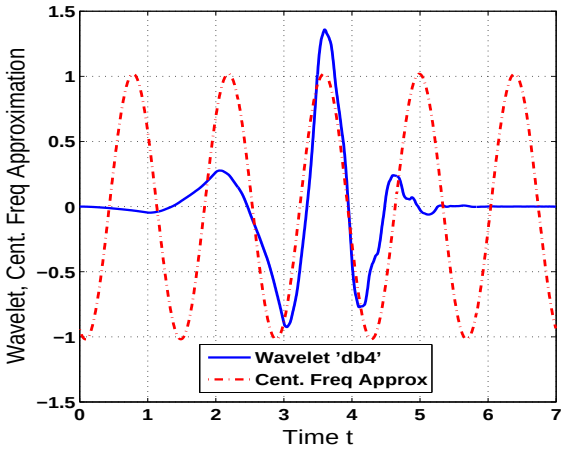

Figure 6. Center Frequency Approximation for Wavelet db4

The wavelet coefficients of the signal are significantly large when the pseudo-frequency f_p of the wavelet corresponds to the locally dominant frequencies in the underlying signal. Upon selection of the wavelet basis and scale range, the wavelet coefficients are obtained using Eq. 9.

4.2. Symbolization of Wavelet-data

The wavelet coefficients are stacked at selected time-shift positions, starting with the smallest value of scale and ending with its largest value and then back from the largest value

to the smallest value of the scale at the next instant of time shift. In the sequel, this one-dimensional array of re-arranged wavelet coefficients is called the *scale series* data, which is structurally similar to time series data in the phase space. For symbol generation, the scale series data can be handled in a similar way as time series data. The scale series data are then partitioned to construct the symbol alphabet and to generate symbol sequences. Subsequent analysis in SDF methodology depends on statistics of symbols rather than their order of appearance. Hence ordering of wavelet coefficients does not have a significant impact. Moreover, *scale series* construction simplifies the process of symbol generation.

In the wavelet-based partitioning scheme the maximum and minimum of the scale series are calculated and the ordinates between the maximum and minimum are divided into equal-sized regions. These regions are mutually disjoint and thus form a partition. Each region is then labelled with one symbol from the alphabet. If the data point lies in a particular region, it is coded with the symbol associated with that region. Thus, a sequence of symbols is created from a given sequence of scale series data. This type of partitioning is called uniform partitioning. Note that the partition segments in uniform partitioning are of equal size. Intuitively, it is more reasonable if the information-rich regions of the data set are partitioned finer and those with sparse information are partitioned coarser. To achieve this objective, a partitioning method is adopted such that the entropy of the generated symbol sequence is maximized [15]. Details of maximum entropy partitioning are presented in Section 5..

As an alternative to the partitioning of *scale series* data different partitions can also be constructed for each chosen scale and subsequently statistical information can be extracted from the corresponding symbol sequences generated from each scale separately. This enables more specific information extraction at each scale. Another alternative is to treat each scale as a separate dimension and the coefficients of that scale can be considered as the evolution in time in that scale. This leads to a multi-dimensional scale space analogous to phase space. This multi-dimensional space can then be partitioned to generate symbol sequences [34].

4.3. Validation of Wavelet-Based Partitioning

This section presents simulation cases to validate symbolization of measured time series data via partitioning of the wavelet coefficients. The underlying concepts are illustrated by two examples [15][34]. Example 1 illustrates how the choice of basis and scale affect the wavelet transform coefficients and example 2 illustrates noise suppression using wavelets.

4.3.1. Example 1: Choice of Wavelet Parameters

This example illustrates how the choice of wavelet basis and scale range affects the coefficients that, in turn, determine symbol generation for pattern recognition and anomaly detection [6]. Let us consider the following signal,

$$y(t) = cos(2\pi t) \qquad \forall t \in [-5, +5]. \tag{16}$$

The frequency of $y(t)$ in Equation (16) is 1.00 Hz. The signal $y(t)$ is sampled at 100 Hz (i.e., the sampling interval $\Delta t = 0.01s$). The only information that is necessary to describe

this signal is its frequency. So a wavelet well localized in the frequency domain would be ideally suited for analyzing this signal. Since gaussian wavelets provide good localization in frequency, they are considered suitable candidates. To demonstrate the importance of choosing a suitable wavelet the signal is also analyzed with wavelet 'db1' which suffers from poor frequency localization.

The next step in wavelet selection is determining the best wavelet in the gaussian family. For this purpose, the correlation measure in Equation (11) is utilized. Table 1 provides the correlation of the signal with gaussian wavelets 1 through 10.

Table 1. Gaussian Wavelet Correlation

Wavelet	Γ
$gaus1$	0.5960
$gaus2$	0.5939
$gaus3$	0.5965
$gaus4$	0.5726
$gaus5$	0.5949
$gaus6$	0.5734
$gaus7$	0.5948
$gaus8$	0.5745
$gaus9$	0.5901
$gaus10$	0.5908

It is observed that the correlation of all wavelets with the signal are almost equal. This is expected since gaussian wavelets are successive derivatives of the gaussian scaling function. For analysis, wavelet 'gaus3' is chosen. Figure 7 depicts an appropriately scaled and translated version of the 'gaus3' wavelet with the signal $y(t)$.

The wavelet coefficients of the signal $y(t)$ are obtained for various scales with both wavelets, 'gaus3' and 'db1'. The norm of the coefficients corresponding to each scale and the pseudo-frequencies of the wavelet corresponding to the chosen scales are calculated. Figure 8 shows the plot of the norm of coefficients and the pseudo-frequencies of the wavelet.

It is observed in Figure 8 that, for both wavelets 'gaus3' and 'db1', the maximum of the norm is obtained at $f_p \approx 1.00$ Hz. In fact, it is exactly at 1.00 Hz for 'gaus3'. Furthermore, the value of the peak norm achieved with 'gaus3' is appreciably greater than that with wavelet 'db1'. In other words, the coefficients obtained with 'gaus3' are more significant, at select scales, than those obtained with 'db1'. Another observation is that the norm curve for 'gaus3' shows a greater rate of decay across pseudo-frequencies than that of 'db1'. More energy is concentrated in a narrow band frequencies around 1.00 Hz in the case of 'gaus3'. These observations imply that high energy compaction can be achieved with fewer coefficients if the wavelet and the scales are chosen as stated in Sections 4.1.1.and 4.1.2.. A favorable implication of fewer coefficients is fewer number of symbols for analysis and

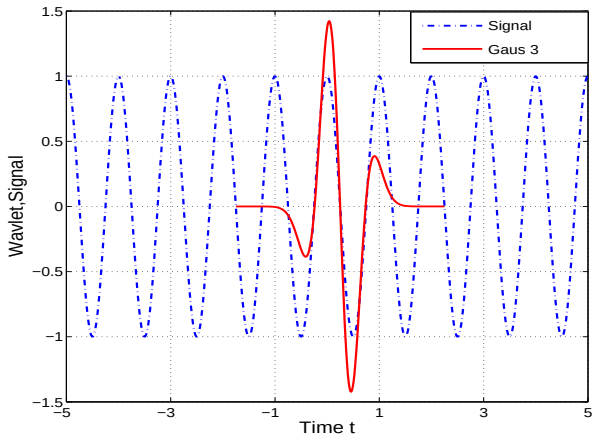

Figure 7. Plot of Wavelet Basis 'gaus3' and Signal [34]

hence an improvement in computational efficiency.

4.3.2. Example 2: Noise Suppression

This example demonstrates the noise suppression achieved with wavelets. Let the signal $y(t)$ in Eq. (16) be corrupted with additive zero-mean white Gaussian noise $w(t)$,

$$\tilde{y}(t) = y(t) + \sigma\, w(t). \tag{17}$$

A common measure of noise in a noise-corrupted signal is the signal-to-noise ratio that is defined as:

$$SNR \triangleq \frac{||y||^2_{\mathbb{H}}}{||\sigma\, w||^2_{\mathbb{H}}}, \tag{18}$$

where y and w are functions of time. Similar to Equation (18), the signal-to-noise ratio in the wavelet domain is defined as:

$$\widetilde{SNR} \triangleq \frac{||\mathbb{L}y||^2_{\mathbb{M}}}{||\sigma\, \mathbb{L}w||^2_{\mathbb{M}}}, \tag{19}$$

where $\mathbb{L}y$ and $\mathbb{L}w$ represent the wavelet coefficients of the signal y and the noise w.

Numerical experiments have been performed with $\sigma \in \{0.05, 0.1\}$. The signal is sampled at 100 Hz (i.e., $\Delta t = 0.01s$). The scales are determined following Eq. (15), such that the pseudo-frequency of the wavelet matches the frequency of the signal. Figure 9 depicts the time domain plot (left plate) and coefficient plot (right plate) of the signal y and white Gaussian noise having standard deviation $\sigma = 0.05$. Similarly, Figure 10 depicts the time domain plot (left plate) and coefficient plot (right plate) of the signal y and white Gaussian noise having standard deviation $\sigma = 0.10$. Table 2 lists the values of SNR and $\widetilde{SNR}$, averaged over 20 simulation runs.

It can be observed from Table 2 that $\widetilde{SNR} \gg SNR$ which implies that the wavelet-transformed signal is significantly de-noised relative to the time-domain signal. This is

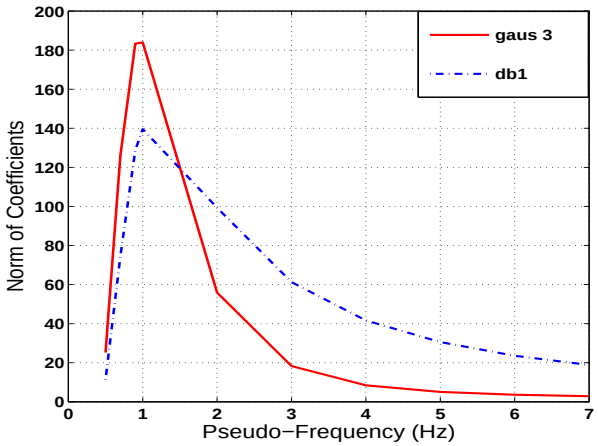

Figure 8. Coefficient Norm and Pseudo-Frequency for Different Wavelets [34]

Table 2. SNR Values

	$\sigma = 0.05$	$\sigma = 0.1$
SNR	191.55	50.89
$\widetilde{SNR}$	25195	4281.5

expected because the noise is incoherent with the wavelet while the signal enjoys a great degree of coherence with the wavelet. Thus, symbols generated from wavelet coefficients would reflect the characteristics of the signal with more fidelity than those obtained with time domain signals.

5. Maximum Entropy Partitioning

As discussed earlier in Section 4.2. the partitioning is done such that the regions with more information are partitioned finer and those with sparse information are partitioned coarser. This is achieved by maximizing the Shannon entropy [35], which is defined as:

$$H = -\sum_{i=1}^{|\mathcal{A}|} p_i \, log(p_i) \tag{20}$$

where p_i is the probability of the i^{th} segment of the partition and summation is taken over all segments. As a consequence of maximum entropy, uniform probability distribution of states is obtained (i.e., $p_i = \frac{1}{|\mathcal{A}|} \; \forall$ i) that makes the partition coarser in regions of low data density and finer in regions of high data density. Figure 11 shows an example of wavelet-based maximum entropy partitioning from the time series data of ultrasonic signals generated from a special purpose fatigue damage test apparatus [2]. The partitioning in Figure 11 is shown for alphabet set $\mathcal{A}$=$\{0,1,...,5\}$ for alphabet size $|\mathcal{A}| = 6$.

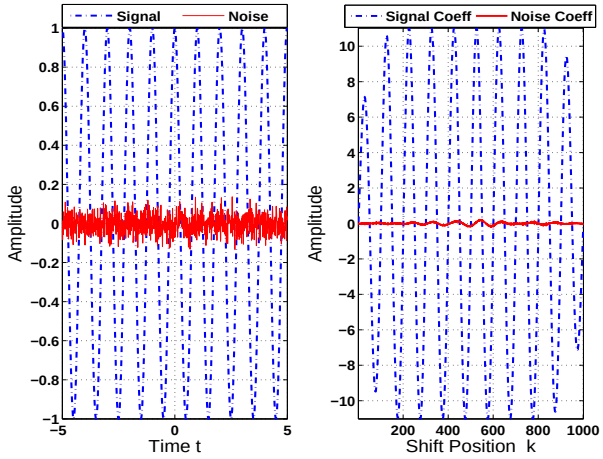

Figure 9. Signal and Noise Profiles at $\sigma = 0.05$ [34]

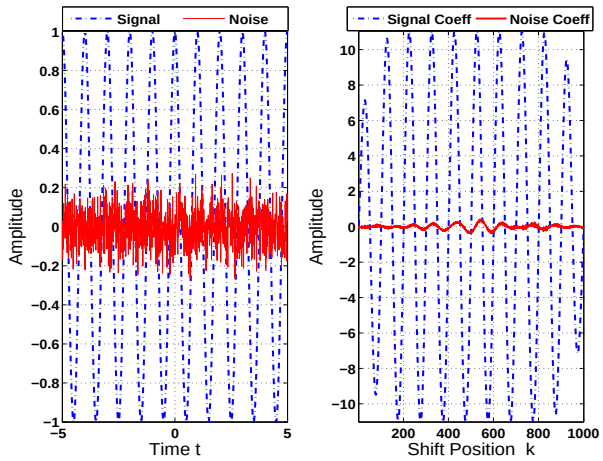

Figure 10. Signal and Noise Profiles at $\sigma = 0.10$ [34]

A comparison of wavelet-based partitioning using maximum entropy principle and other partitioning approaches such as using *symbolic false nearest neighbors* [14] and uniform partitioning is reported in recent publications [15] where wavelet-based partitioning has shown comparable performance with several orders of magnitude smaller execution time. This feature is well suited for real-time applications for early detection of anomalies. However, construction of an optimum partitioning scheme for symbol sequence generation is still an area of active research and is suggested as a future work.

5.1. Algorithm of Maximum Entropy Partitioning

The algorithm for obtaining maximum entropy partition is straightforward and is described in this section. Since the consequence of maximum entropy is uniform distribution of states,

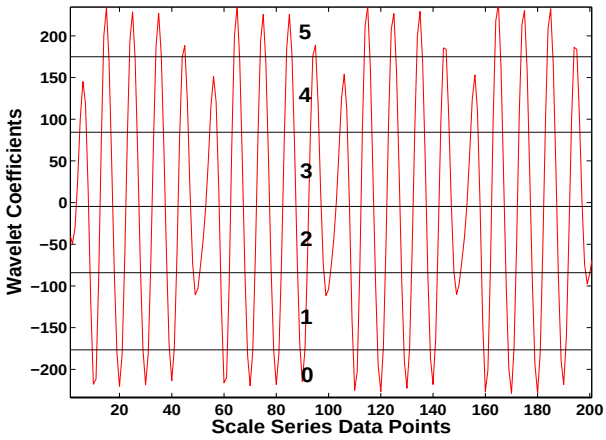

Figure 11. Wavelet space maximum entropy partitioning of ultrasonic data from a special purpose fatigue test apparatus [2].

the algorithm consists of determining the partition such that each element has equal number of data-points. Let N be the length of the data set and $|\mathcal{A}|$ be the size of the symbol alphabet (i.e., the number of disjoint elements in the partition). The data set is sorted in ascending order. Starting from the first point in the sorted data, every consecutive data segment of size $\lfloor \frac{N}{|\mathcal{A}|} \rfloor$ is obtained that forms a distinct element of the partition, where $\lfloor x \rfloor$ represents the greatest integer less than or equal to x. This procedure generates the maximum entropy partitioning.

5.2. Selection of Alphabet Size

Selection of the alphabet size $|\mathcal{A}|$ is an area of active research; an entropy-based approach has been adopted for selecting $|\mathcal{A}|$ in this chapter. Let $H(k)$ denote the Shannon entropy of the symbol sequence obtained by partitioning the data set with k symbols.

$$H(k) = -\sum_{i=1}^{i=k} p_i log(p_i), \tag{21}$$

where $H(1) = 0$ because $p_i = 1$ with $i = 1$. If the underlying data set has sufficient information content, then the entropy achieved under maximum entropy partitioning would be $log(k)$, which corresponds to the uniform distribution. We define a quantity $h(\cdot)$ to represent the change in entropy with respect to the number $|\mathcal{A}|$ of symbols as,

$$h(k) \triangleq H(k) - H(k-1) \quad \forall k \geq 2. \tag{22}$$

The algorithm for alphabet size selection is given below.

Step 1: Set k = 2. Choose a threshold ϵ_h, where $0 < \epsilon_h << 1$.

Step 2: Sort the data set (of length N) in the ascending order.

Step 3: Every consecutive segment of length $\lfloor \frac{N}{k} \rfloor$ in the sorted data set (of length N) forms a distinct element of the partition.

Step 4: Convert the raw data into a symbol sequence with the partition obtained in Step 3. If the data point lies within or on the lower bound of a partition, it is coded with the symbol associated with that partition.

Step 5: Compute the symbol probabilities p_i, i=1,2,...k.

Step 6: Compute
$H(k) = -\sum_{i=1}^{i=k} p_i log_2 p_i$ and
$h(k) = H(k) - H(k-1)$.

Step 7: If $h(k) < \epsilon_h$, then exit and set $|\mathcal{A}| = k$; else increment k by 1 and go to Step 3.

In general, a small ϵ_h leads to a large size of the symbol alphabet, resulting in increased computation. Also a larger alphabet makes the partitioning finer. This might increase the probability of false symbols being induced by noise. On the other hand, a large ϵ_h leads to a small alphabet size that may prove inadequate for capturing the pertinent information. Hence, there is a trade-off between accuracy and computational speed when ϵ_h is chosen. The variance of the noise process associated with the signal may serve as a guideline for selection of ϵ_h.

For the purpose of pattern recognition and anomaly detection, the partitioning is performed with alphabet size $|\mathcal{A}|$ at the nominal condition (time epoch t_0), and subsequently it is kept constant for all (slow time) epochs $\{t_1, t_2, ...t_k....\}$, i.e. the structure of the partition is fixed at the nominal condition. In other words, the partitioning structure generated at the nominal condition serve as the reference frame for data analysis at subsequent slow time epochs.

6. Finite State Machine Construction

Given the intricacy of phase trajectories in complex dynamical systems, the challenge is to identify their *patterns* in an appropriate category by using one of the following two alternative approaches:

- The single-item approach, which relies on Kolmogorov Chiatin (KC) complexity, also known as algorithmic complexity [35], for exact pattern regeneration;
- The ensemble approach, which regards the pattern as one of many possible experimental outcomes, for estimated pattern regeneration.

While the single-item approach is common in coding theory and computer science, the ensemble approach has been adopted in this chapter due to its physical and statistical relevance. As some of the legal symbol sequences may occur more frequently than others, a probability is attributed to each observed sequence. The collection of all legal symbol

sequences $S_{-M} \cdots S_{-2}S_{-1}S_0S_1 \cdots S_N,\ N, M = 0, 1, 2 \cdots$, defines a stochastic process that is a symbolic probabilistic description of the continuous system dynamics.

Let us symbolically denote a discrete-time, discrete-valued stochastic process as:

$$\mathbb{S} \equiv \cdots, S_{-2}S_{-1}S_0S_1S_2 \cdots \tag{23}$$

where each random variable S_i takes exactly one value in the (finite) alphabet $\mathcal{A}$ of m symbols (see Section 3.). The symbolic stochastic process $\mathbb{S}$ is dependent on the specific partitioning of the phase space and is non-Markovian, in general. Even if a partitioning that makes the stochastic process a Markov chain exists, identification of such a partitioning is not always feasible because the individual cells may have fractal boundaries instead of being simple geometrical objects. In essence, there is a trade-off between selecting a simple partitioning leading to a complicated stochastic process, and a complicated partitioning leading to a simple stochastic process. Having defined a partition of the phase space, the time series data is converted to a symbol sequence that, in turn, is used for construction of a finite state machine using the tools of Computational Mechanics [16] as illustrated in Figure 3.

This chapter presents a new information-theoretic technique based on D^{th} order Markov chains for finite-state machine construction from a given symbol sequence $\mathcal{S}$ for identifying patterns based on time series analysis of the observed data. At any instant t, the sequence of random variables can be split into a sequence $\overleftarrow{S}_t$ of the past and a sequence $\overrightarrow{S}_t$ of the future. Assuming conditional stationarity of the symbolic process $\mathbb{S}$ (i.e., $P[\overleftarrow{S}_t|\overrightarrow{S}_t = \overrightarrow{s}]$ being independent of t), the subscript t can be dropped to denote the past and future sequences as $\overleftarrow{S}$ and $\overrightarrow{S}$, respectively. A symbol string, made of the first L symbols of $\overrightarrow{S}$, is denoted by $\overrightarrow{S}^L$. Similarly, a symbol string, made of the last L symbols of $\overleftarrow{S}$, is denoted by $\overleftarrow{S}^L$.

Prediction of the future $\overrightarrow{S}$ necessitates determination of its probability conditioned on the past $\overleftarrow{S}$, which requires existence of a function ϵ mapping histories $\overleftarrow{s}$ to predictions $P(\overrightarrow{S}|\overleftarrow{s})$. In essence, a prediction imposes a partition on the set $\overleftarrow{\mathbf{S}}$ of all histories. The cells of this partition contain histories for which the same prediction is made and are called the *effective states* of the process under the given predictor.

6.1. The D-Markov Machine

This section presents a new alternative approach for representing the pattern in a symbolic process, which is motivated from the perspective of anomaly detection. The core assumption here is that the symbolic process can be represented to a desired level of accuracy as a D^{th} order Markov chain, by appropriately choosing $D \in \mathbb{N}$.

Definition 6..1 *A stochastic symbolic stationary process $\mathbb{S} \equiv \cdots S_{-2}S_{-1}S_0S_1S_2 \cdots$ is called D^{th} order Markov process if the probability of the next symbol depends only on the previous (at most) D symbols, i.e. the following condition holds:*

$$P(S_i|S_{i-1}S_{i-2} \cdots S_{i-D} \cdots) = P(S_i|S_{i-1} \cdots S_{i-D}) \tag{24}$$

Alternatively, symbol strings $\overleftarrow{S}, \overleftarrow{S}' \in \overleftarrow{\mathbf{S}}$ become indistinguishable whenever the respective substrings $\overleftarrow{S}^D$ and $\overleftarrow{S}'^D$, made of the most recent D symbols, are identical.

Thus, a set $\{\overleftarrow{S}^L : L \geq D\}$ of symbol stings can be partitioned into a maximum of $|\mathcal{A}|^D$ equivalence classes where $\mathcal{A}$ is the symbol alphabet. Each symbol string in $\{\overleftarrow{S}^L : L \geq D\}$ either belongs to one of the $|\mathcal{A}|^D$ equivalence classes or has a distinct equivalence class. All such symbol strings belonging to the distinct equivalence class form transient states, and would not be of concern to anomaly detection for a (fast-time-scale) stationary condition under (slowly changing) anomalies. Given $D \in \mathbb{N}$ and a symbol string $\overleftarrow{s}$ with $|\overleftarrow{s}| = D$, the *effective* state $q\,(D, \overleftarrow{s}\,)$ is the equivalence class of symbol strings as defined below:

$$q(D, \overleftarrow{s}\,) = \{\overleftarrow{S} \in \overleftarrow{\mathbf{S}} : \overleftarrow{S}^D = \overleftarrow{s}\,\} \tag{25}$$

and the set $\mathbf{Q}(D)$ of *effective* states of the symbolic process is the collection of all such equivalence classes. That is,

$$\mathbf{Q}(D) = \{q(D, \overleftarrow{s}\,) : \overleftarrow{s} \in \overleftarrow{\mathbf{S}}^D\} \tag{26}$$

and hence $|\mathbf{Q}(D)| = |\mathcal{A}|^D$. A random variable for a state in the above set $\mathbf{Q}$ of states is denoted by $\mathcal{Q}$ and the j^{th} state as q_j.

Definition 6..2 *The probability of transitions from state q_j to state q_k belonging to the set Q of states under a transition $\delta : Q \times \mathcal{A} \to Q$ is defined as*

$$\pi_{jk} = P\left(s \in \overrightarrow{S}^1 \mid q_j \in \mathbf{Q},\ \ (\mathrm{s}, \mathrm{q}_j) \to \mathrm{q}_k\right);\ \ \sum_k \pi_{jk} = 1; \tag{27}$$

Given an initial state and the next symbol from the original process, only certain successor states are accessible. This is represented as the allowed state transitions resulting from a single symbol. Note that $\pi_{ij} = 0$ if $s_2 s_3 \cdots s_D \neq s'_1 \cdots s'_{D-1}$ whenever $q_i \equiv s_1 s_2 \cdots s_D$ and $q_j \equiv s'_1 s'_2 \cdots s'_D$. Thus, for a D-Markov machine, the stochastic matrix $\mathbf{\Pi} \equiv [\pi_{ij}]$ becomes a branded matrix with at most $|\mathcal{A}|^{D+1}$ nonzero entries.

6.2. Machine Construction

Once the symbol sequence is obtained, the next step is the construction of a finite state machine and calculation of the state visit frequencies to generate the state probability vector as depicted in Figure 3 by the histograms. The partitioning as described in the Section 5. is performed at time epoch t_0 of the nominal condition that is chosen to be a healthy condition. A finite state machine [8] is then 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 while the window length D is the length of consecutive symbol words forming the states of the machine [6]. The states of the machine are chosen as all possible words of length D from the symbol sequence, thereby making the number n of states to be equal to the total permutations of the alphabet symbols within words of length D (i.e., $n \leq |\mathcal{A}|^D$; some of which may be forbidden with zero probability of occurrence). 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 Q=$\{$00, 01, 10, 11$\}$ as shown in Fig. 12. A large *alphabet* may be noise-sensitive while a small alphabet could miss the details of signal dynamics. Similarly, a high value of D is extremely sensitive to small

signal distortions but would lead to a large number of states requiring more computation power. Using the symbol sequence generated from the time series data, the state machine is constructed on the principle of sliding block codes [7] as explained below.

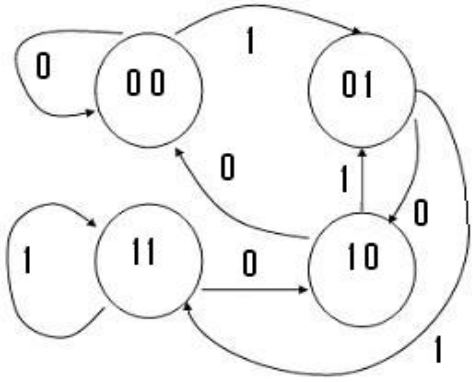

Figure 12. Finite state automaton with D=2 and $\mathcal{A} = \{0, 1\}$.

The construction of a D-Markov machine is fairly straightforward. Given $D \in \mathbb{N}$, the states are as defined in Eqs. (25) and (26). The window of length D on the symbol sequence $\ldots s_{i_1}\ s_{i_2} \ldots s_{i_k} \ldots$ 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 s_{i_ℓ} at the end. The symbolic permutation in the current window gives rise to a new state. The machine constructed in this fashion is called D-Markov machine [6] because the probability of occurrence of symbol s_{i_ℓ} on a particular state depends only on the configuration of that state, i.e., the previous D symbols. The partitioning alphabet $\mathcal{A}$ and word length D determined at the nominal condition (time epoch t_0) are kept constant for all (slow time) epochs $\{t_1, t_2, ...t_k....\}$, i.e. the structure of the machine is fixed at the nominal condition. That is, the partitioning and the state machine structure generated at the nominal condition serve as the reference frame for data analysis at subsequent slow time epochs. For D=1, the set of states bears an equivalence relation to the alphabet $\mathcal{A}$ of symbols [36]. The states of the machine are marked with the corresponding symbolic word permutation and the edges joining the states indicate the occurrence of an event s_{i_ℓ}. The occurrence of an event at a state may keep the machine in the same state or move it to a new state. The language of the machine is usually incomplete in the sense that all states might not be reachable from a given state.

Thus, for a D-Markov machine, the irreducible stochastic matrix $\mathbf{\Pi} \equiv [\pi_{ij}]$ describes all transition probabilities between states such that it has at most $|\mathcal{A}|^{D+1}$ nonzero entries. The left eigenvector $\mathbf{p}$ corresponding to the unit eigenvalue of $\mathbf{\Pi}$ is the state probability vector under the (fast time scale) stationary condition of the dynamical system [6]. On a given symbol sequence $....s_{i_1}s_{i_2}...s_{i_l}....$ generated from the time series data collected at slow time epoch t_k, a window of length D is moved by keeping a count of occurrences of word sequences $s_{i_1} \cdots s_{i_D} s_{i_{D+1}}$ and $s_{i_1} \cdots s_{i_D}$ which are respectively denoted by $N(s_{i_1} \cdots s_{i_D} s_{i_{D+1}})$ and $N(s_{i_1} \cdots s_{i_D})$. Note that if $N(s_{i_1} \cdots s_{i_D}) = 0$, then the state

$q \equiv s_{i_1} \cdots s_{i_D} \in Q$ has zero probability of occurrence. For $N(s_{i_1} \cdots s_{i_D}) \neq 0$, the transitions probabilities are then obtained by these frequency counts as follows

$$\pi_{jk} \equiv P[q_k|q_j] = \frac{P[q_k,\ q_j]}{P[q_j]} = \frac{P(s_{i_1} \cdots s_{i_D} s)}{P(s_{i_1} \cdots s_{i_D})}$$
$$\Rightarrow \pi_{jk} \approx \frac{N(s_{i_1} \cdots s_{i_D} s)}{N(s_{i_1} \cdots s_{i_D})} \tag{28}$$

where the corresponding states are denoted by $q_j \equiv s_{i_1} s_{i_2} \cdots s_{i_D}$ and $q_k \equiv s_{i_2} \cdots s_{i_D} s$. The time series data under the nominal condition (set as a benchmark) generates the *state transition matrix* $\mathbf{\Pi}^0$ that, in turn, is used to obtain the *state probability vector* $\mathbf{p}^0$ whose elements are the stationary probabilities of the state vector, where $\mathbf{p}^0$ is the left eigenvector of $\mathbf{\Pi}^0$ corresponding to the (unique) unit eigenvalue. Subsequently, state probability vectors $\mathbf{p}^1, \mathbf{p}^2, \ldots \mathbf{p}^k$.... are obtained at slow-time epochs $t_1, t_2, \ldots t_k$.... based on the respective time series data. Machine structure and partitioning should be the same at all slow-time epochs; only the entries of the $\mathbf{\Pi}$-matrix may change at different slow-time epochs. The evolution of the derived state probability vectors from the nominal condition at different slow time epochs determine the pattern changes occurring in the dynamics of the system. Having discussed the details of finite state machine construction the next few sections provide the details for the choice of parameters such as the depth D and the length of the symbol sequence required to generate the stochastic matrix $\mathbf{\Pi}$.

6.3. Selection of Depth D

The procedure of selection of the alphabet size $|\mathcal{A}|$ was discussed in Section 5.2.. This section provides a procedure for selection of the depth D. D is a crucial parameter since the number of states varies exponentially with D. A very small depth could mean insufficient memory for the D-Markov machine to appropriately represent the symbolic process. On the other hand, an unnecessarily large D would result in a large number of states, leading to extremely small values of state probabilities and an inaccurate $\mathbf{\Pi}$-matrix. A procedure based on entropy rate has been developed for selecting the depth of the D-Markov machine. The key idea is that increasing the depth beyond a certain value does not lead to any appreciable change in entropy; equivalently, the entropy rate would be very small. Given the current state, the entropy rate h_μ of a symbolic stochastic process is defined as the uncertainty in the next symbol.

$$h_\mu = -\sum_{i=1}^{n} p_i \sum_{j=1}^{|\mathcal{A}|} \tilde{\pi}_{ij} log_2 \tilde{\pi}_{ij} \tag{29}$$

where p_i is the probability of occurrence of i^{th} state; $\tilde{\pi}_{ij}$ is the probability of occurrence of j^{th} symbol in the i^{th} state; n is the number of states in the probabilistic finite state machine; and $|\mathcal{A}|$ is the alphabet size. Being a measure of uncertainty, h_μ monotonically decreases as the depth D of the D-Markov machine is increased. Beyond a certain point, increasing D will not lead to any change in the entropy rate. This is the asymptotical entropy rate and the corresponding D is optimal for the machine. With ideal noise-free data h_μ converges to zero. However, with noisy data, h_μ may only monotonically decrease to a small non-zero

value, depending on the magnitude and the type of noise. Thus, the test for the optimum D relies on how h_μ converges as D is increased.

6.4. Machine State Reduction

For a chosen depth D, the machine contains $|\mathcal{A}|^D$ states, i.e., the number of states increase in exponential steps of alphabet size $|\mathcal{A}|$. Therefore, for the purpose of state reduction the states with very small probabilities can be deleted without affecting the ability of the machine to represent the underlying symbolic process. The states of a D-Markov Machine can be classified into two categories:

- *Recurrent states*: These are the states that are visited an infinite number of times if one runs the machine infinitely.

- *Transient States*: These are the states that are visited finitely often with only small probabilities. For all practical purposes, for a transient state q_{tr}

$$\lim_{N\to\infty} P(q_{tr}) = 0$$

  where N is the length of the symbolic data string.

6.4.1. Removing Transient States

Transient states can be eliminated by setting a threshold ϵ ($0 < \epsilon \ll 1$) on the state visit probability. The value of ϵ can be chosen based on the length of the symbol sequence such as $\epsilon = \frac{1}{N}$. Therefore, all states having probability less than ϵ can be removed as transient states that are visited possibly due to noise. In addition to removing transient states, the number of states may be further reduced by merging similar states. The criterion and procedure for state merging is given below.

6.4.2. State Merging Algorithm

A symbol string u is called a descendent of its ancestor v if $u = wv$, where w is a non-null string. Similarly, a string u is called a child of v, if $u = av$, where $a \in \mathcal{A}$ such that a child has one symbol more than its parent into the past. Therefore, the following are implied:

- Any string v can have at most have $|\mathcal{A}|$ number of children.

- If $\sigma = \sigma_1\sigma_2...\sigma_D$ and $\gamma = \gamma_1\gamma_2...\gamma_D$ are children of the same parent, then $\sigma_i = \gamma_i \; \forall \; i = 2, 3, ...D$

If σ and γ are the states of D-Markov machine and are children of the same parent then upon occurrence of a new symbol 'a' they lead to the same state transition, i.e.

$$\delta(\sigma, a) = \delta(\gamma, a)$$

where δ is the state transition function. Therefore, states that are children of the same parent and have the same transition probabilities, can be merged to form a single state, i.e., if q_i and q_j are children of the same parent and if

$$\tilde{\pi}_{ik} = \tilde{\pi}_{jk} \qquad \forall \ k = 1, 2, ...|\mathcal{A}|$$

then the states q_i and q_j can be merged to form a single state; $\tilde{\pi}_{jk}$ is as defined in Eq. (29). If for depth $D > 1$, the generated $\mathbf{\Pi}$ matrix has two identical rows, then it implies that the corresponding states are children of the same parent and have the same symbol probabilities. Hence, these states can be merged. In case of noisy data, a threshold $\epsilon > 0$ is defined to check the equality of two rows such that if

$$\max_k |\pi_{ik} - \pi_{jk}| < \epsilon$$

then the two states can be merged. After state merging the corresponding entries in the state probability vector and the $\mathbf{\Pi}$ matrix have to be appropriately modified. Consider a machine where states q_i and q_j are to be merged, then the following steps are required:

1. If $j > i$, remove state q_j and merge with state q_i
2. Set $p_i = p_i + p_j$
3. Delete the j^{th} row of the $\mathbf{\Pi}$ matrix.
4. Set $\pi_{ki} = \pi_{ki} + \pi_{kj} \ \forall \ k = 1, 2.....n$ and delete the j^{th} column of the $\mathbf{\Pi}$ matrix.

As an illustrative example for selection of D and appropriate state reduction, let us consider a data set that yields a symbol stream $\vec{S} = ...000100010001...$ on the alphabet $\mathcal{A} = \{0, 1\}$ [34]. Table 3 provides the number of states after state merging and the corresponding entropy rate of the inferred D-Markov machine for various depths. As seen in Table 3, the number of states in the generated machine remains the same for depth $D \geq 3$. Correspondingly, the entropy rate remains at zero. This implies that the minimum depth for correct representation for this symbol string is 3. The required number of states is less than $|\mathcal{A}|^D$ in this case. Next we consider the case where a small amount of white noise is added to the raw data that produced the symbol stream $\vec{S}$. Table 4 provides the number of states after state merging and the entropy rate of the D-Markov machine for various depths. Although the number of states inferred seem to increase with increasing depth, it can observed that the change in entropy rate h_μ is very small beyond D = 3. This means that very little information is gained by increasing the depth and the uncertainty in the system is largely due to the noise. Hence a criterion for the selection of optimal depth of the D-Markov machine can be established in terms of a lower bound on the change in the entropy rate.

6.5. Stopping Rule for Length of the Symbol Sequence

Another important parameter is the length of symbol sequence required to generate the statistics from the D-Markov machine. This section presents a stopping rule that is necessary to find a lower bound on the length of symbol sequence required for parameter identification of the stochastic matrix $\mathbf{\Pi}$. The stopping rule [37] [38] is based on the properties of irreducible stochastic matrices [39]. The state transition matrix is constructed at

Table 3. Number of states and Entropy Rate for ideal string [34]

Depth (D)	Number of States after state merging	Entropy Rate (h_μ)
0	1	0.810
1	2	0.689
2	3	0.500
3	4	0.000
4	4	0.000
5	4	0.000

Table 4. Number of states and Entropy Rate for noisy string [34]

Depth (D)	No of States after state merging	Entropy Rate (h_μ)
0	1	0.818
1	2	0.721
2	4	0.530
3	6	0.070
4	8	0.050
5	12	0.045

the r^{th} iteration (i.e., from a symbol sequence of length r) as (r) that is an $n \times n$ irreducible stochastic matrix under stationary conditions. Similarly, the state probability vector $\mathbf{p}(r) \equiv [p_1(r)\ p_2(r)\ \cdots\ p_n(r)]$ is obtained as

$$p_i(r) = \frac{r_i}{\sum_{j=1}^{n} r_i} \tag{30}$$

where r_i is the number of symbols in the i^{th} state such that $\sum_{i=1}^{n} r_i = r$ for a symbol sequence of length r. The stopping rule makes use of the Perron-Frobenius Theorem [39] to establish a relation between the vector $\mathbf{p}(r)$ and the matrix (r). Since the matrix (r) is stochastic and irreducible, there exists a unique eigenvalue $\lambda = 1$ and a corresponding left eigenvector $\mathbf{p}(r)$ (normalized to unity in the sense of absolute sum). The (normalized) 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,

$$\mathbf{p}(r) = \mathbf{p}(r)(r) \quad \text{as } r \to \infty \tag{31}$$

Following Eq. (30), the absolute error between successive iterations is obtained such

... ↑↓↓↑↓↑ ... ↑↑↓↑↓ ... ≡ ...100101...11010...

Figure 13. Equivalence between the one dimensional structure of Ising spin system and a symbolic sequence with alphabet size $|\mathcal{A}| = 2$ where $\mathcal{A}$ = {0,1}.

that

$$\| \ (\mathbf{p}(r) - \mathbf{p}(r+1)) \ \|_\infty = \| \ \mathbf{p}(r) \ (\mathbf{I} - (r)) \ \|_\infty \leq \frac{1}{r} \tag{32}$$

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

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

$$\frac{\| \ (\mathbf{p}(r) \ - \ \mathbf{p}(r+1)) \ \|_\infty}{\| \ (\mathbf{p}(r)) \ \|_\infty} \leq \eta \quad \forall \, r \geq r_{stop} \tag{33}$$

The objective is to obtain the least conservative estimate for r_{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 $\frac{1}{n}$, where n is the dimension of $\mathbf{p}(r)$, the best worst case value of the stopping point is obtained from Eqs. (32) and (33) as:

$$r_{stop} \equiv int \left(\frac{n}{\eta} \right) \tag{34}$$

where $int(\bullet)$ is the integer part of the real number $\bullet$.

6.6. Statistical Mechanical Concept of D-Markov Machine

In statistical mechanics, a few macroscopic parameters (e.g. pressure and temperature) are used to describe the global properties of the system in terms of the estimates derived from the distribution of the elementary particles in various micro states [40]. In the same fashion, the behavior of a dynamical system can be investigated both from *microscopic* and *macroscopic* points of view [10]. In the study of a dynamical system, the measured time series data of the observable variables on fast time scale can be analyzed to generate the pattern vectors in terms of probability distributions, which can be used to describe the macroscopic or global behavior of the system at a particular slow time epoch. The information derived from these pattern vectors can be further compressed into a few scalar macroscopic parameters such as the entropy, and the Euclidean norm. This analogy is termed as the thermodynamic formalism of dynamical systems [10].

This section outlines an analogy between the structural features of the D-Markov machine and those of spin models in Statistical Mechanics [6] [41]. The primary concept of symbolic dynamic analysis of a dynamical system is to represent the dynamics with a sequence of symbols. Analogously, one may consider the one dimensional lattice chain of spins in spin models as being a stationary time series of discrete measurements or the symbolic dynamics arising from the partitioning of the phase space of a dynamical system [42]. The Ising model is one of the foundations of statistical mechanics commonly used to describe the magnetic properties of ferromagnetic substances [40] [43]. Each spin s_n in the

Ising model can take only two possible orientations which are represented as $+1$ and -1. It is intuitive that in a physical system spins that are located at neighboring lattice sites strongly influence each other, whereas spins far away from each other do not have much influence on each other [40] [10]. The structure of one dimensional Ising model is analogous to the symbolic sequence generated from partitioning the time series data with alphabet size $|\mathcal{A}| = 2$. The Potts model is a generalization of the Ising model [43] to more than two components [44] [45], i.e., it describes a spin model where each spin s_n can take one of the 'r' different possible spin values $s_k : k \in 1, 2, \cdots, r$. As such, the structure of one dimensional Potts model is structurally analogous to the symbolic sequence generated from partitioning the time series data with alphabet size $|\mathcal{A}| = r > 2$. The analogy is illustrated in Fig. 13 for the simple case of Ising model which is analogous to the symbol sequence with *alphabet* size $|\mathcal{A}| = 2$ where $\mathcal{A}$ = {0,1}.

A symbolic sequence is called D-Markov if the probability of occurrence of a symbol depends on the previous D symbols. The range of interaction between the spins can be considered analogous (to some extent) to depth D of the D-Markov machine. For D=1, the finite-state machine construction is analogous to the one-dimensional spin model with nearest neighbor interactions. For $D \geq 2$, the spin interactions extend beyond the nearest neighbor and represent a higher order Markov process.

7. Pattern Identification and Anomaly Detection

Behavioral pattern changes may take place in dynamical systems due to accumulation of faults and progression of anomalies. The pattern changes are quantified as deviations from the nominal pattern (i.e., the probability distribution at the nominal condition). The resulting anomalies (i.e., deviations of the evolving patterns from the nominal pattern) are characterized by a scalar-valued function, called *Anomaly Measure* (ψ). Several measures can be defined based on the structure of the D-Markov machine. One such measure is the induced norm of the difference between the state transition matrix $\mathbf{\Pi}^k$ at slow time epoch t_k and the nominal state transition matrix $\mathbf{\Pi}^0$

$$\psi^k = ||\mathbf{\Pi}^k - \mathbf{\Pi}^0|| \tag{35}$$

Alternatively, measures of anomaly can be derived directly from the state probability vector $\mathbf{p}$ that is the left eigenvector corresponding to the unique unity eigenvalue of the Π-matrix. The anomaly measure at a slow time epoch t_k is then obtained as:

$$\psi^k \equiv d\left(\mathbf{p}^k, \mathbf{p}^0\right) \tag{36}$$

where $d(\bullet, \bullet)$ is an appropriately defined distance function and $\mathbf{p}^k$ is the state probability vector at the slow time epoch t_k. The distance function can chosen to be a standard norm of the difference between the probability vector at slow time epoch t_k and the probability vector at the nominal condition t_0. For example, a possible choice for anomaly measure is:

$$\psi^k \equiv ||\mathbf{p}^k - \mathbf{p}^0||_r \tag{37}$$

where $||\bullet||_r$ for $r \in [1, \infty)$ is the Hölder norm of $\bullet$, which is the Euclidean norm for $r = 2$. In general, other distance measures can also be chosen because, in a finite-dimensional

vector space, all norms are equivalent [36][46]. Another candidate for the anomaly measure is the angle between the two state probability vectors. This measure is defined as:

$$\psi^k = \arccos\left(\frac{< \mathbf{p}^k, \mathbf{p}^0 >}{||\mathbf{p}^k||_2||\mathbf{p}^0||_2}\right) \tag{38}$$

where $< \mathbf{x}, \mathbf{y} >$ is the inner product between the vectors $\mathbf{x}$ and $\mathbf{y}$. The measures mentioned above, satisfy the requirements of a metric. But other measures, that do not qualify as a metric, for example, the Kullback-Leibler distance [35] may also be used.

$$\psi^k = -\sum_{i=1}^{|\mathcal{A}|} \mathbf{p}_i^k log_2 \frac{\mathbf{p}_i^k}{\mathbf{p}_i^0}. \tag{39}$$

The anomaly measure ψ can also be constructed based on the following information-theoretic quantities: entropy rate, excess entropy, and complexity measure of a symbol string $\mathcal{S}$ (see Appendix A.).

- The entropy rate $h_\mu(\mathcal{S})$ quantifies the intrinsic randomness in the observed dynamical process.
- The excess entropy $\mathbf{E}(\mathcal{S})$ quantifies the memory in the observed process.
- The statistical complexity $\mathcal{C}_\mu(\mathcal{S})$ of the state machine captures the average memory requirements for modelling the complex behavior of a process.

Given two symbol strings $\mathcal{S}$ and $\mathcal{S}_0$, it is possible to obtain a measure of anomaly by adopting any one of the following three alternatives:

$$\mathcal{M}(\mathcal{S}, \mathcal{S}_0)) = \begin{cases} |h_\mu(\mathcal{S}) - h_\mu(\mathcal{S}_0)|, \text{or} \\ |\mathbf{E}(\mathcal{S}) - \mathbf{E}(\mathcal{S}_0)|, \text{or} \\ |C_\mu(\mathcal{S}) - C_\mu(\mathcal{S}_0)| \end{cases}$$

Note that each of the anomaly measures, defined above, is a *pseudo metric* [36]. For example, let us consider two periodic processes with unequal periods, represented by $\mathcal{S}$ and $\mathcal{S}_0$. For both processes, $h_\mu = 0$, so that $\mathcal{M}(\mathcal{S}, \mathcal{S}_0) = 0$ for the first of the above three options, even if $\mathcal{S} \neq \mathcal{S}_0$. It is to be noted that choice of the anomaly measure depends on the application example and sensitivity of change detection.

Having presented the different possible forms of anomaly measure, an example is presented to demonstrate robustness of the state probability vector $\mathbf{p}$ [15]. The vector $\mathbf{p}$ must be robust relative to measurement noise and spurious disturbances and, at the same time, be sensitive enough to detect small slowly-varying anomalies from the observed data set. A distortion measure for the symbol probability vector is introduced below.

$$\delta_t \triangleq ||\mathbf{p}_t - \tilde{\mathbf{p}}_t||_1, \tag{40}$$

where the subscript *t* denotes that the probability vectors correspond to symbols generated from time domain signals; and $|| \bullet ||_1$ is the sum of the absolute values of the elements of the vector $\bullet$. The vector $\mathbf{p}_t$, with $||\mathbf{p}_t||_1 = 1$, corresponds to the uncorrupted signal and

$\tilde{\mathbf{p}}_t$ corresponds to the signal corrupted with additive zero-mean white Gaussian noise with standard deviation σ (Eq. (16)). Similar to Eq. (40), distortion measure in the wavelet scale domain is defined as:

$$\delta_s \triangleq ||\mathbf{p}_s - \tilde{\mathbf{p}}_s||_1, \tag{41}$$

where the subscript *s* denotes that the probability vectors correspond to symbols generated from wavelet scale domain signals (i.e., scale series data). Therefore, lower is the distortion ratio, closer is the probabilistic representation of the corrupted signal to that of the uncorrupted signal, i.e., greater is the robustness to noise and spurious disturbances.

The partitions are obtained, in case of time domain, by employing the maximum entropy criterion on the time series data of the signal. In the wavelet domain, the partitions are obtained with the corresponding scale series data, as defined in Section 4.1.. In both time domain and wavelet domain, the probability vectors $\mathbf{p}$ and $\tilde{\mathbf{p}}$ are computed with the same partitions for the uncorrupted and corrupted signals.

The symbol alphabet size and depth are chosen to be $|\mathcal{A}|$=4 and D=1 respectively. The partitions are obtained as mentioned before for the signal y (Eq. (16)) and its transform, i.e., the coefficient vector $\mathbb{L}y$. Table 5 lists the values of distortion ratios δ_t and δ_s, averaged over 20 simulation runs.

Table 5. Distortion Ratios

	$\sigma = 0.05$	$\sigma = 0.1$
δ_t	0.040	0.054
δ_s	0.006	0.010

It is seen that distortion measures are far smaller in the wavelet scale domain than those in the time domain. This observation implies that the symbol probabilities are significantly more robust to measurement noise and spurious disturbances in the wavelet domain than in the time domain. Hence, it may be inferred that symbols generated from the wavelet coefficients would be better for anomaly detection as the effects of noise to induce errors in the symbol probabilities are significantly mitigated.

8. Summary and Advantages of SDF

The SDF procedure of anomaly detection is summarized below.

8.1. Summary of SDF Procedure

- Acquisition of time series data from appropriate response variable(s) under the nominal condition at time epoch t_0, when the system is assumed to be in the healthy state (i.e., zero anomaly measure)

- Generation of the wavelet transform coefficients, obtained with an appropriate choice of the wavelet basis and range of scales [15]

- Maximum entropy partitioning of the wavelet space at the nominal condition with alphabet size $|\mathcal{A}|$; and generation of the corresponding symbol sequence (Note: The partitioning is fixed for subsequent time epochs.)

- Construction of the D-Markov machine states from the symbol alphabet size $|\mathcal{A}|$ and the window length D (Note: The structure of the finite state machine is also fixed for subsequent slow time epochs.)

- Generation of the pattern vector defined by the state probability vector $\mathbf{p}^0$ by passing the symbol sequence obtained at time epoch t_0 through the finite state machine

- Time series data acquisition at subsequent slow time epochs, $t_1, t_2, ...t_k...$, and their conversion to the wavelet domain to generate respective symbolic sequences based on the partitioning at time epoch t_0

- Generation of the state probability vectors $\mathbf{p}^1, \mathbf{p}^2, ...\mathbf{p}^k, ..$ at different slow time epochs, $t_1, t_2, ...t_k....$ from the respective symbolic sequences using the finite state machine constructed at time epoch t_0

- Computation of scalar anomaly measures $\psi^1, \psi^2, ..., \psi^k, ...$ at different slow-time epochs, $t_1, t_2, ..., t_k, ...$

8.2. Advantages of SDF

After having discussed the underlying principles and essential features of SDF-based pattern recognition and anomaly detection, the major advantages of SDF are listed below:

- *Robustness to measurement noise and spurious signals*[15]- The procedure of SDF is robust to measurement noise and spurious disturbances and it filters out the noise at different steps. First of all, coarse graining of the continuous data (i.e., partitioning into finite blocks) and generation of a symbol sequence eliminate small measurement noise [6]. Secondly, the wavelet transform also contributes in signal-noise separation of the raw time series data by proper choice of scales [15]. Finally, the state probabilities are generated by passing a long symbol sequence over the finite state machine, which further eliminates small (zero-mean) measurement noise;

- Adaptability to low-resolution sensing due to coarse graining in space partitions [6];

- Capability for early detection of anomalies because of sensitivity to signal distortion and real-time execution on commercially available inexpensive platforms [2] [18];

- Applicability to networked communication systems due to the capability of data compression into low-dimensional pattern vectors and error-free transmission over networked systems. Future research is envisioned in the area.

9. Forward and Inverse Problems

As stated earlier in Section 1., the anomaly detection problem is separated into two sub-problems: 1) the *forward (or analysis) problem* and 2) the *inverse (or synthesis) problem*. *The forward problem* consists of prediction of outcomes, given a priori knowledge of the underlying model parameters. In absence of an existing model this problem requires generation of behavioral patterns of the system evolution through off-line analysis of an ensemble of the observed time series data. On the other hand, *the inverse problem* consists of estimation of critical parameters characterizing the system under investigation using the actual observations. Inverse problems arise in different engineering disciplines such as geophysics, structural health monitoring, weather forecasting, and astronomy. Inverse problems often become ill-posed and challenging due to the following reasons: (a) high dimensionality of the parameter space under investigation and (b) in absence of a unique solution where change in multiple parameters can lead to the same observations.

In presence of sources of uncertainties, any parameter inference strategy requires estimation of parameter values and also the associated confidence intervals, or the error bounds, to the estimated values. As such, inverse problems are usually solved using the Bayesian methods that allow observation based inference of parameters and provide a probabilistic description of the uncertainty of inferred quantities. A good discussion of inverse problems is presented by Tarantola [47].

In context of anomaly detection, the tasks and solution steps of these two problems as followed in this chapter are discussed below.

9.0.1. Forward Problem

The primary objective of the forward problem is identification of changes in the behavioral patterns of system dynamics due to evolving anomalies on the slow time scale. Specifically, the forward problem aims at detecting the deviations in the statistical patterns in the time series data, generated at different time epochs in the slow time scale, from the nominal behavior pattern. The solution procedure of the forward problem requires the following steps:

F1. Collection of time series data sets (at fast time scale) from the available sensor(s) at different slow time epochs;

F2. Analysis of these data sets using the SDF method as discussed in earlier sections to generate pattern vectors defined by the probability distributions at the corresponding slow time epochs. The profile of anomaly measure is then obtained from the evolution of this pattern vector from the nominal condition;

F3. Generation of a family of such profiles from multiple experiments performed under identical conditions to construct a statistical pattern of anomaly growth. Such a family represents the uncertainty in the evolution of anomalies in dynamical systems due to its stochastic nature. This step is required in systems where there is a source of parametric or non-parametric uncertainty. For eg., in case of fatigue damage, the uncertainty arises from the random distribution of microstructural flaws in the body of the component leading to a stochastic behavior [48].

9.0.2. Inverse Problem

The objective of the inverse problem is to infer the anomalies and to provide estimates of system parameters from the observed time series data and system response in real time. The decisions are based on the information derived in the forward problem. For eg., in the context of fatigue damage, identical structures operated under identical loading and environmental conditions show different trends in the evolution of fatigue due to surface and sub-surface material uncertainties [20]. Therefore, as a precursor to the solution of the inverse problem, generation of an ensemble of data sets is required during the forward problem for multiple fatigue tests conducted under identical operating conditions. Anomaly estimates can be obtained at any particular instant in a real-time experiment with certain confidence intervals using the information derived from the ensemble of data sets of damage evolution generated in the forward problem [6]. The solution procedure of the inverse problem requires the following steps:

I1. Collection of time series data sets (in the fast time scale) from the available sensor(s) at different slow time epochs up till the current time epoch in a real-time experiment as in step F1 of the forward problem;

I2. Analysis of these data sets using the SDF method to generate pattern vectors defined by probability distributions at the corresponding slow time epochs. The value of anomaly measure at the current time epoch is then calculated from the evolution of this pattern vector from the nominal condition. The procedure is similar to the step F2 of the forward problem. As such, the information available at any particular instant in a real-time experiment is the value of the anomaly measure calculated at that particular instant;

I3. Detection, identification and estimation of an anomaly (if any) based on the computed anomaly measure and the statistical information derived in step F3 of the forward problem.

The family of anomaly measure profiles is analyzed in the inverse problem section to generate the requisite statistical information. In general inverse problem corresponds to pattern identification for estimation of parametric or non-parametric changes based on the knowledge assimilated in the forward problem and the observed time series data of quasi-stationary process response. The information available in real time is the value of the anomaly measure obtained from the analysis of time series data of sensors at any particular time epoch. This information is entered in the inverse problem section that provides the estimates of the useful parameters. The estimates can only be obtained within certain bounds at a particular confidence level. The online statistical information of the damage status is significant because it can facilitate early scheduling for the maintenance or repair of critical components or to prepare an advance itinerary of the damaged parts. The information can also be used to design control policies for damage mitigation and life extension.

10. Application Examples

The concept of *SDF* has been experimentally validated on two laboratory apparatuses for behavioral pattern identification. The first apparatus is an active nonlinear electronic system with a slowly varying dissipation parameter and the second apparatus is a special-purpose computer-controlled fatigue test machine that is instrumented with ultrasonic flaw detectors and an optical travelling microscope.

10.1. Anomaly Detection in Nonlinear Electronic Systems

This example demonstrates efficacy of the symbolic dynamic filtering (SDF) method for anomaly detection in nonlinear systems. Experiments have been conducted on a laboratory apparatus [6] that emulates a second-order non-autonomous, forced Duffing equation in real time [49], modelled as:

$$\frac{d^2y}{dt^2} + \beta\frac{dy}{dt} + y(t) + y^3(t) = A\cos(\Omega t), \tag{42}$$

where the dissipation parameter β varies slowly with respect to the response of the dynamical system; $\beta = 0.1$ represents the nominal condition; and a change in the value of β is considered as an anomaly. With amplitude $A = 22.0$ and $\Omega = 5.0$, a sharp change in the behavior is noticed around $\beta = 0.29$, possibly due to bifurcation. The phase plots, depicting this drastic change behavior, were presented by Ray [6] and are shown in Fig. 14.

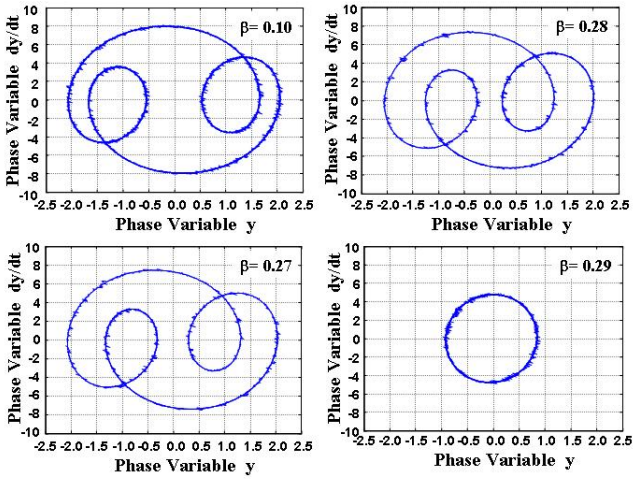

Figure 14. Phase Plots for Electronic Circuit Experiment [6]

The objective of anomaly detection is to identify small changes in the parameter β as early as possible and well before it manifests a drastic change in the system dynamics. The experimental setup is a combination of electronic circuit designed with resistors (R), capacitors (C) and operational amplifiers and a computer interfaced with the circuit. The circuit consists of R-C networks, which model the linear dynamics of the process, adders and voltage amplifiers. The nonlinearity is generated in the computer [34]. The adder sums up the

input signal and the terms generated by the computer, thereby making the overall system nonlinear. Further details of the experimental implementation are provided in [34]. Time series data of the signal $y(t)$ from the experimental apparatus were used for anomaly detection using SDF. Since 'gaus1' [32] matches the shape of the signal, the wavelet 'gaus1' was chosen for SDF. The scale series data (see Section 4. for details), at the nominal condition, was partitioned into a symbol sequence starting with $|\mathcal{A}| = 2$ and the threshold parameter ϵ_h was chosen to be 0.2. It was seen that h monotonically decreases with $|\mathcal{A}|$ and became less than ϵ_h for $|\mathcal{A}|$ = 8. Accordingly, the number of symbols $|\mathcal{A}|$ was chosen to be 8. A smaller value of ϵ_h results in increased number of symbols, which would increase computation with (possibly) no significant gain in accuracy of anomaly detection. The partition was obtained using the maximum entropy principle from the data at the nominal ($\beta = 0.1$) condition. Once the partition is generated, it remains invariant. As the dynamical behavior of the system changes due to variations in β, the statistical characteristics of the symbol sequences are also altered and so do the symbol probabilities. Finite state machine was constructed using D=1 and state probability vectors were generated both under nominal and anomalous conditions. Anomaly measure was chosen as the angle between these vectors, Eq. (38).

Figure 15 depicts the anomaly measure plots obtained using wavelet-based partitioning and phase space partitioning using symbolic false nearest neighbors ($SFNN$) [14]. With β increasing from 0.1, there is a gradual increase in the anomaly measure before the abrupt change in the vicinity of $\beta = 0.29$ takes place. This indicates growth and detection of the anomaly even before a drastic change in the dynamical behavior takes place. It is observed that the results from maximum entropy partitioning with 'gaus1' wavelet are comparable to $SFNN$ partitioning. However, in this problem, the execution time for $SFNN$ to generate the partition is found to be ≈ 4 hours, while that for maximum entropy partitioning is ≈ 100 milliseconds on the same computer. Therefore, it may be inferred from this experiment that maximum entropy partitioning is computationally several orders of magnitude less intensive than $SFNN$ partitioning while they yield similar performance from the perspectives of anomaly detection.

10.2. Detection of Fatigue Damage in Mechanical Systems

This example demonstrates efficacy of the symbolic dynamic filtering (SDF) method for anomaly detection in mechanical systems. Fatigue damage is considered as the source of anomaly. The experimental apparatus, shown in Figure 16, is a special-purpose uniaxial fatigue testing machine, which is operated under load control or strain control at speeds up to 12.5 Hz; a detailed description of the apparatus and its design specifications are reported in [50]. The fatigue tests were conducted using center notched 7075-T6 aluminium specimens, as shown in Fig. 17, at a constant amplitude sinusoidal load, where the maximum and minimum loads were kept constant at 87MPa and 4.85MPa. The specimens used are $3\ mm$ thick and $50\ mm$ wide, and have a slot of $1.58\ mm \times 4.5\ mm$ at the center [41]. The central notch is made to increase the stress concentration factor that ensures crack initiation and propagation at the notch ends. The test apparatus is equipped with two types of sensors that have been primarily used for damage detection:

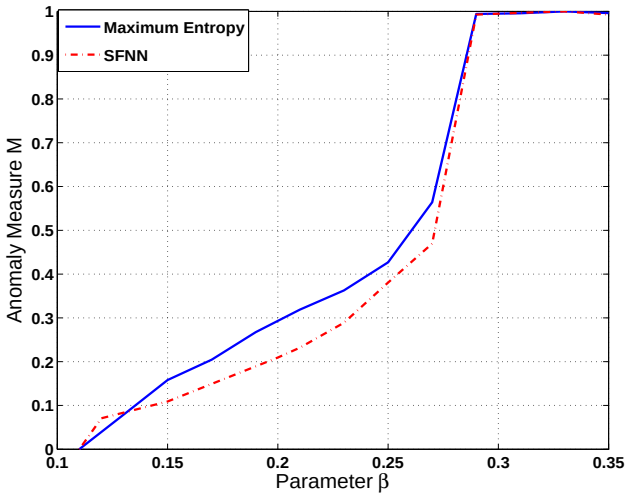

Figure 15. Anomaly Measure Plots

10.2.1. Travelling Optical Microscope

The travelling optical microscope, shown as part of the test apparatus in Figure 16, provides direct measurements of the visible portion of a crack. The resolution of the optical microscope is about 2 microns at a working distance of 10 to 35 cm and the images are taken at a magnification of 75x.

10.2.2. Ultrasonic Flaw Detector

A piezoelectric transducer is used to inject ultrasonic waves in the specimen and an array of receiver transducers is placed on the other side of notch to measure the transmitted signal. The ultrasonic waves produced were 5MHz sine wave signals and they were emitted during a very short portion at the peak of every load cycle. The sender and receiver ultrasonic transducers are placed on two positions, above and below the notch, so as to send the signal through the region of crack propagation and receive it on the other side, as seen in Figure 18. As with the propagation of any wave, it is possible that discontinuities in the propagation media will cause additive and destructive interference. Since material characteristics (e.g., voids, dislocations and short cracks) influence ultrasonic impedance, a small fault in the specimen is likely to change the signature of the signal at the receiver end. Therefore, the received signal can be used to capture minute details and small changes during the early stages of fatigue damage, which are not possible to detect by an optical microscope [51] [52] [53].

The ultrasonic sensing device was triggered at a frequency of 5 MHz at each peak of the ($\sim$12.5 Hz) sinusoidal load. The slow time epochs were chosen to be 3000 load cycles (i.e., $\sim$240 sec) apart. At the onset of each slow time epoch, the ultrasonic data points were collected on the fast time scale of $\sim$8 sec, which produced a string of 10,000 data points. It is assumed that during this fast time scale, no major changes occurred in the fatigue crack

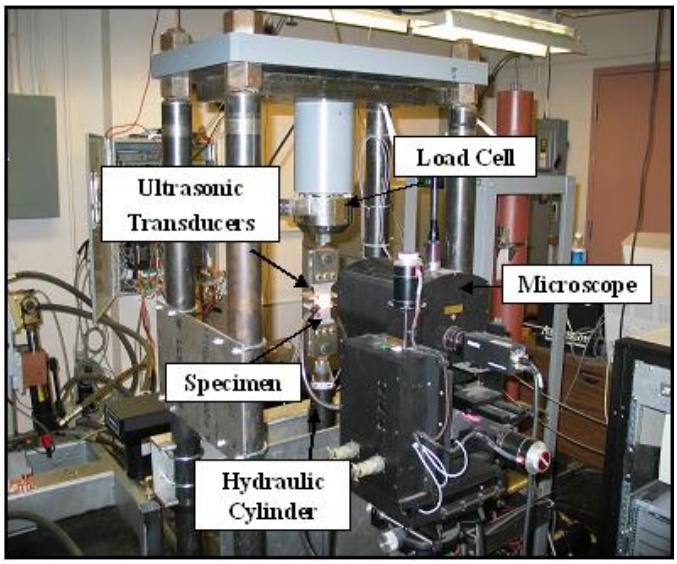

Figure 16. Computer-instrumented Apparatus for Fatigue Testing

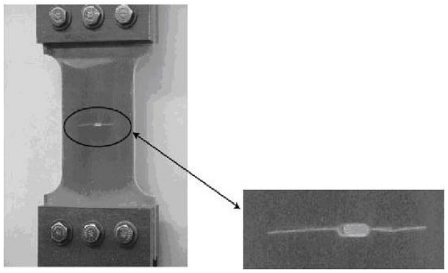

Figure 17. Cracked Specimen with a Central Notch

behavior. The nominal condition at the slow time epoch t_0 was chosen to be 5.0 kilocycles to ensure that the electro-hydraulic system of the test apparatus had come to a steady state and that no significant damage occurred till that point. The anomalies at subsequent slow-time epochs, $t_1, t_2, ...t_k...$, were then calculated with respect to the nominal condition at t_0. Following the SDF procedure for anomaly detection, the alphabet size for partitioning was chosen to be $|\mathcal{A}| = 8$ and window length of $D = 1$, while the mother wavelet chosen to be 'gaus2' [32] because it closely matched the shape of the sinusoidal signals. (Absolute values of the wavelet scale series data were used to generate the partition because of the symmetry of the data sets about their mean.) This combination of parameters was capable of capturing the anomalies earlier than the optical microscope. Increasing the value of $|\mathcal{A}|$ further did not improve the results and increasing the value of D created a large number of states of the finite state machine, many of them having very small or zero probabilities, and required larger number of data points at each time epoch to stabilize the state probability vectors. State probability vector $\mathbf{p}^0$ was obtained at the nominal condition of time epoch t_0

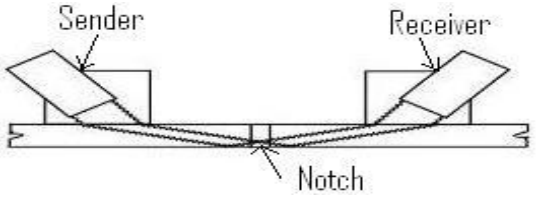

Figure 18. Schematic of Ultrasonic Sensors on a Test specimen

and the state probability vectors $\mathbf{p}^1, \mathbf{p}^2, \ldots \mathbf{p}^k$.... were obtained at other slow-time epochs $t_1, t_2, \ldots t_k$..... It is emphasized that the anomaly measure is relative to the nominal condition which is fixed in advance and should not be confused with the actual damage at an absolute level.

The six triplets of plates in Figure 19 show two-dimensional images of a specimen surface, ultrasonic data and histograms of probability distribution of automaton states at six different time epochs, approximately 5, 30, 40, 45, 60 and 78 kilocycles, exhibiting gradual evolution of fatigue damage [41]. In each triplet of plates from (a) to (f) in Figure 19, the top plate exhibits the surface image of the test specimen as seen by the optical microscope. As exhibited on the top plates, the crack originated and developed on the right side of the notch at the center. Histograms in the bottom plates of six plate triplets in Figure 19 show the evolution of the state probability vector corresponding to fatigue damage growth on the test specimen at different slow time epochs, signifying how the probability distribution gradually changes from uniform distribution (i.e., minimal information) to delta distribution (i.e., maximum information). The middle plates show the ultrasonic time series data collected at corresponding slow time epochs. As seen in Figure 19, the visual inspection of the ultrasonic data does not reveal much information during early stages of fatigue damage but the statistical changes are captured in the corresponding histograms.

The top plate in plate triplet (a) of Figure 19 shows the image at the nominal condition ($\sim$5 kilocycles) when the anomaly measure is taken to be zero, which is considered as the reference point with the available information on potential damage being minimal. This is reflected in the uniform distribution (i.e., maximum entropy) as seen from the histogram at the bottom plate of plate pair (a). Both the top plates in plate triplets (b) and (c) at $\sim$30 and $\sim$40 kilocycles, respectively, do not yet have any indication of surface crack although the corresponding bottom plates do exhibit deviations from the uniform probability distribution. This is an evidence that the analytical measurements, based on ultrasonic sensor data, produce damage information during crack initiation, which is not available from the corresponding optical images.

The top plate in plate triplet (d) of Figure 19 at $\sim$45 kilocycles exhibits the first noticeable appearance of a $\sim$300 micron crack on the specimen surface, which may be considered as the boundary of the crack initiation and propagation phases. This small surface crack indicates that a significant portion of the crack or multiple small cracks might have already developed underneath the surface before they started spreading on the surface. The histogram of probability distribution in the corresponding bottom plate shows further deviation from the uniform distribution at $\sim$5 kilocycles. The top plate in plate triplet (e) of

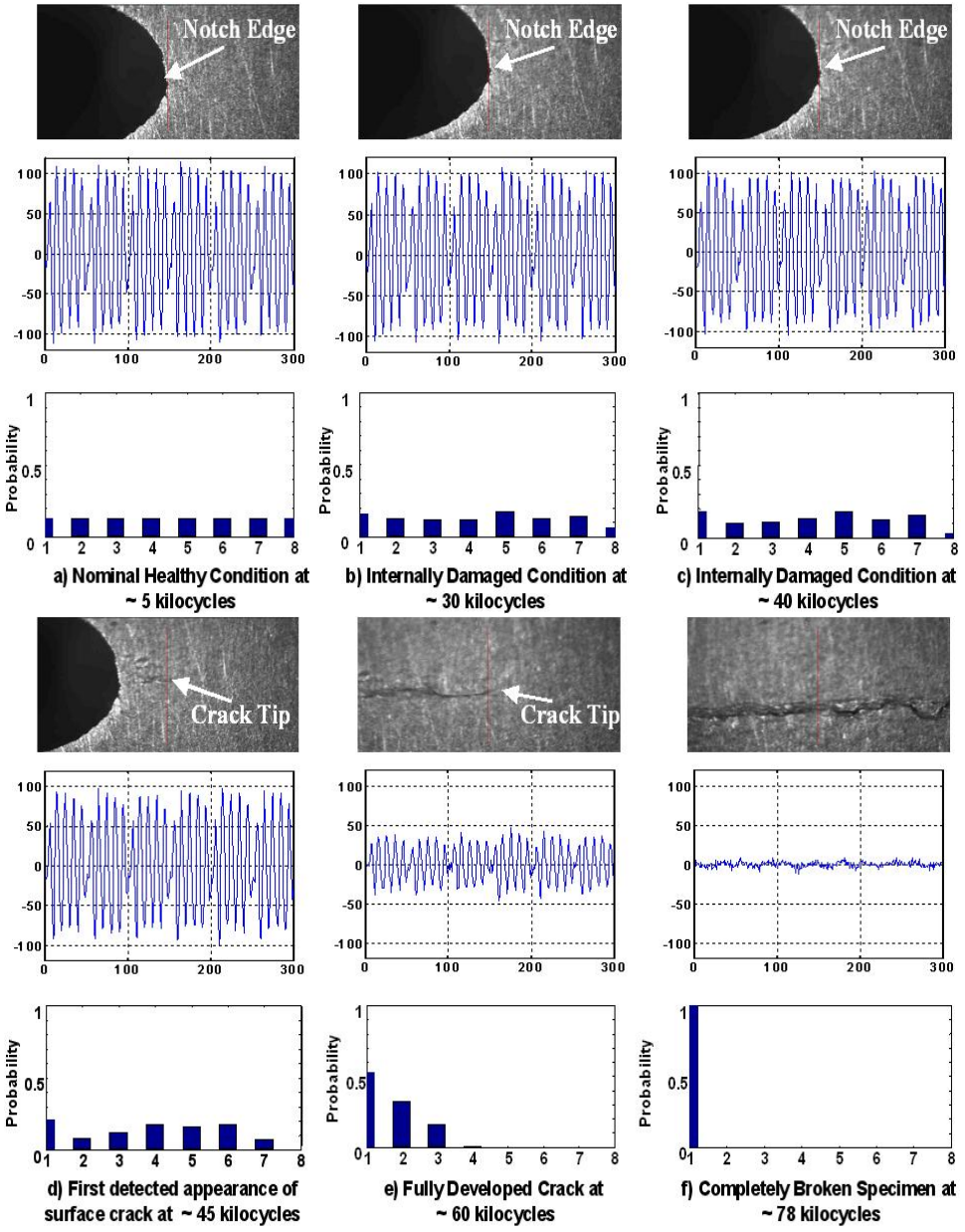

Figure 19. Pictorial view of the evolving fatigue crack damage, corresponding ultrasonic data and histograms of probability distribution [41].

Figure 19 at ∼60 kilocycles exhibits a fully developed crack in its propagation phase. The corresponding bottom plate shows the histogram of the probability distribution that is significantly different from those in earlier cycles in plate triplets (a) to (d), indicating further gain in the information on crack damage. In this case, the middle plate also shows significant drop in the amplitude of ultrasonic signals due to development of a large crack. The top plate in plate triplet (f) of Figure 19 at ∼78 kilocycles exhibits the image of a completely broken specimen. The corresponding bottom plate shows delta distribution indicating complete information on crack damage. The middle plate shows a complete attenuation of the ultrasonic signals.

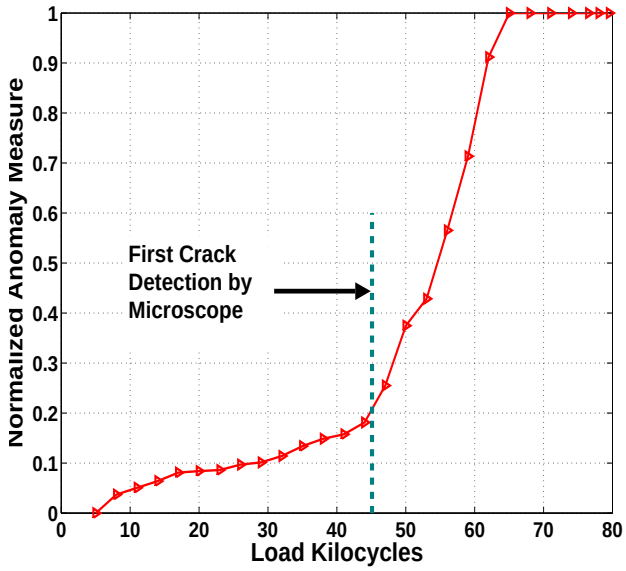

Figure 20. Performance comparison for fatigue damage detection.

The normalized anomaly measure curve in Figure 20 shows a possible bifurcation where the slope of the anomaly measure changes dramatically indicating the onset of crack propagation phase. First appearance of a fatigue crack on the surface of the specimen was detected by the optical microscope at approximately 45 kilocycles, which is marked by the dashed vertical line in Figure 20. The slope of the anomaly measure represents the anomaly growth rate while the magnitude indicates the changes that have occurred relative to the nominal condition. An abrupt change in the slope (i.e., a sharp change in the curvature) of anomaly measure profile provides a clear insight into a forthcoming failure. The critical information lies in the region to the left of the vertical line where no crack was visible on the surface. The slope of anomaly measure curve showed a clear trend of growth of anomaly right after ∼15 kilocycles. This was the region where multiple small cracks were possibly formed inside the specimen, which caused small changes in the ultrasonic signal profile. Fatigue damage detection using SDF of ultrasonic data has been successfully implemented in real time [2].

11. Conclusions

This chapter presents a recently reported technique, called Symbolic Dynamic Filtering (*SDF*), for pattern recognition and anomaly detection in dynamical systems. The underlying concept of *SDF* is built upon the principles of *Symbolic Dynamics*, *Information Theory*, and *Statistical Signal Processing*, where time series data from selected sensor(s) in the *fast* time scale of the process dynamics are analyzed at discrete epochs in the *slow* time scale of anomaly evolution. The chapter describes the underlying features of symbolic dynamics and various aspects of wavelet-based partitioning for extraction of the relevant information from the time series data of observable variables. Efficacy of the wavelet-based partitioning tool has been demonstrated via different examples for noise mitigation and robustness to spurious disturbances. Furthermore, the problem of anomaly detection is constructed into two problems: (i) *Forward problem of Pattern Recognition* for (offline) characterization of the anomalous behavior, relative to the nominal behavior; and (ii) *Inverse problem of Pattern Identification* for (online) estimation of parametric or non-parametric changes based on the knowledge assimilated in the forward problem and the observed time series data of quasi-stationary process response.

The concept of *SDF* has been experimentally validated on two laboratory apparatuses for identification of anomalous patterns. The first apparatus is an active nonlinear electronic system with a slowly varying dissipation parameter and the second apparatus is a special-purpose computer-controlled fatigue test machine that is instrumented with ultrasonic flaw detectors and an optical travelling microscope. Statistical patterns generated from time series data of observed variables have been used to validate the afore-said forward and inverse problems.

Recent literature [2] [18] has also demonstrated experimental validation of SDF-based pattern recognition by comparison with other existing techniques such as Principal Component Analysis (PCA) and Artificial Neural Networks (ANN); SDF has been shown to yield superior performance in terms of early detection of anomalies, robustness to noise [15], and real-time execution in different applications such as electronic circuits [18], mechanical vibration systems [19], and fatigue damage in polycrystalline alloys [2]. The codes of SDF are executable in real time and have been demonstrated in the laboratory environment for on-line detection of fatigue damage, based on the analysis of ultrasonic sensor signals, before any surface cracks are visible through the optical microscope in a special-purpose fatigue testing apparatus.

The work, reported in this chapter, is a step toward building a reliable instrumentation system for early detection of incipient faults and prognosis of potential failures. Further theoretical and experimental research is necessary before its usage in industry. The on-line information, provided by the anomaly measure, is useful for decision and control of human-engineered complex system to sustain order and normalcy under both anticipated and unanticipated faults and disturbances. For example, damage mitigation is an area of future work in life extending control and self healing control. In this context, solution of the inverse problem and development of performance bounds for safe reliable operation of different engineering applications is an active area of current research.

Acknowledgements

The authors acknowledge the assistance and technical contributions of Dr. Rajagopalan in the reported work.

Appendix A. Information Theoretic Quantities

This appendix introduces the concepts of standard information-theoretic quantities: *entropy rate*, *excess entropy* and *statistical complexity* [54], which are used to establish the anomaly measure in Section 7..

Entropy Rate (h_μ): The entropy rate of a symbol string $\mathcal{S}$ is given by the Shannon entropy as follows:

$$h_\mu = \lim_{L \to \infty} \frac{H[L]}{L} \tag{43}$$

where, $H[L] \equiv -\sum_{s^L \in \mathcal{A}^L} P(s^L)\ log_2(P(s^L))$ is the Shannon entropy of all L-blocks (i.e., symbol sequences of length L) in $\mathcal{S}$. The limit is guaranteed to exist for a stationary process [35]. The entropy rate quantifies the irreducible randomness in sequences produced by a source: the randomness that remains after the correlation and the structures in longer and longer sequence blocks are taken into account. For a symbol string $\mathcal{S}$ represented as an ϵ-machine, $h_\mu = H[\overrightarrow{S}^1|\mathcal{S}]$.

Excess Entropy ($\mathbf{E}$): The excess entropy of a symbol string $\mathcal{S}$ is defined as:

$$\mathbf{E} = \sum_{L=1}^{\infty} [h_\mu(L) - h_\mu] \tag{44}$$

where $h_\mu(L) \equiv H[L] - H[L-1]$ is the estimate of how random the source appears if only L-blocks in $\mathcal{S}$ are considered. Excess entropy measures how much additional information must be gained about the sequence in order to reveal the actual per-symbol uncertainty h_μ, and thus measures difficulty in the prediction of the process. Excess entropy has alternate interpretations such as: it is the intrinsic redundancy in the process; geometrically it is a sub-extensive part of $H(L)$; and it represents how much historical information stored in the present is communicated to the future.

Statistical Complexity (C_μ)[54]: The information of the probability distribution of causal states, as measured by Shannon entropy, yields the minimum average amount of memory needed to predict future configurations. This quantity is the *statistical complexity* of a symbol string $\mathcal{S}$, defined by Crutchfield and Young [16] as :

$$C_\mu \equiv H(\mathcal{S}) = -\sum_{k=0}^{n-1} [Pr(S_k)\ log_2 Pr(S_k] \tag{45}$$

where n is the number of states of the finite state machine constructed from the symbol string $\mathcal{S}$. As shown in [54], $\mathbf{E} \leq C_\mu$ in general, and $C_\mu = \mathbf{E} + Dh_\mu$.

Appendix B. D-Markov Machine and Epsilon Machine

This appendix presents a comparison of two alternative techniques of finite-state machine construction from a symbol sequence $\mathcal{S}$. The the D-Markov machine, presented in Section 6.1., is compared with the ϵ-machine formulation [55] for identifying patterns based on time series analysis of the observed data. Both techniques rely on information-theoretic principles (see Appendix A.) and are based on Computational Mechanics [16].

The ϵ-Machine: Like Statistical Mechanics [42][10], Computational Mechanics is concerned with dynamical systems consisting of many partially correlated components. Whereas Statistical Mechanics deals with the local space-time behavior and interactions of the system elements, Computational Mechanics relies on the joint probability distribution of the phase-space trajectories of a dynamical system. The ϵ-machine construction [16] [55] makes use of the joint probability distribution to infer the information processing being performed by the dynamical system. This is developed using the statistical mechanics of orbit ensembles, rather than focusing on the computational complexity of individual orbits.

Let the symbolic representation of a discrete-time, discrete-valued stochastic process be denoted by: $\mathbb{S} \equiv \cdots S_{-2}S_{-1}S_0S_1S_2\cdots$ as defined earlier in Section 6.. At any instant t, this sequence of random variables can be split into a sequence $\overleftarrow{S}_t$ of the past and a sequence $\overrightarrow{S}_t$ of the future. Assuming conditional stationarity of the symbolic process $\mathbb{S}$ (i.e., $P[\overleftarrow{S}_t | \overrightarrow{S}_t = \overrightarrow{s}]$ being independent of t), the subscript t can be dropped to denote the past and future sequences as $\overleftarrow{S}$ and $\overrightarrow{S}$, respectively. A symbol string, made of the first L symbols of $\overrightarrow{S}$, is denoted by $\overrightarrow{S}^L$. Similarly, a symbol string, made of the last L symbols of $\overleftarrow{S}$, is denoted by $\overleftarrow{S}^L$.

Prediction of the future $\overrightarrow{S}$ necessitates determination of its probability conditioned on the past $\overleftarrow{S}$, which requires existence of a function ϵ mapping histories $\overleftarrow{s}$ to predictions $P(\overrightarrow{S}|\overleftarrow{s})$. In essence, a prediction imposes a partition on the set $\overleftarrow{\mathbf{S}}$ of all histories. The cells of this partition contain histories for which the same prediction is made and are called the *effective states* of the process under the given predictor. The set of effective states is denoted by $\mathbf{R}$; a random variable for an effective state is denoted by $\mathcal{R}$ and its realization by ρ.

The objective of ϵ-machine construction is to find a predictor that is an optimal partition of the set $\overleftarrow{\mathbf{S}}$ of histories, which requires invoking two criteria in the theory of Computational Mechanics [17]:

1. *Optimal Prediction*: For any partition of histories or effective states $\mathcal{R}$, the conditional entropy $H[\overrightarrow{S}^L|\mathcal{R}] \geq H[\overrightarrow{S}^L|\overleftarrow{S}]$, $\forall L \in \mathbb{N}$, $\forall \overleftarrow{S} \in \overleftarrow{\mathbf{S}}$, is equivalent to remembering the whole past. Effective states $\mathcal{R}$ are called *prescient* if the equality is attained $\forall L \in \mathbb{N}$. Therefore, optimal prediction needs the effective states to be prescient.

2. *Principle of Occam Razor*: The prescient states with the least complexity are selected, where complexity is defined as the measured Shannon information of the effective states:

$$H[\mathcal{R}] = -\sum_{\rho \in \mathbf{R}} P(\mathcal{R} = \rho) \log P(\mathcal{R} = \rho) \tag{46}$$

Equation (46) measures the amount of past information needed for future prediction and is known as *Statistical Complexity* denoted by $C_\mu(\mathcal{R})$ (see Appendix A.).

For each symbolic process $\mathbb{S}$, there is a unique set of prescient states known as *causal states* that minimize the statistical complexity $C_\mu(\mathcal{R})$.

Definition B.1 *[55] Let $\mathbb{S}$ be a (conditionally) stationary symbolic process and $\overleftarrow{\mathbf{S}}$ be the set of histories. Let a mapping $\epsilon : \overleftarrow{\mathbf{S}} \rightarrow \Upsilon(\overrightarrow{\mathbf{S}})$ from the set $\overleftarrow{\mathbf{S}}$ of histories into a collection $\Upsilon(\overrightarrow{\mathbf{S}})$ of measurable subsets of $\overleftarrow{\mathbf{S}}$ be defined as:*

$$\begin{aligned} &\forall \Gamma \in \Upsilon(\overrightarrow{\mathbf{S}}),\ \ \epsilon(\overleftarrow{s}) \equiv \{\overleftarrow{s}' \in \overleftarrow{\mathbf{S}} \text{ such that} \\ &P(\overrightarrow{S} \in \Gamma | \overleftarrow{S} = \overleftarrow{s}) = P(\overrightarrow{S} \in \Gamma | \overleftarrow{S} = \overleftarrow{s}')\} \end{aligned} \tag{47}$$

Then, the members of the range of the function ϵ are called the causal states of the symbolic process $\mathbb{S}$. The i^{th} causal state is denoted by q_i and the set of all causal states by $\mathbf{Q} \subseteq \Upsilon(\overrightarrow{\mathbf{S}})$. The random variable corresponding to a causal state is denoted by $\mathcal{Q}$ and its realization by q.

Given an initial causal state and the next symbol from the symbolic process, only successor causal states are possible. This is represented by legal transitions among the causal states, and the probabilities of these transitions. Specifically, the probability of transition from state q_i to state q_j on a single symbol s is expressed as:

$$T_{ij}^{(s)} = P\left(\overrightarrow{S}^1 = s, \mathcal{Q}' = q_j |\ \mathcal{Q} = q_i\right)\ \forall q_i, q_j \in \mathbf{Q} \tag{48}$$

$$\sum_{s \in \mathcal{A}} \sum_{q_j \in \mathbf{Q}} T_{ij}^{(s)} = 1 \tag{49}$$

The combination of causal states and transitions is called the *ϵ-machine* (also known as *the causal state model* [55]) of a given symbolic process. Thus, the ϵ-machine represents the way in which the symbolic process stores and transforms information. It also provides a description of the pattern or regularities in the process, in the sense that the pattern is an algebraic structure determined by the causal states and their transitions. The set of labelled transition probabilities can be used to obtain a stochastic matrix [39] given by: $\mathcal{T} = \sum_{s \in \mathcal{A}} \mathcal{T}^s$ where the square matrix $\mathcal{T}^s$ is defined as: $\mathcal{T}^s = [T_{ij}^s]\ \forall s \in \mathcal{A}$. Denoting $\mathbf{p}$ as the left eigenvector of $\mathcal{T}$, corresponding to the eigenvalue $\lambda = 1$, the probability of being in a particular causal state can be obtained by normalizing $\|\mathbf{p}\|_{\ell_1} = 1$. A procedure for construction of the ϵ-machine is outlined below.

The original ϵ-machine construction algorithm is the subtree-merging algorithm as introduced in [16] [17]. This approach has several shortcomings, such as lack of a systematic procedure for choosing the algorithm parameters, may return non-deterministic causal states, and also suffers from slow convergence rates. Recently, Shalizi et al. [55] have developed a code known as Causal State Splitting Reconstruction (*CSSR*) that is based on state splitting instead of state merging as was done in the earlier algorithm of subtree-merging [16]. The *CSSR* algorithm starts with a simple model for the symbolic process and elaborates the model components only when statistically justified. Initially, the algorithm assumes the process to be independent and identically distributed (iid) that can be represented

by a single causal state and hence zero statistical complexity and high entropy rate. At this stage, *CSSR* uses statistical tests to determine when it must add states to the model, which increases the estimated complexity, while lowering the entropy rate h_μ (see Appendix A.). A key and distinguishing feature of the *CSSR* code is that it maintains homogeneity of the causal states and deterministic state-to-state transitions as the model grows. Complexity of the *CSSR* algorithm is: $O(m^{L_{max}}) + O(m^{2L_{max}+1}) + O(N)$, where m is the size of the alphabet $\mathcal{A}$; N is the data size and L_{max} is the length of the longest history to be considered. Details are given in [55].

Comparison of D-Markov Machine and ϵ-Machine: An ϵ-machine seeks to find the patterns in the time series data in the form of a finite-state machine, whose states are chosen for optimal prediction of the symbolic process; and a finite-state automaton can be used as a pattern for prediction [55]. An alternative notion of the pattern is one which can be used to compress the given observation. The first notion of the pattern subsumes the second, because the capability of optimal prediction necessarily leads to the compression as seen in the construction of states by lumping histories together. However, the converse is not true in general. For the purpose of anomaly detection, the second notion of pattern is sufficient because the goal is to represent and detect the deviation of an anomalous behavior from the nominal behavior. This has been the motivating factor for proposing an alternative technique, based on the fixed structure D-Markov machine. It is possible to detect the evolving anomaly, if any, as a change in the probability distribution over the states.

Another distinction between the D-Markov machine and ϵ-machine can be seen in terms of *finite-type shifts* and *sofic shifts* [7] (see Appendix C.). Basic distinction between finite-type shifts and sofic shifts can be characterized in terms of the *memory*: while a finite-type shift has finite-*length* memory, a sofic shift uses finite *amount* of memory in representing the patterns. Hence, finite-type shifts are strictly proper subsets of sofic shifts. While, any finite-type shift has a representation as a graph, sofic shifts can be represented as a *labelled graph*. As a result, the finite-type shift can be considered as an "extreme version" of a D-Markov chain (for an appropriate D) and sofic shifts as an "extreme version" of a Hidden Markov process [56], respectively. The shifts have been referred to as "extreme" in the sense that they specify only a set of allowed sequences of symbols (i.e., symbol sequences that are actually possible, but not the probabilities of these sequences). Note that a Hidden Markov model consists of an internal D-order Markov process that is observed only by a function of its internal-state sequence. This is analogous to sofic shifts that are obtained by a labelling function on the edge of a graph, which otherwise denotes a finite-type shift. Thus, in these terms, an ϵ-machine infers the Hidden Markov Model (sofic shift) for the observed process. In contrast, the D-Markov Model proposed in this paper infers a (finite-type shift) approximation of the (sofic shift) ϵ-machine.

Appendix C. Finite-type Shift and Sofic Shift

This appendix very briefly introduces the concept of shift spaces with emphasis on finite shifts and sofic shifts that respectively characterize the D-Markov machine and the ϵ-machine described in Appendix B.. The shift space formalism is a systematic way to study the properties of the underlying grammar, which represent the behavior of dynamical

systems encoded through symbolic dynamics. The different shift spaces provide increasingly powerful classes of models that can be used to represent the patterns in the dynamical behavior.

Definition C.1 *Let $\mathcal{A}$ be a finite alphabet. The full $\mathcal{A}$-shift is the collection of all bi-infinite sequences of symbols from $\mathcal{A}$ and is denoted by:*

$$\mathcal{A}^Z = \{x = (x_i)_{i \in Z} : x_i \in \mathcal{A} \; \forall i \in Z\} \tag{50}$$

Definition C.2 *The shift map σ on the full shift $\mathcal{A}^Z$ maps a point x to a point $y = \sigma(x)$ whose ith coordinate is $y_i = x_{i+1}$.*

A block is a finite sequence of symbols over $\mathcal{A}$. Let $x \in \mathcal{A}^Z$ and w be a block over $\mathcal{A}$. Then w occurs in x if $\exists$ indices i and j such that $w = x_{[i,j]} = x_i x_{i+1} \cdots x_j$. Note that the empty block ϵ occurs in every x.

Let $\mathcal{F}$ be a collection of blocks, i.e., finite sequences of symbols over $\mathcal{A}$. Let $x \in \mathcal{A}^Z$ and w be a block over $\mathcal{A}$. Then w occurs in x if $\exists$ indices i and j such that $w = x_{[i,j]} = x_i x_{i+1} \cdots x_j$. For any such $\mathcal{F}$, let us define $X_{\mathcal{F}}$ to be the subset of sequences in $\mathcal{A}^Z$, which do not contain any block in $\mathcal{F}$.

Definition C.3 *A shift space is a subset X of a full shift $\mathcal{A}^Z$ such that $X = X_{\mathcal{F}}$ for some collection $\mathcal{F}$ of forbidden blocks over $\mathcal{A}$.*

For a given shift space, the collection $\mathcal{F}$ is at most countable (i.e., finite or countably infinite) and is non-unique (i.e., there may be many such $\mathcal{F}$'s describing the shift space). As subshifts of full shifts, these spaces share a common feature called *shift invariance*. Since the constraints on points are given in terms of forbidden blocks alone and do not involve the coordinate at which a block might be forbidden, it follows that if $x \in X_{\mathcal{F}}$, then so are its shifts $\sigma(x)$ and $\sigma^{-1}(x)$. Therefore $\sigma(X_{\mathcal{F}}) = X_{\mathcal{F}}$, which is a necessary condition for a subset of $\mathcal{A}^Z$ to be a shift space. This property introduces the concept of shift dynamical systems.

Definition C.4 *Let X be a shift space and $\sigma_X : X \to X$ be the shift map. Then (X, σ_X) is known as a shift dynamical system.*

The shift dynamical system mirrors the dynamics of the original dynamical system from which it is generated (by symbolic dynamics). Several examples of shift spaces are given in [7].

Rather than describing a shift space by specifying the forbidden blocks, it can also be specified by allowed blocks. This leads to the notion of a *language* of a shift.

Definition C.5 *Let X be a subset of a full shift, and let $\mathcal{B}_n(X)$ denote the set of all n-blocks (i.e., blocks of length n) that occur in X. The langauge of the shift space X is defined as:*

$$\mathcal{B}(X) = \bigcup_{n=0}^{\infty} \mathcal{B}_n(X) \tag{51}$$

Sliding Block Codes: Let X be a shift space over $\mathcal{A}$, then $x \in X$ can be transformed into a new sequence $y = \cdots y_{-1}y_0y_1 \cdots$ over another alphabet $\mathcal{U}$ as follows. Fix integers m and n such that $-m \leq n$. To compute y_i of the transformed sequence, we use a function Φ that depends on the "window" of coordinates of x from $i - m$ to $i + n$. Here $\Phi : \mathcal{B}_{m+n+1}(X) \to \mathcal{U}$ is a fixed *block map*, called a $(m + n + 1)$-*block map* from the allowed $(m + n + 1)$-blocks in X to symbols in $\mathcal{U}$. Therefore,

$$y_i = \Phi(x_{i-m}x_{i-m+1} \cdots x_{i+n}) = \Phi(x_{[i-m,i+n]}) \tag{52}$$

Definition C.6 *Let Φ be a block map as defined in Eq. (52). Then the map $\phi : X \to (\mathcal{U})^Z$ defined by $y = \phi(x)$ with y_i given by Eq. (52) is called the sliding block code with memory m and anticipation n induced by Φ.*

Definition C.7 *Let X and Y be shift spaces, and $\phi : X \to Y$ be a sliding block code.*

- *If $\phi : X \to Y$ is onto, then ϕ is called a factor code from X onto Y.*
- *If $\phi : X \to Y$ is one-to-one, then ϕ is called an embedding of X into Y.*
- *If $\phi : X \to Y$ has a inverse (i.e., $\exists$ a sliding block code $\psi : Y \to X$ such that $\psi(\phi(x)) = x \ \forall x \in X$ and $\phi(\psi(y)) = y \, \forall \, y \in Y$), then ϕ is called a conjugacy from X to Y.*

If $\exists$ a conjugacy from X to Y, then Y can be viewed as a copy of X, sharing all properties of X. Therefore, a conjugacy is often called topological conjugacy in literature.

Finite-type Shifts: We now introduce the concept of finite-type shift that is the structure of the shift space in the D-Markov machine proposed in the subsection 6.1..

Definition C.8 *A finite-type shift is a shift space that can be described by a finite collection of forbidden blocks (i.e., X having the form $X_{\mathcal{F}}$ for some finite set $\mathcal{F}$ of blocks).*

An example of a finite shift is the *golden mean shift*, where the alphabet is $\mathcal{A} = \{0, 1\}$ and the forbidden set $\mathcal{F}$=$\{11\}$. That is, $X = X_{\mathcal{F}}$ is the set of all binary sequences with no two consecutive 1's.

Definition C.9 *A finite-type shift is M-step or has memory M if it can be described by a collection of forbidden blocks all of which have length $M + 1$.*

The properties of a finite-type shift are listed below:

- If X is a finite-type shift, then $\exists\ M \geq 0$ such that X is M-step.
- The langauge of the finite-type shift is characterized by the property that if two words overlap, then they can be glued together along their overlap to from another word in the langauge. Thus, a shift space X is an M-step finite-type shift iff whenever uv, $vw \in \mathcal{B}(X)$ and $|v| \geq M$, then $uvw \in \mathcal{B}(X)$.
- A shift space that is conjugate to a finite-type shift is itself a finite-type shift.
- A finite-type shift can be represented by a finite, directed graph and produces the collection of all bi-infinite walks (i.e. sequence of edges) on the graph.

Sofic Shifts: The sofic shift is the structure of the shift space in the ϵ-machines [16] [55]. Let us label the edges of a graph with symbols from an alphabet $\mathcal{A}$, where two or more edges are allowed to have the same label. Every bi-infinite walk on the graph yields a point in $\mathcal{A}^Z$ by reading the labels of its edges, and the set of all such points is called a *sofic shift*.

Definition C.10 *A graph G consists of a finite set $\mathcal{V} = \mathcal{V}(G)$ of vertices together with a finite set $\mathcal{E} = \mathcal{E}(G)$ of edges. Each edge $e \in \mathcal{E}(G)$ starts at a vertex denoted by $i(e) \in \mathcal{V}(G)$ and terminates at a vertex $t(e) \in \mathcal{V}(G)$ (which can be the same as $i(e)$). There may be more that one edge between a given initial state and terminal state; a set of such edges is called a set of multiple edges. An edge e with $i(e) = t(e)$ is called a self-loop.*

Definition C.11 *A labelled graph $\mathcal{G}$ is a pair $(G, \mathcal{L})$, where G is a graph with edge set $\mathcal{E}$, and $\mathcal{L} : \mathcal{E} \rightarrow \mathcal{A}$ assigns a label $\mathcal{L}(e)$ to each edge e of G from the finite alphabet $\mathcal{A}$. The underlying graph of $\mathcal{G}$ is G.*

Definition C.12 *A subset X of a full shift is a sofic shift if $X = X_{\mathcal{G}}$ for some labelled graph $\mathcal{G}$. A presentation of a sofic shift X is a labelled graph $\mathcal{G}$ for which $X_{\mathcal{G}} = X$.*

An example of a sofic shift is the *even shift*, which is the set of all binary sequences with only even number of 0's between any two 1's. That is, the forbidden set $\mathcal{F}$ is the collection $\{10^{2n+1} : n \geq 0\}$.

Some of the salient characterization of sofic shifts are presented below [7]:

- Every finite-type shift qualifies as a sofic shift.
- A shift space is sofic iff it is a factor of a finite-type shift.
- The class of sofic shifts is the smallest collection of shifts spaces that contains all finite-type shifts and also contains all factors of each space in the collection.
- A sofic shift that does not have finite-type subshifts is called a *strictly sofic*. For example, the *even shift* is strictly sofic [7].

- A factor of a sofic shift is a sofic shift.
- A shift space conjugate to a sofic shift is itself sofic.
- A distinction between finite-type shifts and sofic shifts can be characterized in terms of the *memory*. While finite-type shifts use finite-*length* memory, sofic shifts require finite *amount* of memory. In contrast, context-free shifts require infinite amount of memory [8].

References

[1] R. Badii and A. Politi, *Complexity hierarchical structures and scaling in physics*. Cambridge University Press, United Kingdom, 1997.

[2] S. Gupta, A. Ray, and E. Keller, "Symbolic time series analysis of ultrasonic data for early detection of fatigue damage," *Mechanical Systems and Signal Processing*, vol. 21, no. 2, pp. 866–884, 2007.

[3] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[4] C. Andrieu, A. Doucet, S. Singh, and V. B. Tadic, "Particle methods for change detection, system identification, and control," *Proceedings IEEE*, vol. 92, no. 3, pp. 423–438, 2004.

[5] S. Ozekici, *Reliability and Maintenance of Complex Systems*, vol. 154. NATO Advanced Science Institutes (ASI) Series F: Computer and Systems Sciences, Berlin, Germany, 1996.

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

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

[8] H. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation, 2nd ed.* Addison Wesley, Boston, 2001.

[9] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. John Wiley & Sons Inc., 2001.

[10] C. Beck and F. Schlögl, *Thermodynamics of chaotic systems: an introduction*. Cambridge University Press, United Kingdom, 1993.

[11] C. S. Daw, C. E. A. Finney, and E. R. Tracy, "A review of symbolic analysis of experimental data," *Review of Scientific Instruments*, vol. 74, no. 2, pp. 915–930, 2003.

[12] H. D. I. Abarbanel, *The Analysis of Observed Chaotic Data*. Springer-Verlag, New York, 1996.

[13] R. L. Davidchack, Y. C. Lai, E. M. Bolt, and H. Dhamala, "Estimating generating partitions of chaotic systems by unstable periodic orbits," *Physical Review E*, vol. 61, pp. 1353–1356, 2000.

[14] M. B. Kennel and M. Buhl, "Estimating good discrete partitions form observed data: Symbolic false nearest neighbors," *Physical Review E*, vol. 91, no. 8, pp. 084–102, 2003.

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

[16] J. P. Crutchfield and K. Young, "Inferring statistical complexity," *Physical Review Letters*, vol. 63, pp. 105–108, 1989.

[17] J. P. Crutchfield, "The calculi of emergence: Computation dynamics and induction," *Physica*, vol. D, no. 75, pp. 11–54, 1994.

[18] S. Chin, A. Ray, and V. Rajagopalan, "Symbolic time series analysis for anomaly detection: A comparative evaluation," no. 9, pp. 1859–1868, September 2005.

[19] A. Khatkhate, *Anomaly Detection in Electromechanical Systems using Symbolic Dynamics*. PhD thesis, Department of Mechanical Engineering, Pennsylvania State University, State College, PA, 2006.

[20] S. Gupta and A. Ray, "Real-time fatigue life estimation in mechanical systems," *Measurement Science and Technology*, vol. 18, no. 7, pp. 1947–1957, 2007.

[21] F. Takens, "Detecting strange attractors in turbulence," *Proceedings of the Symposion Dynamical Systems and Turbulence, D. Rand, L.S. Young (Eds.), Warwick, 1980, Lecture Notes in Mathematical, Vol. no 898, Springer, Berlin*, p. 366, 1981.

[22] E. Ott., *Chaos in Dynamical Systems*. Cambridge University Press, 1993.

[23] H. Zhang, A. Ray, and S. Phoha, "Hybrid life extending control of mechanical systems: Experimental validation of the concept," *Automatica*, vol. 36, no. 1, pp. 23–36, 2000.

[24] H. Kantz and T. Schreiber, *Nonlinear Time Series Analysis, 2nd ed.* Cambridge University Press, United Kingdom, 2004.

[25] C.J.Veenman, M.J.T.Reinders, E.M.Bolt, and E.Baker, "A maximum variance cluster algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1273–1280, 2002.

[26] T.Chau and A.K.C.Wong, "Pattern discovery by residual analysis and recursive partitioning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, no. 6, pp. 833–852, 1999.

[27] Y. Kakizawa, R. Shumway, and N. Taniguchi, "Discrimination and clustering for multivariate time series," *J. Amer. Stat. Assoc.*, vol. 93, no. 441, pp. 328–340, 1999.

[28] T. Liao, "Clustering of time series data - a survey," *Pattern Recognition*, vol. 38, pp. 1857–1874, 2005.

[29] S. Mallat, *A Wavelet Tour of Signal Processing 2/e*. Academic Press, 1998.

[30] A. Teolis, *Computational Signal Processing with Wavelets*. Birkhäuser, Boston, MA, 1998.

[31] G.Kaiser, "A friendly guide to wavelets," *Birkhäuser, Boston, MA*, 1994.

[32] *MATLAB Wavelet Toolbox*. Mathworks Inc, 2006.

[33] P. Abry, "Ondelettes et turbulence, multirésolutions, algorithmes de décomposition, invariance déchelles," *Diderot Editeur, Paris*, 1997.

[34] V. Rajagopalan, *Symbolic Dynamic Filtering of Complex Dynamical Systems*. PhD thesis, Department of Electrical Engineering, Pennsylvania State University, State College, PA, 2007.

[35] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley, New York, 1991.

[36] A. W. Naylor and G. R. Sell, *Linear Operator Theory in Engineering and Science*. Springer-Verlag, New York, 1982.

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

[38] S. Gupta, A. Ray, and A. Mukhopadhyay, "Anomaly detection in thermal pulse combustors using symbolic time series analysis," *Proceedings of the Institution of Mechanical Engineers- Part I- Journal of Systems and Control Engineering*, vol. 220, no. 5, pp. 137–146, 2006.

[39] R. B. Bapat and T. E. S. Raghavan, *Nonnegative Matrices and Applications*. Cambridge University Press, 1997.

[40] R. K. Pathria, *Statistical Mechanics*. Elsevier Science and Technology Books, 1996.

[41] S. Gupta, *Behavioral Pattern Identification for Structural Health Monitoring in Complex Systems*. PhD thesis, Department of Mechanical Engineering, Pennsylvania State University, State College, PA, 2006.

[42] D. P. Feldman, *Computational Mechanics of Classical Spin Systems*. PhD thesis, Department of Physics, University of California Davis, 1998.

[43] E. Ising *Z. Phys.*, vol. 21, p. 613, 1925.

[44] R. B. Potts, "Some generalized order - disorder transformations," *Proc. Cambridge Phil. Soc.*, vol. 48, pp. 106–109, 1952.

[45] P. Martin, *Potts models and related problems in Statistical Mechanics*. World Scientific, 1991.

[46] W. Rudin, *Real and Complex Analysis, 3rd ed.* McGraw Hill, New York, 1988.

[47] A. Tarantola, *Inverse Problem Theory*. Society for Industrial and Applied Mathematics, 2005.

[48] K. Sobczyk and B. F. Spencer, *Random Fatigue: Data to Theory*. Academic Press, Boston, MA, 1992.

[49] J.M.T.Thompson and H.B.Stewart, "Nonlinear dynamics and chaos," *John Wiley, Chichester, United Kingdom*, 1986.

[50] E. E. Keller, *Real time sensing of fatigue crack damage for information-based decision and control*. PhD thesis, Department of Mechanical Engineering, Pennsylvania State University, State College, PA, 2001.

[51] E. E. Keller and A. Ray, "Real time health monitoring of mechanical structures," *Structural Health Monitoring*, vol. 2(3), pp. 191–203, 2003.

[52] S. I. Rokhlin and J. Y. Kim, "In situ ultrasonic monitoring of surface fatigue crack initiation and growth from surface cavity," *International journal of fatigue*, vol. 25, pp. 41–49, 2003.

[53] J. L. Rose, *Ultrasonic waves in solid media*. Cambridge university press, 2004.

[54] D. P. Feldman and J. P. Crutchfield, *Discovering Non-critical Organization: Statistical Mechanical, Information Theoretic, and Computational Views of Patterns in One-Dimensional Spin Systems*. Santa Fe Institute (SFI) Working Paper 98-04-026, 1998.

[55] C. R. Shalizi, K. Shalizi, and J. Crutchfield, *An Algorithm for Pattern Discovery in Time Series*. Santa Fe Institute (SFI) Working Paper 02-10-060, 2002.

[56] D. Upper, *Theory and Algorithms for Hidden Markov Models and Generalized Hidden Markov Models*. PhD Dissertation in Mathematics, University of california, Berkeley, 1997.