Abstract—This paper proposes a syntactic method of system identification in dynamical systems. The underlying concept involves abstraction of a qualitative description of the dynamical system by state-space embedding of the output data stream and discretization of the resultant pseudo-state and input spaces. The task of system identification is achieved through grammatical inference techniques, and the deviation of the plant output from the nominal estimated language generates a measure of anomaly in the system. This system identification technique has been experimentally validated for detection of anomalous behavior on a laboratory apparatus of a permanent magnet synchronous motor (PMSM) undergoing gradual demagnetization.

Index Terms: System Identification, Symbolic Dynamics, Finite State Automata, Permanent Magnet Synchronous Motor

I. INTRODUCTION

In the last few decades, health monitoring of complex engineering systems has evolved to be an issue of paramount importance. However, the inherent complexity and uncertainty of such systems pose a challenging problem to health monitoring, since physics-based models, if available, are routinely oversimplified, or in worst cases they may not be available at all.

Recent research has extensively explored the problem of anomaly detection using symbolic dynamic filtering (SDF) [1]–[3]. But, apparently the system identification aspect of health monitoring 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, degradation even in a single component may affect the input streams to the neighboring components and thus pervade the entire system. Furthermore, the dynamical system could be operated in different regimes and thus under diverse input conditions. Operating conditions may also change due to exogenous factors, such as varying load requirements in a power plant or extreme maneuvering necessities of a fighter jet. The major challenges here are: (i) detection and isolation of faults and (ii) estimation of the fault magnitude, for the purpose of prognoses without a high-fidelity component-level model of the system.

The purpose of the work reported in this paper is to address this issue and develop a robust and computationally inexpensive system identification technique. The underlying theory is built upon the concept of formal language formulation. A central step in this class of identification methodology is discretization of the raw time-series measurements into a corresponding sequence of symbols. Key advantages of working with symbols include computational efficiency and robustness [1] [2]; specifically, the proposed system identification method is designed to be robust with respect to sensor noise and spurious disturbances and yet it is simple enough to be embedded within the hardware and software of 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 of a Permanent Magnet Synchronous Motor (PMSM) drive system. This apparatus is particularly relevant for testing & validation of condition-monitoring algorithms, because PMSMs are being increasingly used (for example, in both commercial and fighter aircraft) in a variety of commercial and non-commercial applications.

II. THEORETICAL BACKGROUND

Grammatical inference is an inductive inference problem where the target domain is a formal language and the representation class is a family of grammars. In the present context, a grammatical inference procedure is adopted for identification of non-autonomous dynamical systems as an entity (i.e., a linguistic source) that is capable of generating a specific language. The grammar $G$ of the language is a set of rules that specifies all words in the language and their relationships. Many physical processes can be represented by a dynamical system operating in continuous time and in continuous state space. 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 to abstract a discrete qualitative counterpart of the general dynamical system representing the physical process.

B. Identification

The learning task is to identify a “correct” grammar for the unknown target language based on a finite number of examples of the language.
C. Inference

Upon completion of the identification phase, the output of the abstracted system is compared to the actual output in the sense of an appropriate metric for estimation of the system’s degradation.

The above concepts are now developed in a systematic manner with reference to health monitoring of a human-engineered system (e.g., permanent magnet synchronous motor (PMSM)) undergoing gradual degradation.

Definition A general dynamical system (GDS), representing the underlying structure of a physical dynamical system, is defined as an 8-tuple (see [4] for details)

\[ D = (T, U, W, Q, P, f, g, \leq) \]  

where

- \( T \) is the time set,
- \( U \) and \( W \) are input and output sets, respectively,
- \( Q \) are internal states,
- \( P \) maps the input process \( P : T \rightarrow X \),
- \( f \) maps the global state transition as:
  \[ f : T \times Q \times P \rightarrow Q \]  
  for time-invariant systems (2)

- \( g \) denotes the output function
  \[ g : Q \rightarrow W \]  
  for time-invariant systems (3)

Let \( D_i \) be a dynamical system indexed by \( i \) representing different parametric conditions, \( D_0 \) being the nominal or healthy condition of the system, and \( i = 1, 2 \ldots \) signifying deteriorating health conditions of the plant due to a progressing anomaly. Let \( U_k \), \( k = 1, 2 \ldots K \), denote \( K \) different input conditions, and \( y_k^i \) be the output from the \( i \)th component \( D_i \), receiving the \( k \)th input \( U_k \). Next let the notion of grammatical representation be introduced through a Qualitative Dynamical System (QDS), of the nominal plant \( D_0 \).

Definition The quantized abstraction, called a Qualitative Dynamical System (QDS), of a GDS is a 5-tuple

\[ \mathcal{G} = \{Q, \Lambda, \Sigma, \delta, \gamma\} \]  

where

- \( Q \) is the finite set of qualitative states of the automaton, i.e. \( Q = \{q_1, q_2, ..., q_f\} \),
- \( \Lambda = \{\lambda_1, \lambda_2, ..., \lambda_m\} \) is the alphabet of input symbols,
- \( \Sigma = \{\sigma_1, \sigma_2, ..., \sigma_n\} \) is the alphabet of output symbols, where the output symbols are one-to-one with the quantized values of output from the dynamical system.
- \( \delta : Q \times \Lambda \rightarrow Q \) is the state transition function which maps the current state into the next state on receiving the input \( \lambda \). The transition function can also be stochastic in which case, \( \delta : Q \times \Lambda \rightarrow \text{Pr}\{Q\} \)
- \( \gamma : Q \rightarrow \Sigma \) is the output generation function which determines the output symbol from the current state. In its full generality, \( \gamma \) can be stochastic as well, i.e. \( \gamma : Q \rightarrow \text{Pr}\{\Sigma\} \)

Let \( \chi \) denote a set of qualitative abstraction functions

\[ \chi : D_0 \rightarrow \mathcal{G} \]  

It is noted that \( \chi \) is a 3-tuple, consisting of three individual abstraction functions.

\[ \chi = (\chi_{TQX}, \chi_Q, \chi_W), \]  

where

\[ \chi_{TQX} : T \times Q \times X \rightarrow \Lambda \]  

\[ \chi_Q : Q \rightarrow \Sigma \]  

\[ \chi_W : W \rightarrow \Sigma \]  

Kokar [4] introduced a set of necessary and sufficient conditions, or “consistency postulates” that the pair \( (\mathcal{G}, \chi) \) must satisfy in order to be a valid representation of the general dynamical system. In this paper, since the transfer function \( \delta \) of the QDS is probabilistic, the consistency postulates are redefined in a probabilistic sense. The modified consistency postulates are stated as follows.

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

\[ \gamma(\chi_Q(q)) = \chi_W(g(q)) \]  

\[ \chi_Q(f(t, q, x)) \sim \delta(\chi_Q(q), \chi_{TQX}(t, q, x)) \]

where \( X \sim P \) means the random variable \( X \) is distributed according to the probability distribution \( P \).

D. Abstraction

Theorem 2.1 (Kokar [4]): 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 \( Q_\pi \) describe a partition of \( Q \) defined as an inverse image of \( W_\pi \) through \( g \).

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

and let \( TQX_\pi \) describe a partition of \( T \times Q \times X \) defined as an inverse image of \( Q_\pi \) through \( f \).

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

Then \( Q_\pi \) is a maximal admissible partition of \( Q \), and \( TQX_\pi \) is an admissible partition of \( T \times Q \times X \) [4].

If the system model, i.e the equations governing the general dynamical system is known, the critical hypersurfaces or partitions can be analytically evaluated using theorem 2.1 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 \( g \).
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 [5]. The embedded phase space can be denoted by
\[ x(k) = [x_{k-\tau}, ..., x_{k-m\tau}], \]
where \( \tau \) is the time lag, and \( m \) is the embedding dimension. Takens’ theorem guarantees that, at least in the noise-free case, a system of state dimension \( s \) may be embedded using a maximum of \( m\tau \) lags where \( m\tau \geq 2s + 1 \).

In order to find optimum values of the embedding parameters \( m \) and \( \tau \), the literature reports many optimization routines. In this case, following [6] a differential entropy based method is used to simultaneously estimate the optimal set of embedding parameters \( (m^*, \tau^*) \). The method employs a single criterion - the entropy ratio between the phase space representation of a signal and an ensemble of its surrogates.

2) Partitioning: Sensor time series data are obtained from the input and output data streams of the dynamical system \( \mathcal{D}_0 \) at the nominal (i.e., healthy) condition under different input excitations. Let \( \mathcal{Y} = \{ y_1, y_2, \ldots \} \), \( y_k \in \Sigma \), the output alphabet, denote the discretized output sequence. A probabilistic finite state machine, called \( D \)-Markov machine [1], is next constructed, with states defined by symbol blocks of length \( D \) from \( \mathcal{Y} \). The reader is referred to references [1] and [2] for an in-depth description of the procedure.

![Partitioning Scheme](image)

The state space is constructed from the output space using Taken’s theorem as discussed in the last section. In the very next step, this phase space and the input space are individually discretized. The key issue in this method is the synchronisation between changes in input and output symbols. The partitioning scheme is illustrated in figure 1.

Remark A partition constructed this way is admissible but not maximal because this partition is a subpartition of the original partition proposed in Theorem 2.1.

Let \( \mathcal{U} = \{ u(1), u(2), \ldots \} \) denote the discretized input data sequence. Similarly let \( \mathcal{Q} = \{ q_1, q_2, \ldots \} \) 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 spaces are both discretized, they can be combined to form the discretized augmented input space \( \Lambda = \{ \lambda(1), \lambda(2), \ldots \} \), where each \( \lambda(i) = \{ q(i), u(i) \} \).

The transition function used in the current methodology has been designed to be stochastic. \( \delta : Q \times \Lambda \rightarrow Pr\{ Q \} \) gives the probability distribution of transition from state \( q_i \) to \( \{ q_1, q_2, \ldots, q_f \} \) on receiving an input \( \lambda_j \).

However, the function \( \gamma : Q \rightarrow \Sigma \), which maps the current state \( q_i \) to the current output symbol \( \sigma_i \), is completely deterministic. This is really an artifact of the state construction procedure [1].

E. Identification

It is assumed that inputs and outputs are time-synchronized. The state transition function \( \delta \) can be expanded into two dimensional matrices \( \delta^{\lambda_i} \), indexed by the input symbols. That means
\[ \delta = \{ \delta^{\lambda_1}, \delta^{\lambda_2}, \ldots, \delta^{\lambda_m} \} \]
where \( \delta^{\lambda_i} : q_j \times \lambda_i \rightarrow Pr\{ Q \} \) maps the current state and input to the probability distribution over all possible states. The algorithm for estimating the matrices \( \delta^{\lambda_i} \) 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 \( \delta^i \) converge. The convergence depends on the length of the input-output symbol sequences. In this work, a stopping rule [1] 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.

F. Anomaly Detection Scheme

Figure 2 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 \( \delta \) is fully deterministic, i.e. given the current state \( q_j \) and the current input symbol \( \lambda_i \),
\[ \delta^{\lambda_i}(q_j, \lambda_i) = [p_{q_{i1}} p_{q_{i2}} \ldots p_{q_{i|\Sigma|}}]^T \] (13)
where \( p_{qi} = 1 \) for one and only one \( k \) (14)
 otherwise (15)

In the current scheme, the state transition probability vectors \( \pi^{\lambda_i} \), which are the rows of the state transition matrix \( \delta \), serve as feature vectors, and are used for the purpose of anomaly detection.
In the present study, a novel intuitive **Pseudo-Learning Technique** of utilizing the stochastic state transition function \( \delta \) 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 \( \pi_n \), which is characteristic of the nominal system.

It may be noted, that the pattern vector \( \pi_n \), 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 \( \{ \pi_n \} \) and \( \{ \tilde{\pi}_n \} \) obtained at each time instant, to construct a pattern vector. This is followed by calculation of mean State Probability vectors \( p \) and \( \tilde{p} \) from the collections \( \{ \pi_1, \pi_2, ..., \pi_n \} \) and \( \{ \tilde{\pi}_1, \tilde{\pi}_2, ..., \tilde{\pi}_n \} \) respectively over time instants 1, 2, ..., \( n \). It may be noted that in an ideal case, \( p \) should converge to \( \tilde{p} \), while they should start to diverge from each other as the fault progresses. Thus the difference of these two probability vectors, \( p - \tilde{p} \) is a natural choice for constructing the pattern vector corresponding to that specific fault condition.

Once the pattern vectors for a fault condition are obtained, a suitable classification algorithm, such as a support vector machine can be utilized to create the hyperplane separating the nominal patterns from the possibly off-nominal pattern vectors.

### III. PMSM Application

The proposed concept of anomaly detection is validated on a permanent magnet synchronous machine (PMSM) experimental testbed.

#### 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:

\[
\frac{d v^r_q}{dt} = \left( v^r_q - R_s i^r_q - \omega_s L_q i^r_d - \omega_s L_d i^r_q \right) / L_q
\]

\[
\frac{d i^r_d}{dt} = \left( v^r_d - R_s i^r_d + \omega_s L_q i^r_q \right) / L_d
\]

\[
\frac{d \omega_r}{dt} = \left( T_e - T_L - B \omega_r \right) / J
\]

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. \( R_s \) is the stator resistance and \( \omega_r = \frac{P}{2} \omega \) is electrical rotor velocity respectively, \( P \) being the number of pole pairs and \( \omega_r \) being the rotor speed. \( \Lambda_{PM} \) is the flux linkage of the rotor magnets with the stator. Superscript \( r \) denotes that the equations have been set up in the rotating reference frame.

The generated electromagnetic torque can be expressed as:

\[
T_{e,3ph} = \frac{3P}{4} \Lambda_{PM} \left[ -i_q \sin(\theta_{re}) + i_r \cos(\theta_{re}) \right]
\]

where \( \theta_{re} = \frac{P}{2} \theta_r \) is the electrical rotor angle. The corresponding expression for torque in the rotor reference frame is given by:

\[
T_e = T_L + B \omega_r + J \frac{d \omega_r}{dt}
\]

where \( T_L \) is the load torque, \( B \) is the damping coefficient, and \( J \) is the moment of inertia.

The schematic of the entire experimental setup has been shown in Figure 3. The whole figure is divided into several blocks for convenience, each of which will be described next.
1) Controller Block: The three-phase four-pole \textit{PMSM} is rated at \(160\, \text{V bus voltage, 4000 rpm}\) 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: \(L_q = L_d = 16.5 \times 10^{-3}\, \text{H}\); and the rotor inertia is \(J = 0.06774\, \text{kg cm}^2\). Two PI controllers in two loops have been employed for controlling the power circuit that drives the \textit{PMSM}. The inner loop regulates the motor’s stator currents, while the outer loop regulates the motor’s speed.

2) Direct magnetic flux linkage estimation block: As the permanent magnet inside the \textit{PMSM} slowly deteriorates, it is imperative to be able to measure the extent of demagnetization by some kind of direct technique, at each stage of demagnetization, so that the degree of anomaly predicted by \textit{SDF} can be mapped to this physical quantity. Several researches have dealt with the estimation of the flux of a \textit{PMSM}. It is of no doubt that the no-load test method has become the most popular one. In this method, an auxiliary motor is required to drive the \textit{PMSM} at constant speed. The windings of \textit{PMSM} are at open-circuit so that the flux can be estimated by the \textit{EMF} of the \textit{PMSM}. The two phase stator voltage in the rotor reference frame is given by

\[
\tilde{V}^r = R\tilde{I}^r + L \frac{d\tilde{I}^r}{dt} + \omega_{re} J \left( L\tilde{I}^r + \tilde{\Lambda}_{PM}^r \right)
\]

Here \(J\) is the \(90^\circ\) rotation matrix. Under steady state conditions, the derivative of the current vector is dropped, and the voltage expression becomes:

\[
\tilde{V}^r = R\tilde{I}^r + \omega_{re} J \left( L\tilde{I}^r + \tilde{\Lambda}_{PM}^r \right)
\]

where \(\omega_{re}\) is the steady state electrical rotor velocity. In open circuit there are no currents flowing in the windings of the machine, so the voltage is entirely due to the permanent magnet flux linkage, hence

\[
\|\tilde{V}\| = \|J\omega_{re}\tilde{\Lambda}_{PM}^r\| = \frac{P}{2} \omega_{r} \Lambda_{PM}
\]

The permanent magnet flux linkage is therefore

\[
\Lambda_{PM} = \frac{\|\tilde{V}\|}{\frac{P}{2} \omega_{r}} = \frac{\sqrt{2} V_{l-1\, \text{RMS}}}{P \times 2\pi \times \text{rpm} \times 60}
\]

At each stage of demagnetization, the line-to-line voltage and the motor speed in \(\text{rpm}\) is recorded. The permanent magnet flux linkage is estimated from these following Eqn. 25.

3) Operating Condition and Input/Output Block: The desired rpm and the load torque is set in the operating condition block. The load torque is directly set by the dyne. All the variables are captured and stored in the output block, while the input voltage commands are also saved in the input block.

It may be noted, that blocks III-A.1 and III-A.2 are never engaged simultaneously, because that would result in a fight between the external back-driving motor and the \textit{PMSM} controller.

\[\text{Fig. 4. Degradation in current output}\]

B. Demagnetization Procedure

The electromagnetic torque (Eqn. 19) is proportional to the cross-product between the current vector and the permanent magnet flux linkage vector. For a given current magnitude, torque is therefore maximized if the direction of the field generated by the stator windings is perpendicular to the direction of the field generated by the permanent magnets. The resulting torque attempts to align these fields. The PWM controller attempts to generate the line current in the three phases in such a way, that this orthogonality is maintained.

It may be noted that orientation feedback is vital to this scheme. In the present experiment, during demagnetization, just enough offset is added to the encoder orientation so that instead of being orthogonal, the stator winding field directly opposes the field generated by the permanent magnets. Thus, since the two fields are aligned, no torque is generated, instead the permanent magnets slowly lose their magnetism. The considerable amount of heat generated in the process enhances the loss of magnetic property in the permanent magnets.

C. Details of the experiment

In each set of experiments, the following order has been maintained.

1 A direct estimation of the magnetic flux is made and the temperature is noted.
2 Voltage and line current data is recorded by making the motor spin at 1000 \(\text{rpm}\). 2 sets of data are collected, corresponding to a load of \(1.5\, \text{lb} - \text{in}\) and \(2\, \text{lb} - \text{in}\) respectively. It is apparent from Figure 4 that the fault
signature in the output signal \(i_c\) is apparently absent. This is because the controller actively adjusts for any deterioration in the system components in the control loop.

3. The motor is demagnetized following the procedure outlined in section III-B till the temperature rises to a predetermined value. The rise in temperature of the motor, which is an effect of this demagnetization procedure, is also used as an indicator of how much the demagnetization has progressed.

4. The procedure is repeated from Step 1.

The whole experiment is repeated 14 times to assess the robustness of the procedure. The temperatures which mark the end of each individual demagnetization run and the corresponding reduction in the magnetic flux linkage measured by the no-load test method are listed in Fig. 5.

IV. RESULTS AND DISCUSSION

The training set comprises of the input signal profile of one of the three line voltages \(v_c\) and the output signal profile of the line current \(i_c\) for both load conditions. The output signal \(i_c\) is discretized via maximum entropy partitioning. The number of states in the PFSA is varied from 5 to 25 to estimate the effect of \(|\Sigma|\) on detection performance. The augmented input space is constructed by discretizing the input and phase space. The input specific probabilistic state transition matrices are next constructed, which concludes the training of the PFSA.

In the validation part, the input and output data corresponding to a single fault level (for example, a loss of 0.3879% of PMSM magnetic strength) but for different load conditions, is fed into the algorithm. The pattern vector cluster, formed by data from multiple runs and different load conditions, corresponding to this fault condition is calculated according to the algorithm.

A support vector machine with linear kernel is used to classify the nominal from the off-nominal cases. The validation is done by choosing one of the data sets as test data and using the remaining data as the training data, and noting whether it could be classified correctly. This is repeated for all the data sets to yield a True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR) and False Negative Rate (FNR). Here “positive” denotes nominal condition and “negative” denotes off-nominality.

Figure 6 shows the result of the SVM classifier when applied to pattern vectors corresponding to different motor health conditions. Fig.6(a) shows that all levels of demagnetization, even a decrease as small as 0.3879% can be correctly detected (TPR=100%, FPR=0%) when \(|\Sigma| = 25\).

The corresponding result for alphabet size \(|\Sigma| = 15\) is shown in Fig. 6(b). The ROC curve moves away from the top left hand corner implying that an alphabet size of 15 does not have the resolving capacity to distinguish between nominal signal and very weak fault signature. For higher levels of fault, the ROC moves closer to the left hand top corner. For \(|\Sigma| = 5\), (Fig. 6(c)) the same trend is repeated, thus validating that there exists a \(|\Sigma|_{\text{min}}\) below which the classifier performance deteriorates.

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. The following theoretical aspects are currently under investigation:

- Estimation of a theoretical bound on the error incurred in this process of anomaly detection.
- Estimation of a theoretical \(|\Sigma|_{\text{min}}\)

REFERENCES