Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Data Driven Anomaly detection via Symbolic Identification of Complex Dynamical Systems

Subhadeep Chakraborty Eric Keller Asok Ray
szc138@psu.edu eekeller@psu.edu axr2@psu.edu

The Pennsylvania State University
University Park, PA 16802

***Abstract*—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 method involves abstraction of a qualitative description from a general dynamical system structure, using state space embedding of the output data-stream and discretization of the resultant pseudo state and input spaces. The system identification is achieved through grammatical inference techniques, and the deviation of the plant output from the nominal estimated language gives a measure of anomaly in the system. The technique is validated on an experimental test-bed of a permanent magnet synchronous motor undergoing a gradual degradation of the encoder orientation feedback.**

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

I. Introduction

Most of the natural and human-engineered systems of practical interest 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 complexity 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], [3], [4], [5]. But, apparently the system identification aspect of the health monitoring problem has not received much attention for systems that are composed of many smaller components. 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. Also, in most practical situations, the system might need to operate in different operating regimes and thus under diverse input conditions. Variations in operating conditions may result from exogenous factors, for instance, varying load requirements in a power plant or extreme maneuvering necessities of a fighter jet. The major challenges here are detection and isolation of faults and estimation of the fault magnitude, for the purpose of prognoses, without a high-fidelity component-level model of the system.

There are several 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. Moreover, in 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, an on-board real-time 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 embedded in the sensors themselves. Thus, it facilitates construction of a reliable sensor network to serve as a backbone to higher levels in the decision-making hierarchy of large-scale complex systems.

The method of real-time diagnostic and fault estimation technique proposed in this paper has been validated on an experimental apparatus consisting of a Permanent Magnet Synchronous Motor ($PMSM$) drive. This set-up is particularly relevant for testing condition-monitoring algorithms, since $PMSM$s are being increasingly used in all commercial and non-commercial applications. For example, both commercial and fighter aircrafts use $PMSM$s as critical components, and an early indication about their performance deterioration is significantly valuable.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

II. Problem Description

Grammatical inference is an inductive inference problem where the target domain is a formal language and the representation class is a family of grammars. 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. Almost all physical process can be truthfully represented by a dynamical system operating in continuous time and in continuous state space. It is evident that estimation of deviation of such a system from its nominal operating condition with grammatical inference techniques need to be decomposed into three main tasks:

A. Abstraction

The first task is abstracting a discrete qualitative counterpart of the general dynamical system representing the physical process, such that the system output of this abstracted description constitutes a unique language in the setting of the formal language (Chomsky) hierarchy.

B. Identification

The learning task is to identify a 'correct' grammar for the unknown target language, given a finite number of examples of the language. 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.

C. Comparison

Once the inference/identification phase is complete, the output of the abstracted system can be compared to the actual output in the sense of some suitably defined metric for estimation of degradation of the system under investigation.

The next section develops these ideas in a systematic manner, with reference to a Permanent Magnet Synchronous Motor ($PMSM$) system undergoing a slow degradation.

III. Theoretical Background of the Methodology

Definition The underlying structure of a dynamical system, can be represented by a General Dynamical System (*GDS*), defined as an 8-tuple (see [6] for details)

$$\boldsymbol{D} = (T, U, W, Q, P, f, g, \leq) \tag{1}$$

where

- T is a time set,
- U and W are input and output sets respectively,
- Q are inner states,
- P denotes the input process $P : T \to X$,
- f denotes the global state transition

$$f : T \times T \times Q \times P \to Q \quad \text{time-varying system} \tag{2}$$
$$f : T \times Q \times P \to Q \quad \text{time-invariant} \tag{3}$$

- g denotes the output function

$$g : T \times Q \to W \quad \text{time-varying system} \tag{4}$$
$$g : Q \to W \quad \text{time-invariant} \tag{5}$$

Let $\boldsymbol{D}_i$ be a dynamical 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. From now onwards, any component of the GDS $\boldsymbol{D}_i$ will be denoted by the corresponding symbol with a subscript denoting the health condition of that system. For example, f_i denotes the global state transition function for a system in the i-th health state. 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 $\mathcal{G}$ be the grammatical representation (also called a Qualitative Dynamical System (*QDS*)) of the nominal plant $\boldsymbol{D}_0$.

Definition The quantized abstraction of the *GDS* called a Qualitative Dynamical System (*QDS*) can be represented as a 5-tuple

$$\mathcal{G} = \{\mathcal{Q}, \Lambda, \Sigma, \delta, \gamma\} \tag{6}$$

where,

- $\mathcal{Q}$ is the finite set of qualitative states of the automaton, i.e. $\mathcal{Q} = \{q_1, q_2, ..., q_f\}$.
- $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_m\}$, is the set of qualitative input events.
- $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_n\}$ is the set of output alphabets, where the output symbols are one-to-one with the quantized values of output from the dynamical system.
- $\delta : \mathcal{Q} \times \Lambda \to \mathcal{Q}$ 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,

$$\delta : \mathcal{Q} \times \Lambda \to Pr\{\mathcal{Q}\} \tag{7}$$

- $\gamma : \mathcal{Q} \to \Sigma$ 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.

$$\gamma : \mathcal{Q} \to Pr\{\Sigma\} \tag{8}$$

Let $\boldsymbol{\chi}$ denote a set of qualitative abstraction functions

$$\boldsymbol{\chi} : \boldsymbol{D}_0 \to \mathcal{G} \tag{9}$$

It may be noted that $\boldsymbol{\chi}$ is a 3-tuple, consisting of three individual abstraction functions.

$$\chi = (\chi_{TQX}, \chi_Q, \chi_W), \quad \text{where}$$

$$\chi_{TQX} : T \times Q \times X \to \Lambda \tag{10}$$
$$\chi_Q : Q \to \mathcal{Q} \tag{11}$$
$$\chi_W : W \to \Sigma \tag{12}$$

Kokar [6] introduced a set of necessary and sufficient conditions, or 'consistency postulates' that the pair $\mathcal{G}, \boldsymbol{\chi}$ must satisfy in order to be a valid representation of the general dynamical system. However, if the purpose of discretization is ultimately deviation measurement or health monitoring, the consistency postulates devised in [6] can be modified as follows:

Definition Let $\mathcal{D}$,$\mathcal{G}$ and $\boldsymbol{\chi}$ represent a GDS, QDS and an abstraction function. Then the pair $(\mathcal{G}, \boldsymbol{\chi})$ form a consistent representation in a probabilistic sense if, $\forall q, x, t$,

$$\gamma(\chi_Q(q)) = \chi_W(g(q)) \tag{13}$$

$$d_i\left(E\left[\delta\left(\chi_Q(q), \chi_{TQX}(t,q,x)\right)\right], E\left[\chi_Q\left(f_i(t,q,x)\right)\right]\right) \approx 0, \quad \text{for} \quad i=0 \tag{14}$$

$$d_j \geq d_i \quad \text{(monotonicity)} \tag{15}$$

where d is a suitably defined distance between probability measures and j denotes a more deteriorated state than i.

Note that since $\delta : \mathcal{Q} \times \Lambda \rightarrow Pr\{\mathcal{Q}\}$, the left hand side $E\left[\delta\left(\chi_Q(q), \chi_{TQX}(t,q,x)\right)\right]$ denotes the estimated probability distribution of occurrence of the qualitative output alphabet in a sufficiently long operation. Since the distance is defined between expected quantities the consistency requirement has been diluted to mean consistency in a weaker sense.

A. Abstraction

Theorem 3.1: Let $W_\pi = W_1, ..., W_n$ be a finite partition of a GDS's output space W, given by $\chi_W^{-1} : \Sigma \rightarrow W_\pi$. Let Q_π describe a partition of Q defined as an inverse image of W_π through g,

$$Q_\pi = g^{-1}(W_\pi),$$

and let TQX_π describe a partition of $T \times Q \times X$ defined as an inverse image of Q_π through f,

$$TQX_\pi = f^{-1}Q_\pi.$$

Then Q_π is a maximal admissible partition of Q, and TQX_π is a maximal admissible partition of $T \times Q \times X$ [6].
In effect,

- a critical hypersurface of partition in Q is an image of the partition in W through g^{-1}, and
- a critical hypersurface partitioning $T \times Q \times X$ is an image of the partition in Q through f^{-1}.

If the system model, i.e the equations governing the general dynamical system is known, the critical hypersurfaces or partitions can be analytically evaluated and utilized as delineated in the preceding section.

However, in the absence of model equations, this scheme is of little practical use, unless

1) there is an alternate means of constructing the phase space purely from output, without using the model equations,
2) there is an alternate means of arriving at the proposed partition without information about the state transition function f and the output function g.

The next two subsections delineate a method for achieving these ends in an approximate way.

1) Phase Space Construction: Starting from the output signal captured by suitable instrumentation, a pseudo phase space can be constructed from delay vectors using Taken's theorem [7]. The embedded phase space can be denoted by

$$\boldsymbol{x}(k) = \left[x_{k-\tau}, ..., x_{k-m\tau}\right],$$

where τ is the time lag, and m is the embedding dimension. In order to find optimum values of the embedding parameters m and τ, the literature reports many optimization routines. In this case, following [8] the KL estimate of the differential entropy

$$H(x) = \Sigma_{j=1}^{N} \ln\left(N\rho_j\right) + ln2 + c_E \tag{16}$$

is first calculated, where N is the number of samples, ρ_j is the Euclidian distance of the j^{th} delay vector to its nearest neighbor, and c_E is the Eular constant ≈ 0.5772. Then

$$I(m,\tau) = \frac{H(x,m,\tau)}{\langle H(x_{s,i},m,\tau)\rangle}$$

is defined using surrogates (see [8] for details) and finally the entropy ratio (ER) is defined as

$$R_{ent}(m,\tau) = I(m,\tau)\left(1 + \frac{m\ln N}{N}\right) \tag{17}$$

by superimposing the minimum description length (MDL) method to penalize higher dimensions. This ratio is minimized to find the optimal set of embedding parameters (m^*, τ^*).

2) Partitioning: Time series sensor data are obtained from the input and output data streams of the dynamical system $\mathcal{D}_0$ under nominal condition under different input conditions. Let $\mathcal{Y} = \{y_1, y_2, ...\}, \quad y_k \in \Sigma$ denote the discretized output sequence. A D-Markov machine is next constructed, with states defined by symbol blocks of length D from $\mathcal{Y}$. The reader is referred to references [2] and [3] for an in-depth description of the procedure.

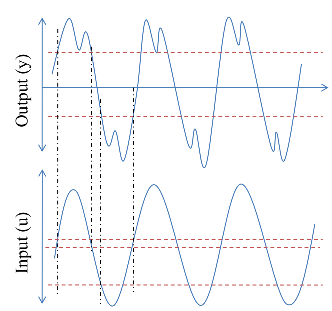

Fig. 1. Partitioning Scheme

The state space is constructed from the output space using Taken's theorem as discussed in the last section. In this case, the entropy rate analysis with the motor current data as the output yields an optimum embedding parameter $m^* = 1$. That is, the output, by itself, in this case constitutes a 1-dimensional phase space.

In the very next step, this phase space and the input space are individually discretized. The crux of the method is to place the partitions in such a way, that there is a change in alphabet in both, at the exact same instant there is a change in the already discretized output sequence. Synchronization of input and output is essential for this step. The partitioning scheme is illustrated in figure1.

Let $\mathcal{U} = \{u_1, u_2, ...\}$ denote the discretized input data sequence. Similarly let $\mathcal{Q} = \{q_1, q_2, ...\}$ denote the discretized state variable sequence. It may be noted that the state space can be multi-dimensional depending on the embedding dimension m^*.

Once the input and state space are both discretized, they can be combined to form the discretized augmented input space $\Lambda = \{\lambda_1, \lambda_2, ...\}$, where each $\lambda_i = \{q_i, u_i\}$.

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 [9] 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].

B. Identification

The training scheme shown in Fig. 2 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.

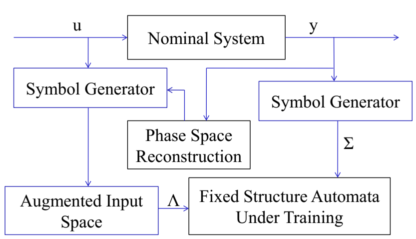

Fig. 2. Training Scheme

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

$$\delta = \{\delta^{\lambda_1}, \delta^{\lambda_2}, ..., \delta^{\lambda_m}\} \tag{18}$$

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 involves counting the frequency of each transition in the training phase. Since the state transition matrices are constructed simply by counting, this method is ideal for implementing in the sensor electronics for real-time prognoses.

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. In this work, a stopping rule [2] has been used for detecting the optimal data length.

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*.

C. Anomaly Detection Scheme

Figure 3 gives a schematic representation of the anomaly detection philosophy. 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.

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,

$$\delta^{\lambda_i}(q_j, q_k) = 1 \qquad \text{for one and only one } k \tag{19}$$

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 [10]. 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.

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,

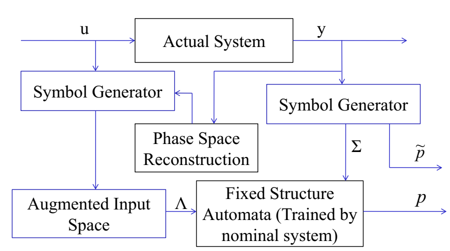

Fig. 3. General Anomaly Detection Scheme

but at each instance of state transition, *i.e.* at every instant the trained automaton produces a *State Transition Probability vector* π_n [10], which is characteristic of the nominal system.

It may be noted, that 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, state 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* [10] 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**.

IV. $PMSM$ APPLICATION

The proposed concept of anomaly detection is validated on a permanent magnet synchronous machine ($PMSM$) experimental testbed. The experiment was used for validating the diagnostic algorithm in extremely noisy situation.

Before the details of the experiment is provided, it may not be amiss to very briefly go through the governing equations for the $PMSM$ system and the control scheme that governs the electronic drive. This is done in the next section.

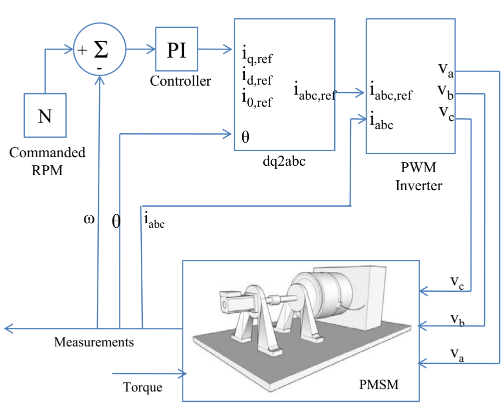

Fig. 4. Inverter-driven permanent magnet synchronous motor (*PMSM*) system

A. Description of the Experimental Apparatus

The $PMSM$ used for the experiment is a Baldor AC brushless servo motor. This particular series of brushless servo motor was chosen because of its use in robotics, aviation and numerous other industrial motion control applications.

In state-space setting, the governing equations of the $PMSM$ take the following form:

$$\frac{di_q}{dt} = (v_q - Ri_q - \omega_s L_d i_d - \omega_s \lambda_{af}) / L_q \quad (20)$$

$$\frac{di_d}{dt} = (v_d - Ri_d + \omega_s L_q i_q) / L_d \quad (21)$$

$$\frac{d\omega_r}{dt} = (T_e - T_L - B\omega_r) / J \quad (22)$$

where the subscripts q and d have their usual significance of quadrature and direct axes in the equivalent 2-phase representation, with v, i, and L being the corresponding axis voltages, stator currents and inductances; R and ω_s are the stator resistance and inverter frequency, respectively, while λ_{af} is the flux linkage of the rotor magnets with the stator.

The generated electromagnetic torque is expressed as:

$$T_e = 1.5P \left[\lambda_{af} i_q + (L_d - L_q) i_d i_q\right] \quad (23)$$

and the equation of motor dynamics is given by:

$$T_e = T_L + B\omega_r + J\frac{d\omega_r}{dt} \quad (24)$$

where P is the number of pole pairs, T_L is the load torque, B is the damping coefficient, ω_r is the rotor speed, and J is the moment of inertia. The rotor speed $\omega_r = \omega_s / P$.

In the control scheme shown, i_d is forced to be zero. Consequently,

$$\lambda_d = \lambda_{af} \quad \text{and} \quad T_e = 1.5P\lambda_{af} i_q \quad (25)$$

In the above equation, the torque T_e is proportional to the quadrature axis current because the magnetic flux linkage λ_{af} is constant.

The $PMSM$ used in the experiment is a three-phase four-pole device rated at $160\ V$ bus voltage, $4000\ rpm$ and is fed by a pulse-width-modulated (*PWM*) inverter. The stator resistance of the motor is $R_s = 11.95\ \Omega$; the quadrature-axis and direct-axis inductances are: $L_q = L_d = 16.5 \times 10^{-3} H$; and the rotor inertia is $J = 0.06774\, kg\ cm^2$.

Two PI controllers in two loops have been employed for controlling the power circuit that drives the $PMSM$. The inner loop regulates the motor's stator currents, while the outer loop regulates the motor's speed. In this control scheme, the error in the measured and the commanded speed generates the command quadrature axis current, which is directly proportional to the electromagnetic torque if the direct axis current is maintained to be 0. The line currents i_a, i_b and i_c are then measured. The reference values are compared with the actual values of the currents, and the error signal, thus constructed is used for generating the gate turn on/off commands.

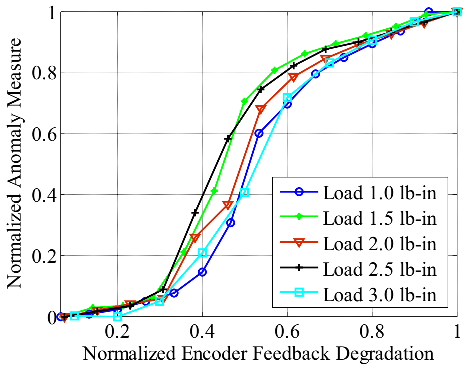

Fig. 5. Plot of Deviation Measure for the $PMSM$ System for different operating conditions

B. Details of the experiment

It may be noted that the $PMSM$ control scheme delineated in the preceding section is strongly dependent on a reliable position feedback which is supplied by the 2500 ppr line count incremental encoder. The encoder reading, in turn is strongly dependent on matching the index pulse to a $S-N$ orientation of one of the permanent magnets inside. In this particular experiment, the possible mismatch of the index pulse is treated as an anomaly. The task is to estimate this error from the input (voltage) and output (current) signals. In order to study the relative effect of input and system parameter variation, the $PMSM$ is made to operate at different load conditions ranging from 0 (no load) to 3.5 $N-m$ (full range of operation), in steps of 0.5 $N-m$.. The objective is estimation of even slight changes of the mismatch (offset) parameter, through observation of statistical behavior as early as possible without getting affected by the different operating conditions.

V. Results and Discussion

A series of $\mathbf{K}$ data is generated for the purpose of training by providing the nominal system a series of inputs, each at a different operating condition. For each operating condition, the motor speed is held constant at 1000 rpm by the speed controller. The line currents are used as output signal from the permanent magnet motor. This signal is partitioned using a maximum entropy scheme of partitioning. The entropy rate optimization yields $m^* = 1$, i.e a single dimensional phase space consisting of only the output signal. This phase space, along with the input signal, which is chosen to be the voltage input to the motor terminals is discretized next. The anomaly detection procedure described in section III is applied next. The angle measure is calculated and is shown in Fig. 5.

For investigating the effect of operating condition on the nature of the anomaly measure developed, the offset in the encoder is allowed to deteriorate more and more while running and collecting data at different motor loads. The different curves observed in figure 5 are obtained this way. It may be noted that irrespective of the operating condition of the motor, the anomaly curves qualitatively follow the exact same pattern Quantitatively, inverse problem techniques can be used to provide a confidence interval within which the offset lies. This implies that the deviation from nominal condition, that is observable in the anomaly measure is a consequence of purely the fault condition inside the motor, and not an artifact of the variation in the input data stream due to change in the operating condition.

VI. 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. Further theoretical, computational, and experimental work is necessary before the SDF-based anomaly detection tool can be considered for incorporation into the instrumentation and control system of commercial-scale plants. For example, an important failure condition in $PMSM$'s is demagnetization of the motors. To investigate the efficacy of the algorithm in detecting incipient demagnetization failure, a more advanced test apparatus, for demagnetizing the motors in a controlled manner is under development. At the same time, the following theoretical aspects are under investigation:

- Development of a multi-dimensional partitioning for a MIMO system, which should be computationally inexpensive.
- Estimation of a theoretical bound on the error incurred in this process of anomaly detection.

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] M. M. Kokar, "On consistent symbolic representations of general dynamic systems," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 8, pp. 1231–1242, 1995.

[7] F. Takens, *Detecting strange attractors in turbulence*. Springer Berlin / Heidelberg, 1981, vol. 898/1981.

[8] T. Gautama, D. Mandic, and M. V. Hulle, "A differential entropy based method for determining the optimal embedding parameters of a signal," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, no. 6-10, pp. VI – 29–32, 2003.

[9] 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.

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