2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Symbolic Identification and Anomaly Detection in Complex Dynamical Systems

Subhadeep Chakraborty Soumik Sarkar Asok Ray
szc138@psu.edu szs200@psu.edu axr2@psu.edu

The Pennsylvania State University
University Park, PA 16802

***Abstract*— Symbolic dynamic filtering (*SDF*) has been reported in recent literature for early detection of anomalies (i.e., deviations from the nominal behavior) in complex dynamical systems. In this context, instead of solely relying on physics-based modeling that may be difficult to formulate and validate, this paper proposes data-driven modeling and system identification based on the concept of *Symbolic Dynamics*, *Automata Theory*, and *Information Theory*. For anomaly detection in inter-connected complex dynamical systems, with or without closed loop control, the input excitation to an individual component is likely to deviate from the nominal condition as a result of deterioration of some other component(s) or to accommodate disturbance rejection by feedback control actions. This paper presents a formal-language-based syntactic method of anomaly detection to account for deviations in the pertinent input excitation. A training algorithm is formulated to generate an automaton model of the underlying subsystem or component from a set of input-output combinations for different classes of inputs, where the objective is to detect (possibly gradually evolving) anomalies under different input conditions. The proposed method has been validated on a test apparatus of nonlinear active electronics.**

Index Terms*—*Anomaly Detection, Symbolic Dynamics, System Identification, Fixed Structure Automata

I. Introduction

Nonlinearity, chaos and unpredictability are regularly observed in both natural and human-engineered systems, many of which are exceedingly complex, involving inherently nonlinear electronic, electromechanical, thermal and chemical processes, complicated interconnections, and elaborate control systems. Health monitoring of these complex engineering systems has evolved to be an issue of paramount importance. However, the inherent nonlinearity and uncertainty in these complex systems pose a challenging problem to health monitoring, since first principle models of these systems, if available, are routinely oversimplified, or in worst cases may not be available at all. In the absence of models, system identification or 'black box' modeling with the help of input-output pair combination has gained in importance over the years. A branch of this system identification science has lead to development of Nonlinear Time Series Analysis (NTSA) techniques using Formal Languages [1].

Recent research has extensively explored the problem of anomaly detection using symbolic dynamic filtering (*SDF*) [2]–[5]. However, since human-engineered multi-component systems are usually interconnected physically as well as through the use of feedback control loops, the effect of any one component degradation may affect the input streams to the remaining components. The major challenges here are detection and isolation of faults in simultaneously degrading components and estimation of the fault magnitude, for the purpose of prognoses without a high-fidelity component-level model of the system. Therefore, for systems that are composed of many interconnected components, system identification turns out to be of significant importance especially.

There are several nonlinear system identification techniques available for such applications; an example is artificial neural networks (*ANN*). However, system identification and anomaly detection in a single component is just a small part of the health monitoring problem in its entirety. In the setting of the bigger problem, complex algorithms and optimization techniques may have certain drawbacks. For example, computationally expensive algorithms are unsuitable for large complex engineering systems such as an aircraft where on-board health monitoring is needed in real time. Future generation health monitoring systems are envisioned to utilize low power mobile computing devices to physically access the local sensors in a network environment, perform the component level health analysis on-board, and communicate with the central decision making console for higher level decisions [6]. In such large-scale remote applications, communication over wireless sensor networks and, as a consequence, dimensionality reduction of the data sets is essential.

The above discussion evinces that, for the purpose of efficient health monitoring of complex interconnected systems, a system-identification tool is necessary. The purpose of the work reported in this paper is to address this issue, and develop a robust and computationally inexpensive system identification technique based on formal language formulation, which achieves the above-mentioned objectives. A central step in this kind of identification methodology is discretization of the raw time-series measurements into a corresponding sequence of symbols. An important practical advantage of working with symbols is increased computational efficiency [2], [3]. The proposed method is designed to be robust with respect to sensor noise, and also simple enough to be implemented in mobile platforms or even embedded in the sensors themselves. Thus, it facilitates construction

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 Grant No. NNX07AK49A.

of a reliable sensor network to serve as a backbone to the higher levels in the decision-making hierarchy of large-scale complex systems.

II. Formal-language-based System Identification

Formal language theory has been used in the past to study nonlinear dynamical systems [7]–[10].In general, grammatical complexity of sequences generated according to some coding have been used to characterize complexity of autonomous dynamical systems. However, keeping in mind that most systems in technological applications are not autonomous but controlled dynamical systems, distinction has to be made between the information that is generated by the system dynamics and the one that depends on the controlling operator. Borrowing the nomenclature from Narendra and Thathachar [11] the external influence on automata may be described as the‘'environment',which interacts with the automata. Martins et al [12] have shown how the modeling framework for controlled dynamical systems leads, almost uniquely, to a context-dependent grammatical formulation.

A. Problem Statement

Grammatical inference is an inductive inference problem where the target domain is a formal language and the representation class is a family of grammars. The learning task is to identify a 'correct' grammar for the unknown target language, given a finite number of examples of the language. To apply grammatical inference procedures to identification of non-autonomous dynamical systems, a dynamical system must be considered as an entity (linguistic source) capable of generating a specific language. The grammar $\mathcal{G}$ of the language is the set of rules that specifies all the words in the language and their relationships. In context of health monitoring of complex systems, the aim of the Grammatical Inference technique is to develop a grammatical description of a dynamical system from the input/output characteristics, in such a way, that it should be invariant with the input conditions, but should be sensitive to changes in the parameters of the actual dynamical system.

Let $\boldsymbol{D}_i$ be a dynamic system indexed by i representing different parametric conditions, $\boldsymbol{D}_0$ being the nominal or healthy condition of the system, and $i = 1, 2, ...$ signifying deteriorating health conditions of the plant due to a progressing anomaly. Let $\boldsymbol{U}_k \quad k = 1, 2, ..., \mathcal{K}$ be $\mathcal{K}$ different input conditions, $\boldsymbol{y}_k^i$ be the output from the $i-th$ system $\boldsymbol{D}_i$ receiving the $k-th$ input $\boldsymbol{U}_k$. Let $\mathcal{G}_i$ be the grammatical representation of $\boldsymbol{D}_i$. Then

$$\mathcal{G}_i : \boldsymbol{U}_k \rightarrow \boldsymbol{y}_k^i \quad \forall k \in \{1, 2, ..., \mathcal{K}\} \tag{1}$$

Let $d(\cdot, \cdot)$ be a distance function defined on the space of all possible grammars, then the requirement is

$$d(\mathcal{G}_i, \mathcal{G}_0) \geq d(\mathcal{G}_j, \mathcal{G}_0) \quad i \geq j \tag{2}$$

Instead of defining the distance d on the space of grammars directly it may be possible to use the actual output from the dynamical system and define a distance on the space of the words (symbol sequences) produced by the grammars. The decision regarding the health of the plant will be based on the distance of the actual word output of the plant $\boldsymbol{D}_i$ and the hypothetical output of $\mathcal{G}_0$, both receiving $\boldsymbol{U}_k$. So, an equivalent way of posing the problem is, if y_i^k is the actual output of $\boldsymbol{D}_i$ receiving input k, then find a grammar $\mathcal{G}_0$, such that,

$$d(y_i^k, \hat{y}_0^k) \geq d(y_j^k, \hat{y}_0^k) \quad i \geq j, \forall k \tag{3}$$

$$d(y_0^k, \hat{y}_0^k) = 0 \quad \forall k \tag{4}$$

where $\hat{y}_0^k$ is the output from $\mathcal{G}_0$ receiving input k. The proposed solution is by using a *Fixed Structure Automata* which is described in detail in the next section.

B. Theoretical Background of the Methodology

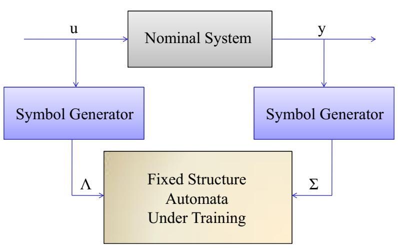

Fig. 1. Training Scheme

Formally, the *Fixed Structure Automata* can be defined as a quintuple

$$\mathcal{G} = \{\mathcal{Q}, \Lambda, \Sigma, \delta, \gamma\} \tag{5}$$

where,

1) $\mathcal{Q}$ is the finite set of states of the automaton, i.e. $\mathcal{Q} = \{q_1, q_2, ..., q_f\}$.
2) $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_m\}$, is the set of input alphabets. The input symbols to the *Fixed Structure Automata* have a one-to-one correspondence to the quantized values of input to the dynamical system.
3) $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_n\}$ is the set of output alphabets, where the output symbols are one-to-one with the quantized values of output from the dynamical system.
4) $\delta : \mathcal{Q} \times \Lambda \rightarrow \mathcal{Q}$ is the state transition function which maps the current state into the next state on receiving the input λ. The transition function can also be stochastic in which case,
$$\delta : \mathcal{Q} \times \Lambda \rightarrow Pr\{\mathcal{Q}\} \tag{6}$$
5) $\gamma : \mathcal{Q} \rightarrow \Sigma$ is the output generation function which determines the output symbol from the current state. In its full generality, γ can be stochastic as well, i.e.
$$\gamma : \mathcal{Q} \rightarrow Pr\{\Sigma\} \tag{7}$$

Time series sensor data (possibly multi-dimensional) are obtained from the input and output data streams of the dynamical system $\mathcal{D}_0$ under nominal condition. Let $\mathcal{U} = \{u_1, u_2, ...\}$ denote the discretized input data sequence, where $u_k \in \Lambda$. Similarly let $\mathcal{Y} = \{y_1, y_2, ...\}, \quad y_k \in \Sigma$ denote the discretized output sequence. A D-Markov

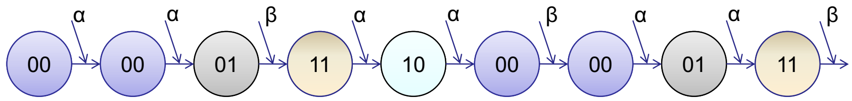

Fig. 2. State Progression for the example Problem

machine is next constructed, with states defined by symbol blocks of length D from $\mathcal{Y}$. There are many interesting issues associated with this process of construction of a D-Markov machine from time series data. Different methods of partitioning, the effects of the depth D on the performance of the machine, issues regarding stopping rule for appropriate data length, and various pre-processing techniques have been studied in details and have been reported in recent literature. The reader is referred to references [2] and [3] for an in-depth description of the procedure.

The transition function used in the current methodology has been designed to be stochastic. $\delta : \mathcal{Q} \times \Lambda \rightarrow Pr\{\mathcal{Q}\}$ gives the probability distribution of transition from state q_i to $\{q_1, q_2, .., q_f\}$ on receiving an input λ_j. A grammar constructed in this way has the advantage over the context sensitive grammar described in [12] in that, the number of production rules may become inconveniently large in case of a context sensitive grammar.

However, the function $\gamma : Q \rightarrow \Sigma$, which maps the current state q_i to the current output symbol σ_i is completely deterministic. This is really an artifact of the state construction procedure [2]. For example, if the depth of the D-Markov process is selected as 2, then $q_n = y_n y_{n-1}$. States, constructed in this manner has two interesting aspects. The first, as mentioned, is that it leads to a very natural way of selecting the output generating function γ. Therefore, effectively the Fixed Structure Automata can be considered as a quadruple instead of a quintuple. Another more important aspect is, this structure is very similar to the classical structure of dynamical systems and thus quite intuitive. For example for a 2^{nd} order system at an instant n the next output y_{n+1} depends on the current and the past output y_n and y_{n-1} respectively and on the current input u_n.

$$y_{n+1} = f(y_n, y_{n-1}, u_n) \tag{8}$$

This corresponds to the *Fixed Structure Automata* of depth 2 where the state transition can be written as

$$\sigma_{n+1}\sigma_n = \delta(\sigma_n\sigma_{n-1}, \lambda_n) \tag{9}$$

This completes the structure of the automaton which is trained to model a component of a complex system.

C. Training Scheme

The training scheme shown in Fig. 1 explains the identification of the state transition function δ from the input-output symbol sequences obtained from experiment on the system while it is under nominal condition.

It is assumed that inputs and outputs are time-synchronized. The state transition function δ can be expanded into two dimensional matrices δ^{λ_i}, indexed by the input variable alphabets. That means

$$\delta = \{\delta^{\lambda_1}, \delta^{\lambda_2}, ..., \delta^{\lambda_m}\} \tag{10}$$

where $\delta^{\lambda_i} : q_j \times \lambda_i \rightarrow Pr\{\mathcal{Q}\}$ maps the current state and input to the probability distribution over all possible states. The algorithm for estimating the matrices δ^{λ_i} is straightforward and is illustrated next with a simple example. Suppose, a dynamical system is characterized as following:

$$\begin{aligned} \Lambda = \{\alpha, \beta\} \quad \Sigma = \{0, 1\} \\ \mathcal{U} = \alpha\alpha\beta\alpha\alpha\beta\alpha\alpha\beta... \\ \mathcal{Y} = 000110001100011... \end{aligned} \tag{11}$$

With depth $D = 2$

$$\mathcal{Q} = \{00, 01, 10, 11\} \tag{12}$$

The state progression is shown in Fig. 2. Hence in this case, $\delta = \{\delta^\alpha, \delta^\beta\}$, where the state transition matrices corresponding to α and β are given in Tables I and II respectively.

TABLE I
STATE TRANSITION PROBABILITIES FOR δ^α

	00	01	10	11
00	0.5	0.5	0	0
01	0	0	0	1
10	1	0	0	0
11	0	0	1	0

TABLE II
STATE TRANSITION PROBABILITIES FOR δ^β

	00	01	10	11
00	0.5	0.5	0	0
01	0	0	0	1
10	1	0	0	0
11	0	0	1	0

The training algorithm has to make sure that the probability values of δ^i converge. The convergence depends on the length of the input-output symbol sequences. The next step of training algorithm is to generate the grammar $\mathcal{G}^k$ for different input-output combinations. For highly nonlinear systems it goes without saying, that the response of the dynamical system to input sequences from the entire range of possible input sequences cannot be covered by a single representative grammar. Instead, the space of possible input sequences has to be partitioned into a finite number of disjoint equivalence classes, such that, their union spans the entire range of possible inputs, i.e. if the set of all possible input sequences is denoted by $\boldsymbol{\mathcal{U}}$, and the different equivalence classes are denoted by $\Delta\mathcal{U}^i$, then

$$\boldsymbol{\mathcal{U}} = \bigcup_{i=1}^{R} \Delta\mathcal{U}^i \tag{13}$$

In the training phase, it has to be ensured that the grammar $\mathcal{G}$ is trained with sufficient input data belonging to a particular equivalence class. This is the so-called *coverage* *problem*. Suppose the grammar is trained with $\mathbf{K}$ input data sets $\{\boldsymbol{U}_1, \boldsymbol{U}_2, ..., \boldsymbol{U}_{\mathbf{K}}\}$ where $\boldsymbol{U}_k \in \Delta\mathcal{U}^j \quad \forall k = 1, ..., \mathbf{K}$ and for some $j \in \{1, ..., R\}$. Let the identified grammar in each case be denoted as $\mathcal{G}^k$. Then the problem reduces to finding a grammar $\mathcal{G} = \mathcal{F}\left(\mathcal{G}^k\right)$ which is optimal with respect to the actual input being fed to the system.

D. Anomaly Detection Scheme

Figure 3 gives a schematic representation of the anomaly detection philosophy being adopted for the present problem. Input and output time series data from the actual plant is discretized to form symbol sequences and is fed to the trained fixed structure automaton. The discretization should be performed using the same partitioning as was done during the training phase.

Before describing the actual anomaly detection methodology, it should be noted that, the FSA uses the output from the actual system in addition to the input, and hence cannot serve as an independent "system identification" procedure in the classical sense of the term. The automata can serve as a system emulator only when the state transition function δ is fully deterministic, i.e. given the current state q_j and the current input symbol λ_i,

$$\delta^{\lambda_i}(q_j, q_k) = 1 \qquad \text{for one and only one } k \tag{14}$$

It can be shown that by a proper redefinition of partitioning and depth used for the construction of states, any stochastic automaton can be converted to a deterministic finite state automata [11]. But that transformation inevitably leads to state explosion and uneconomical growth in the computational complexity.

Instead, in the current scheme, the state transition probability vectors $\pi_{q_j}^{\lambda_i}$, which are the rows of the state transition matrix δ, serve as feature vectors, and are used for the purpose of anomaly detection. An extremely convenient feature of using state transition probabilities as feature vectors, and using stochastic methods to define distances between nominal and off-nominal behavior of plants is that this technique is very robust to noise. Also phase differences, which might be an important issue in case of point by point comparison, are easily handled with this technique, since steady state long term behavior is considered for forming the probability vectors.

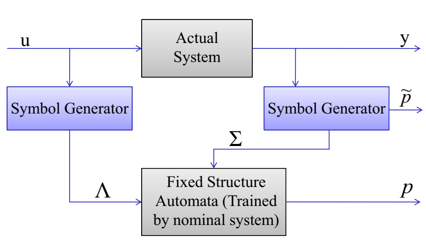

Fig. 3. General Anomaly Detection Scheme

In the present study, a novel intuitive **Pseudo-Learning Technique** of utilizing the stochastic state transition function δ is proposed, for the purpose of Anomaly Detection. In this method, the actual state transitions inside the fixed-structure automaton during the anomaly detection phase occur according to the output symbol sequence obtained from the actual system, but at each instance of state transition, *i.e.* at every instant the trained automaton produces a *State Transition Probability vector* π_n [11], which is characteristic of the nominal system.

For example, let the dynamic system described in Eq. (11) depart from its nominal condition, and produce an output given by

$$\mathcal{Y} = 111001110011100... \tag{15}$$

So, the output states are as shown in Fig. 4. The output probability vectors π_n are the instantaneous predicted state transition probabilities of the nominal system, also shown in the figure.

It may be noted, that this technique conforms to the classical notion about dynamical systems because of the method of construction of the states. For a discrete time 2^{nd} order system it can be written that

$$y_n = f(y_{n-1}, y_{n-2}, u_n) \tag{16}$$

where y and u are output and input respectively. In the present scenario, since the current state contains information about the output in the last two instants, the pattern vector π_n, produced by the trained automaton, is characteristic of the nominal behavior of the plant given the past history of input and output. The current (possibly off-nominal) condition of the plant is characterized by another state probability vector $\tilde{\pi}_n$. This is defined for the actual system output at an instant n, for which only one element of the vector will be 1, rest are zeros. The next step is to use the sequences of instantaneous State Probability vectors $\{\boldsymbol{\pi}_n\}$ and $\{\tilde{\boldsymbol{\pi}}_n\}$ obtained at each time instant, to construct an anomaly measure. Under the assumption of ergodicity of the system, a pattern can be generated from frequency count of the state visits over a wide time window in case of symbolic time series analysis [2]. The equivalent process in the present case would be calculation of mean State Probability vectors $\boldsymbol{p}$ and $\tilde{\boldsymbol{p}}$ from the collections $\{\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, ..., \boldsymbol{\pi}_n\}$ and $\{\tilde{\boldsymbol{\pi}}_1, \tilde{\boldsymbol{\pi}}_2, ..., \tilde{\boldsymbol{\pi}}_n\}$ respectively over time instants 1,2,...,n. A suitable distance function $d(\cdot, \cdot)$ is chosen for measuring the distance between the vectors $\boldsymbol{p}$ and $\tilde{\boldsymbol{p}}$. Anomaly measure μ is defined as the distance

$$\mu = d(\boldsymbol{p}, \tilde{\boldsymbol{p}})$$

In the *Learning Automata* literature, *learning* [11] is done by continuous feedback from environment to the automaton at each time instant. Here also similar feedback technique is taken but not for learning or changing the structure or internal functions of the finite state machine, but only to provide actual history of past outputs to the nominal automaton based model. Thus the technique can be called a **Pseudo-Learning Technique**.

III. Active Electronic System Test Apparatus

The proposed concept of anomaly detection is validated on a test apparatus of an active electronic system [13] that

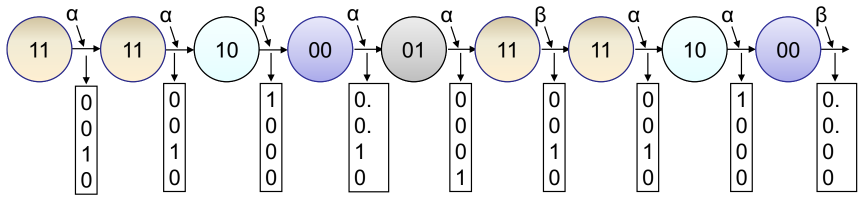

Fig. 4. Anomaly Detection for the example Problem

simulates a second-order forced differential equation, namely the Duffing equation with cubic nonlinearity.

$$\frac{d^2x(t)}{dt^2} + \beta\frac{dx}{dt} + x(t) + x^3(t) = A\cos(\omega t) \tag{17}$$

The dissipation parameter β is the slowly varying parameter in this experiment. $\beta = 0.1$ represents the nominal condition. The parameter β was slowly varied over time from 0.1 to 0.29 by increments of 0.02. 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. Figure 5 depicts the phase plots for select values of β for different forcing conditions, which will be explained in detail in the next subsection. One of the objectives, apart from estimation of parameter β, is to detect the onset of this change in behavior as early as possible without largely getting affected by the different forcing conditions.

A. Sensitivity to Input Conditions

The Duffing equation, being an externally stimulated non-linear differential equation is extremely well suited to study the effects of input variation on the system trajectory. In order to study the relative effect of input and system parameter variation, the Duffing Equation is excited by a series of sinusoidals. Figure 5 illustrates the phase space trajectory for three different input frequencies, namely $\omega = 1.33, 1.67$ and 5.0. For each value of ω, the trajectories corresponding to three values of β are displayed, namely $\beta = 1.0, 1.77$ and 3.5. It may be observed that changing the input frequency, even by a slight amount, (from 1.33 to 1.67) brings about a radical change in the phase space trajectory. Hence, it is imperative to partition the input space into equivalent classes, (for example, into different ranges of frequency) and train the pseudo-identification algorithm for each class of inputs. The challenging task is to give an accurate estimate of the parameter β when the input varies within any particular equivalence class.

IV. Results and Discussion

The first step in the forward problem is selection of the wavelet basis. Wavelet 'db1' [14] is chosen as the wavelet basis. The next step is selection of scales. Following the procedure described in [3], the scales are selected to be 370.99, and 125.61. The wavelet coefficients are obtained at these scales and stacked to form the scale series. The alphabet size is chosen to be $|\Sigma| = 8$. The maximum entropy partition is obtained with scale series data corresponding to the nominal condition at $\beta = 0.1$. The scale series is then converted to symbols based on the partition. With the selection of Depth $D = 1$, the number of states in the D-Markov machine becomes $|\Sigma|^D = 8$.

Using the above partition, symbol sequences are generated from all other scale series data sets. The anomaly detection procedure described in section II-B is applied next. The angle measure, the 1-norm and the 2-norm are calculated. The profile of the angle measure, the 1-Norm and the 2-norm for one experiment, is shown in Fig. 6.

Next, a series of $\mathbf{K}$ data is generated for the purpose of training by providing the nominal system a series of input with different frequency contents. However, it is ensured that all the training inputs are within the same equivalence class for which the automata is being trained. After the training is completed, a set of grammars $\mathcal{G}^k$, is obtained, where $k \in \{1, ..., \mathbf{K}\}$. For the purpose of anomaly detection the last step is to formulate a new grammar $\mathcal{G} = \mathcal{F}(\mathcal{G}^k)$. The method for finding the optimum grammar is still under investigation and has been reported in the future work section. At present, results are provided with

$$\mathcal{G} = \frac{1}{\mathcal{K}}\sum_{k=1}^{\mathcal{K}}\mathcal{G}^k \tag{18}$$

The system is excited with sinusoidal inputs with frequencies between 4.9 and 5.1 radians in steps of 0.5 radians. The trained automata is then used for anomaly detection when the

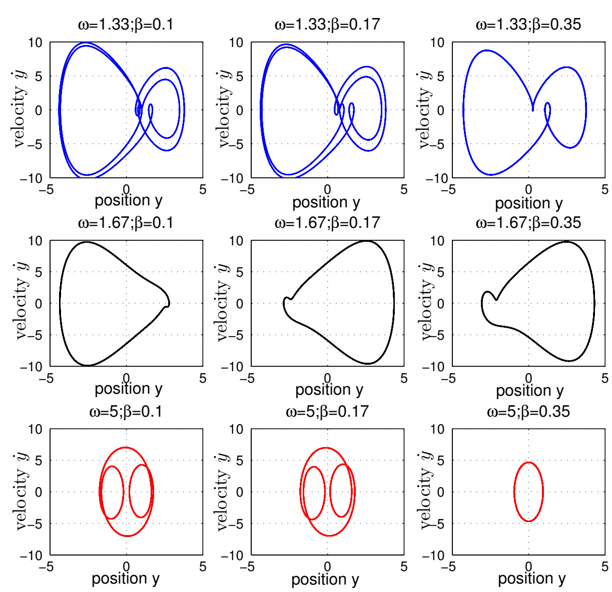

Fig. 5. Duffing System Phase Plots

system is supplied with a sinusoidal input having frequency $\omega = 5.05$ radians. As expected (see Fig. 7), there is some 'residual anomaly', in the sense that the plots indicate some anomaly even when the system is healthy, that is $\beta = 0.1$.

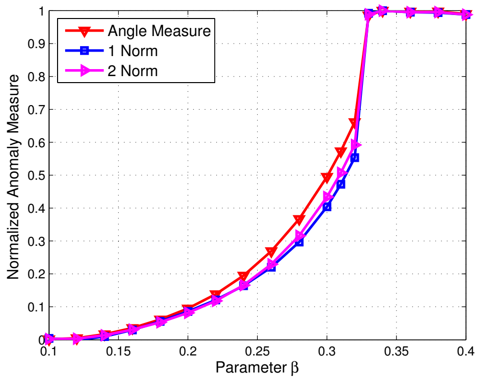

Fig. 6. A Typical Plot of Deviation Measure for the Duffing System

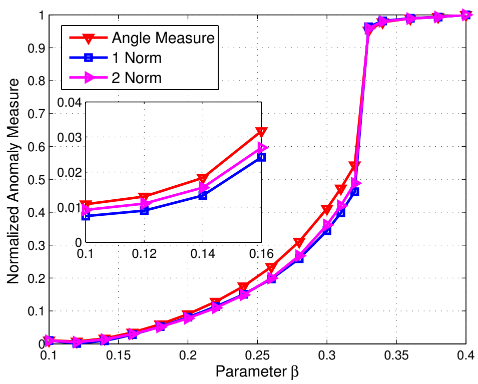

Fig. 7. Plots of Deviation Measure for the Duffing System when excited by an unknown input

The remedy for this problem is to calculate the anomaly measures predicted by the calculated grammar $\mathcal{G}$ by running it on the nominal plant output against different training inputs and subtract the upper bound of this "residue anomaly" from the actual anomaly measure obtained during monitoring the system. This provides a conservative estimate of anomaly in the system, which may be used for health prediction.

V. Summary, Conclusions and Future Work

In this paper, some of the critical and practical issues regarding the problem of health monitoring of multi-component human-engineered systems have been discussed, and a syntactic method has been proposed. The two primary features of this proposed concept are: (i) *Symbolic identification* and (ii) *Pseudo-learning technique*.

The reported work is a step toward building a real-time data-driven tool for estimation of parametric conditions in nonlinear dynamical systems. There are many potential applications of this tool, such as real time anomaly detection and early prognosis of failures in human-engineered systems. However, further theoretical and experimental research is necessary before its application in industry. For example, efficacy of an health monitoring technique can be measured by:

- Quality of the anomaly measure curve,
- Ability to capture the features of a nominal system at all points in an equivalence class of inputs,
- Ability to detect and predict significant changes, when the system is even slightly off-nominal.

All these attributes depend, to a great extent, on the method of partitioning the space of time series data. In the present work, a marginal maximum entropy partitioning [15] for multi-dimensional sensor information has been applied. Here, the shape of the cells were chosen as hypercubes, and the data sets were partitioned along each axis separately, using the principle of maximum entropy.

While there are many other research issues that need to be addressed, the following research topics are being currently pursued.

- Development of a multi-dimensional partitioning for a MIMO system, which should be computationally inexpensive.
- Constructing an optimal grammar $\mathcal{G}$ from the set of grammars obtained in the identification part of the problem. The optimization should be capable of resolving the issue of *Residual anomaly*.

References

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

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

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

[4] A. Khatkhate, A. Ray, E. Keller, S. Gupta, and S. Chin, "Symbolic time series analysis for anomaly detection in mechanical systems," *IEEE/ASME Trans. on Mechatronics*, vol. 11, no. 4, pp. 439–447, August 2006.

[5] 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.

[6] S. Phoha, T. LaPorta, and e. C. Griffin, *Sensor Network Operations*. IEEE Press, Piscataway, NJ, 2006.

[7] J.P.Crutchfield and K.Young, "Computation at the onset of chaos," *Complexity, Entropy and Physics of Information, SFI Studies in the Sciences of Complexity,W.H. Zurek, Ed. Reading, MA: Addison-Wesley*, vol. 8, pp. 223–267, 1990.

[8] D.Auerbach and I.Procaccia, "Grammatical complexity of strange sets," *Phys. Rev.*, vol. A41, p. 66026614, 1990.

[9] G.Troll, "Formal languages in dynamical systems," *Acta Univ. Carolina Math. Phys*, vol. 34, p. 117, 1993.

[10] C.Moore, "Dynamical recognizers: Real-time language recognition by analog computers," *Santa Fe Inst., Santa Fe, NM, working paper*, 96-05-023.

[11] K.S.Narendra and M.A.L.Thathachar, *Learning Automata An Introduction*. Prentice-Hall, 1989.

[12] J.F.Martins, J.A.Dente, A.J.Pires, and R.V.Mendes, "Language identification of controlled systems: Modeling, control, and anomaly detection," *IEEE Transactions On Systems, Man, And CyberneticsPart C: Applications And Reviews*, May 2007.

[13] V. Rajagopalan, A. Ray, R. Samsi, and J. Mayer, "Pattern identification in dynamical systems via symbolic time series analysis," *Pattern Recognition*, vol. 40, no. 11, pp. 2897–2907, 2007.

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

[15] T.Chau, "Marginal maximum entropy partitioning yields asymptotically consistent probability density functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, 2001.