An Investigation of State-Space Model Fidelity for SSME Data

Rodney A. Martin
NASA Ames Research Center
Mail Stop 269-1
Moffett Field, CA 94035-1000, USA
(650) 604-1334
Rodney.Martin@nasa.gov

***Abstract*—In previous studies, a variety of unsupervised anomaly detection techniques for anomaly detection were applied to SSME (Space Shuttle Main Engine) data. The observed results indicated that the identification of certain anomalies were specific to the algorithmic method under consideration. This is the reason why one of the follow-on goals of these previous investigations was to build an architecture to support the best capabilities of all algorithms. We appeal to that goal here by investigating a cascade, serial architecture for the best performing and most suitable candidates from previous studies.**

As a precursor to a formal ROC (Receiver Operating Characteristic) curve analysis for validation of resulting anomaly detection algorithms, our primary focus here is to investigate the model fidelity as measured by variants of the AIC (Akaike Information Criterion) for state-space based models. We show that placing constraints on a state-space model during or after the training of the model introduces a modest level of suboptimality. Furthermore, we compare the fidelity of all candidate models including those embodying the cascade, serial architecture. We make recommendations on the most suitable candidates for application to subsequent anomaly detection studies as measured by AIC-based criteria.

I. Introduction

This paper is a continuation of two previous studies [18] - [19], in which various unsupervised anomaly detection algorithms were applied to SSME data. The SSME (Space Shuttle Main Engine) is a complex re-usable liquid propulsion system, and is outfitted with a comprehensive array of sensors (vibration, facility, and controller measurements). There are three SSME's and two SRB's (Solid Rocket Boosters) used to support the launch of the space shuttle for ongoing missions prior to its retirement. Although the shuttle is due to be retired in 2010, the investigation of anomaly detection algorithms applied to this SSME dataset can be justified for a variety of reasons.

The acquisition of authentic operational and supporting truth data from which to perform rigorous statistical analyses is a rare commodity from the perspective of propulsion for space applications due to their design for high reliability and relatively low failure rates. While the SSME dataset described herein does not by any means represent a consistent, comprehensive dataset from which to generate a statistically significant analysis, there are certainly ways to interpret and analyze the data in a manner that may serve to support continued achievement of ISHM (Integrated System Health Management) goals.

Furthermore, this dataset can act as a baseline for the development of algorithms related to future generations of spaceflight, i.e. the Ares I and Ares I-X launch [29]. The methods investigated and developed from this dataset are also certainly more generally applicable to a broader class of IVHM (Integrated Vehicle Health Management) application platforms. One example would be the application of derived techniques to civil aeronautics platforms, and more fundamentally to aeronautics research. As such, even though the dataset is application-specific, our intent is to demonstrate the utility of our findings from a much broader perspective.

One of our primary goals in this paper is to introduce the notion of model fidelity, a topic previously lacking in other analyses. The suite of algorithms under consideration for our purposes here is a smaller subset of ones used in previous analyses. Many of the algorithms used previously were fairly mature and had been successfully deployed onboard critical application platforms. Others operated at lower TRL (technology readiness level), but are nonetheless viable candidates for consideration. The more mature algorithms are IMS (Inductive Monitoring System) [13], GritBot, and Orca [1], [28]. The research-stage algorithms are SVM (Support Vector Machines) [4], various implementations of the GMM (Gaussian Mixture Model), and an LDS (Linear Dynamic System).

First we will establish the requirements making an algorithm a suitable candidate for inclusion in the serial architecture in Sec. II. An appropriate selection is made based upon these requirements. We will then provide a detailed discussion of the state-space model framework in Sec. III. This section will also cover various initialization and learning strategies under consideration in Sec. III-A. The model likelihood and its relationship to the Kalman filter updates is detailed in Sec. III-B, and the method for assessing model fidelity is covered in Sec. III-C. Finally, we will present the results in Sec. IV and provide concluding remarks in Sec. V.

II. Requirements

The requirements that would make an algorithm a viable candidate for our purposes here are provided in the following list.

TABLE II
TRAINING/VALIDATION BREAKDOWN

Data Sources	Training	Validation	
	Nominal	Nominal	Potential Anomalies
Flight Data	STS-77 (#1)	STS-103 (#2)	STS-77 (#2)
	STS-78 (#1)	STS-103 (#3)	STS-91 (#1)
	STS-78 (#2)	STS-106 (#1)	STS-93 (#1)
	STS-78 (#3)	STS-106 (#2)	STS-93 (#3)
Test Stand Data	A10851	A10852	A10853
	A20726	A20750	A20619

1) The method is conducive to temporal analysis (i.e. a likelihood, probability, or other relevant score-based metric can be generated for each point in time to allow for the construction of an ROC curve and subsequent design of an alarm system).
2) The method provides an informative composite non-zero score for a multivariate time series dataset that is available during the training phase, and will retain inherent dynamic structure.
3) The scores can be compared to some relevant predetermined threshold, whether derived statistically, experimentally, or is inherent to the algorithm itself. By constructing a candidate level-crossing event involving the output of a linear dynamic state-space system and a relevant failure threshold, we can mitigate false alarms by invoking the principle of optimal alarm as suggested in [22].

The first of these requirements, Req. 1 addresses the lack of sufficient data per flight cycle that has been categorized as containing anomalies, faults, or failures. This dearth of anomalous truth data inhibits the ability to generate an ROC curve with any reasonable level of statistical significance. However, using the same dataset we can address this lack of data by constructing an ROC curve based upon a temporal labelling of the truth data in lieu of per flight cycle. The time and severity of each anomaly corresponding to our dataset is shown in Table I. Additionally, the descriptions of the anomalies and their functional categorizations are provided.

Due to the availability of the temporal information for all anomalies, we can construct a statistically significant ROC curve based upon each time point rather than each flight cycle. Table II contains the corresponding meta-data for each documented anomaly. This table identifies the datasets of interest and categorizes them according to their source. They are also categorized according to which flights are used to train models in this study, and which will be used for validation in a subsequent study to be presented in a sequel paper. The validation data is partitioned into flights that contain anomalies and those that are nominal. As determined in [19], two of the flights that were originally categorized as nominal required reclassification due to mild anomalies that had not previously been labelled as such. This is evident in the anomalous flights that have been listed with the time and severity of each anomaly shown in Table I.

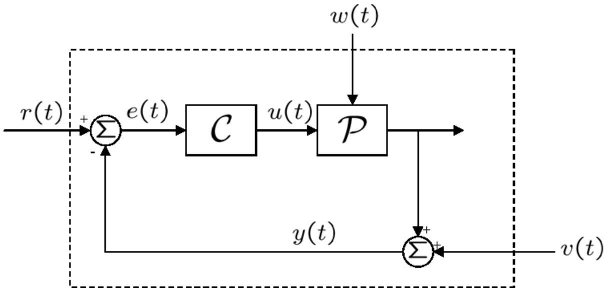

Fig. 1. Closed-Loop Control System Block Diagram

Requirement 2 addresses the need for retaining the inherent dynamic structure of the data as well as any potential failure or anomalous signatures. In order to construct a model that is dynamically informative we require that the training data is assigned a composite score that is non-zero for all time points. The data must also not be randomized in order to preserve its dynamic integrity. This dynamic integrity is important in order to ensure development of a model that can learn hidden causal relationships during training.

Finally, requirement 3 addresses the target anomaly detection algorithm based upon use of the state-space models that we will investigate in subsequent studies. The incorporation of a predefined threshold into the design of a state-space based alarm system is particularly useful due to the inherent capability to mitigate false alarms, as discussed in [22] and [17]. Therefore, we will use a state-space formulation by default, under the assumption that the structure of the model is a linear dynamic system driven by Gaussian noise, and has a univariate output.

One of the primary justifications for requirement 3 is related to the introduction of an architecture to support the best capabilities of all algorithms. A cascade, serial architecture as shown in Fig. 2 promotes synergism among the best performing and most suitable candidates from previous studies.

The idea behind using control system error as the sole indicator for anomaly detection (shown in Fig. 1 as $e(t)$) lies in the fact that the SSME throttle control system was most likely designed with both reference command following and disturbance rejection in mind. As such, when large disturbances influence the plant, $\mathcal{P}$, to the extent that the control system cannot reject them expediently, this may be indicative of a significant event which is cause for diagnostic investigation.

In previous studies, it was found that alarm systems developed on this concept performed moderately well for specific types of anomalies, both in terms of accuracy and time to detection. Another reason for using control system error as the sole indicator for anomaly detection was the possibility of constructing an optimal alarm system based upon a linear dynamic system trained by the control system error. As such, the principles of optimal alarm can be invoked in order to mitigate false alarms.

However, it was found in [19] that when considering the control system error as the basis for training a univariate

TABLE I
CHARACTERIZATION OF FAILURES

Failure Data	Failure Type	Time of Anomaly	Severity
STS-77 (#2)	Anomalous Spike in Sensor Reading (Controller)	74.42 sec	Mild
STS-91 (#1)	Sensor Failure (Controller)	32.76 sec	Mild
A10852	Mixture Ratio Change (Controller)	210 sec	Mild
STS-103 (#3)	Max Noise Failure (Vibration)	38.1 sec	Mild
STS-93 (#1)	Controller Failure (Controller)	11.38 sec	Moderate
STS-93 (#3)	Fuel Leak and Controller Failure (Controller)	11.62 sec	Moderate to Severe
A20619	Knife Edge Seal Crack (Vibration)	119 sec	Moderate to Severe
A10853	Turbine Blade Failure (Vibration)	130 sec	Severe

linear dynamic system, the performance was poor for controller failures. This was due to the total loss of power to the controller, resulting in sensor readings of zero for both commanded and actual throttle. Because control system error is defined as the difference between commanded and actual throttle, as shown in Fig. 1, a very clearly anomalous condition can be mistaken for an otherwise nominal value of zero. This provides further evidence for the use of an architecture that will detect anomalies of all types by incorporating multiple methods.

It was also found in the same study [19] that overall accuracy and time to detect for SVM and Orca were better than for most other algorithms on average. As such, the idea behind the proposed cascade serial architecture shown in Fig. 2 is to allow for a data reduction that will incorporate the characteristics of the algorithms with the best performance.

Although Fig. 2 illustrates parallel branches, the serial portion of the architecture involves the lower branch in which a composite anomaly score is generated. After significant preprocessing is performed in order characterize nominal behavior, the anomaly score should contain no pathologically high values. This anomaly score can then be used as training data for the linear dynamic system. The upper branch corresponds to the control system error, which uses a much smaller fraction of the feature space than the lower branch (as indicated by the thicker line for the lower branch). Independent linear dynamic system models are generated from each branch for comparison only, and the common training data set is shown as the source for both methods solely for convenience.

There are reasons other than pure performance documented in the previous study cited earlier [19] for eliminating certain algorithms as candidates for the serial architecture. These reasons are mainly related to inadmissability due to the lack of meeting the three requirements that have been set forth. All of the candidate algorithms with the exception of LDS can learn a model based upon a multivariate time series. In fact, the SVM algorithm can even generate a composite score that can be compared to a relevant predetermined threshold that is inherent to its theoretical construction (the margin to the hyperplane as measured by Euclidean distance). However, none of these techniques other than LDS can independently apply the principles of optimal alarm based upon a predefined level-crossing event.

This cascade serial architecture (Fig. 2) is an attempt to utilize the best characteristics of algorithmic candidates that meet the requisite criteria. Essentially, SVM and Orca act to transform the training data from a multivariate to a univariate time series of the type required by LDS for training, via data reduction resulting in an anomaly score. It is therefore possible that other techniques such as unreleased future versions of IMS, or even PCA (principle components analysis) could also be used for this data reduction.

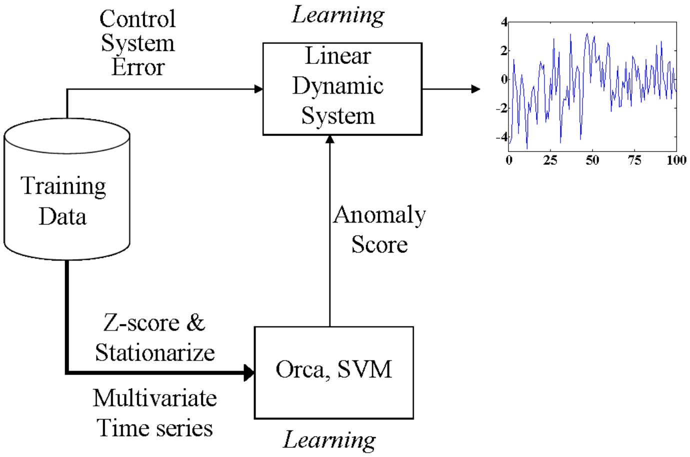

Fig. 2. Serial Architecture

The reason why standard algorithms may be used interchangeably with other types of static maps that are fundamentally transformations and/or data reduction techniques is due to the fact that the training dataset itself is used both during the learning and monitoring phases. The LDS algorithm should not be subject to “inheriting” undesirable characteristics of the “parent” algorithm delivering its anomaly score as training data (i.e. SVM or Orca). This is due to the fact that no decisions made by the parent algorithm are incorporated into LDS validation. As such, the anomaly score should be viewed purely as a transformation of the training data.

Prior to assignment of an anomaly score by these algorithms, the multivariate time series data is preprocessed by z-scoring and using a method called stationarization. Stationarization is a technique that has previously been used for SSME data analysis by Park et al. [25]. Stationarization is used in tandem with z-scoring to remove the effect of nonstationarities that arise in the resulting anomaly score as a result of varying operational modes, which may otherwise trigger spurious alarms. Z-scoring conditions the data so as to eliminate any bias introduced by inconsistencies in measurement units for various parameters.

The datastream for either branch of the serial architecture should appear identical, as shown in Fig. 2. This can be enabled by ensuring that the qualitative characteristics of the relative frequency histogram of data for the input to the LDS learning block appear Gaussian. Alternative Gaussian transformation methods such as the one described in [3] are viable candidates for future study. However, for the purpose of this study, we will consider just the two candidate preprocessing methods previously described, under a number of different settling time scenarios to be discussed shortly.

Unfortunately, for the SVM algorithm, the stationarization step may eliminate the inherent capability of the composite score to demonstrate a qualitative representation of the specific parameter with the most egregious behavior. The composite anomaly score is inherently a distance-based metric, i.e the distance from the monitored point to the hyperplane acting as the decision boundary. However, if only z-scoring is performed, the inherent non-stationarity in the anomaly score will enable all operational regimes visited to present as a multimodal distribution. Such a distribution is clearly not amenable to the univariate Gaussian representation required for training an LDS.

For Orca, the composite score is also a distance-based metric, however there is very little difference between the qualitative nature of the distributions when using both z-scoring and stationarizing as opposed to just using z-scoring for preprocessing. Due to the nature of the manner in which the distance calculations are performed, the distributions are also very skewed to the left. Neither distribution for the SVM or Orca scores will be centered at zero. For either distribution to possess a non-zero mean does not present an issue, because all of the methods to be described use a zero mean without loss of generality, and are used for mathematical convenience.

Fig. 3 shows the relative frequency histogram of the control system error data resulting in the realization of a learned model shown to the right of the LDS block shown on top of Fig. 2. This is qualitatively a good fit to the Gaussian model, as supported by the superimposed fit of the Gaussian curve plotted as a function of relevant measured statistics. The empirical mean and variance prior to training are functions of LDS model parameters, and corresponds to the curve shown in red, while the curve shown in cyan is the fit after training. In Fig. 3 the distinction between these two curves is barely discernable, as they are nearly identical.

Fig. 4 shows the relative frequency histogram for the anomaly scores of both SVM (top) and Orca (bottom). It is apparent that these methods do not provide as good a fit to the data under the Gaussian distribution assumption as shown for the case of Fig. 3 for control system error. As such, one possibility which was explored was to increase the number of points removed due to transient behavior by allowing for an increase in the length of the recovery period following major throttling transients. 1 sec was documented as the settling time allowed for and used in previous studies [18],[19]. We only consider periods of steady-state behavior for the purposes of this and previous studies. Transient periods may be investigated for all subject algorithms in future work.

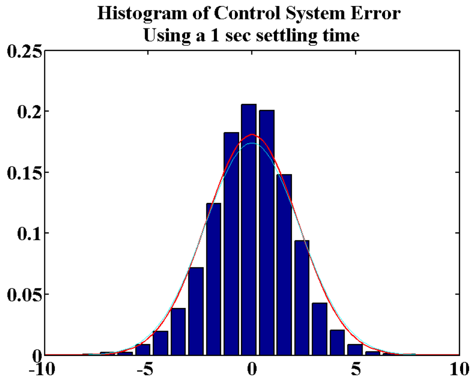

Fig. 3. Empirical Distribution of Control System Error Training Data

It was found that in a completely qualitative sense, the best settling times to allow for the best Gaussian fit to the SVM and Orca anomaly scores were 5 sec and 8 sec, respectively. These settling times were found by investigating values ranging from 1 sec to 10 sec. We will use more thorough quantitative means with which to assess the fit of the model and its Gaussian assumptions to the corresponding score-based data in subsequent sections. It is interesting to note that the histograms of anomaly scores for both SVM and Orca appear skewed, and are skewed in opposite directions to each other. This is most likely due to the manner in which the distance-based metrics were defined. Furthermore, for the SVM algorithm the Gaussian radial basis function (RBF) is used as the kernel operator. In future studies, we will attempt to exploit this fact to achieve a better qualitative fit of the data to the model, in addition to methods such as the one described in [3].

Of the more mature algorithms, GritBot is a commercially available decision tree based algorithm. GritBot does not provide a score for each monitored data point, but instead provides a list of the top anomalous scores that are ranked according to their statistical significance. As such, it is not a suitable candidate for real-time monitoring and implementation. IMS is an unsupervised machine learning algorithm that uses clustering to form a nominal region represented by the union of a finite number of hyper-rectangular clusters. By definition, IMS assigns a score of zero to all monitored points that fall within this nominal region. As such, the nominal training data cannot be used to train an LDS model since they will all have zero values by default.

The GMM was the poorest performer in the JANNAF study from the perspective of accuracy. However, when implemented in its most exhaustive variant (one GMM per parameter), the time to detect was on average on par with Orca. The GMM is a simple technique to fit multimodal Gaussian distributions to data under an IID assumption. Because the GMM is geared for isolation of anomalies in this particular implementation, there is inherently no viable data reduction technique that

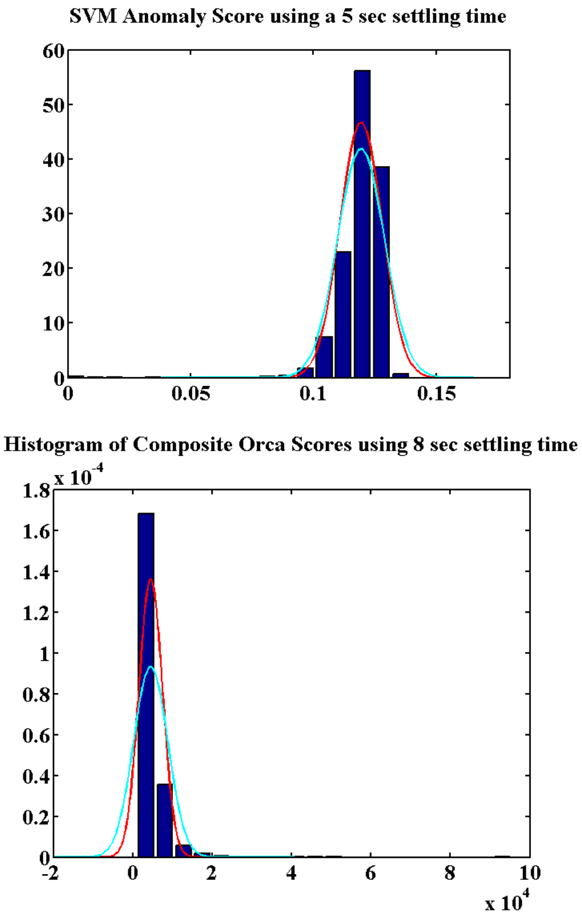

Fig. 4. Empirical Distribution of SVM and Orca composite anomaly scores

TABLE III
REQUIREMENTS MET BY ALGORITHMIC CANDIDATES

Requirement	Orca	IMS	SVM	GritBot	GMM	LDS
1	Yes	Yes	Yes	No	Yes	Yes
2	Yes	No[a]	Yes	No	No[b]	No[c]
3	No	No	No	No	No	Yes

[a] Zero-scores for nominal IMS training data applicable only for current working version of IMS, may change in future versions

[b] No composite score available for best performing variant

[c] Does not process multivariate time series datasets

would result in a single, composite numerical score. For these reasons, it will not be considered as a viable candidate. Table III summarizes the algorithms discussed thus far and which requirements are met by each. It is therefore clear that by using Orca or SVM in tandem with LDS, all of our requirements are fully covered.

With the proposed architecture in mind, we return to our primary focus, which is to assess model fidelity for both branches of the algorithmic setup shown in Fig. 2. In order to make a valid argument for the practical use of any anomaly detection algorithms that result from the application of the architecture in Fig. 2, we will need to address the issue of model fidelity directly, both quantitatively and qualitatively. Ultimately, any such model will also affect its subsequent application to this anomaly detection algorithm. We will fully discuss the technical details of model fidelity in the subsequent section.

III. METHODOLOGY

We will investigate both quantitative and qualitative aspects of model fidelity, as discussed by [2] and [26], respectively. Our objective here is to apply reasonable judgements and metrics with which to assess and ultimately choose the model that best describes the data. Variants of the serial architecture shown in Fig. 2 will be explored. Our underlying assumption is that we can fit measured or transformed data to a model represented by a linear dynamic system driven by Gaussian noise. The state-space formulation is shown in Eqns. 1-2.

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{w}_k &\quad (1)\\ y_k &= \mathbf{C}\mathbf{x}_k + v_k &\quad (2) \end{aligned}$$

where

$$\begin{aligned} \mathbf{w}_k &\sim \mathcal{N}(0, \mathbf{Q})\\ v_k &\sim \mathcal{N}(0, R)\\ \mathbf{x}_0 &\sim \mathcal{N}(\mu_{\mathbf{x}}, \mathbf{P}_0)\\ \mu_{\mathbf{x}} &= E[\mathbf{x}_k]\\ \mathbf{P}_0 &= E[(\mathbf{x}_0 - \mu_{\mathbf{x}})(\mathbf{x}_0 - \mu_{\mathbf{x}})^T] \end{aligned}$$

The state of the system, $\mathbf{x}_k \in \mathbb{R}^n$ evolves according to these equations, and often characterizes some internal physical characteristic of the system, beginning at time $k = 0$, with value $\mathbf{x}_0$ via state matrix $\mathbf{A}$. The scalar output of the system is given by $y_k \in \mathbb{R}$, and evolves through output matrix $\mathbf{C}$. Both input noise ($\mathbf{w}_k$), which influences the state trajectory, and measurement noise, (v_k) which influences the output are introduced in order to allow for a more realistic model. The noise is modelled stochastically via a standard Gaussian distribution with means and covariances specified above. $\mu_{\mathbf{x}}$ is the mean of the state trajectory, and $\mathbf{P}_0$ is the initial state covariance. Therefore the parameters to be learned are specified below, as the parameter θ. These parameters are also shown in Fig. 5, which specify them in relation to the probabilistic graphical modeling paradigm which may be used for machine learning purposes.

$$\theta = (\mu_{\mathbf{x}}, \mathbf{P}_0, \mathbf{A}, \mathbf{C}, \mathbf{Q}, R) \quad (3)$$

Let $|\cdot|$ represent the number of elements that comprise a parameter of θ, for example

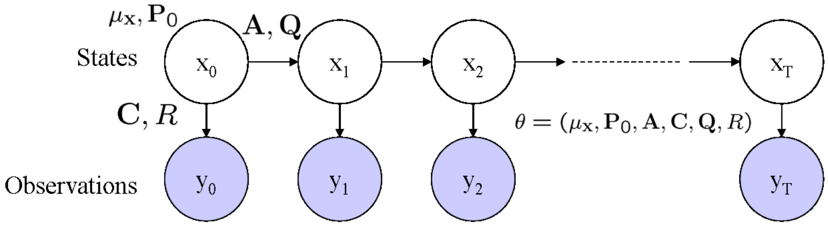
$\mu_{\mathbf{x}}, \mathbf{P}_0$
$\mathbf{A}, \mathbf{Q}$
States
x_0
x_1
x_2
x_T
$\mathbf{C}, R$
$\theta = (\mu_{\mathbf{x}}, \mathbf{P}_0, \mathbf{A}, \mathbf{C}, \mathbf{Q}, R)$
Observations
y_0
y_1
y_2
y_T

Fig. 5. Linear Dynamic System

$$
\begin{aligned}
|\mu_{\mathbf{x}}| &= n \\
|\mathbf{A}| &= n^2 \\
|\mathbf{C}| &= n \\
|\mathbf{Q}| &= n^2 \\
|\mathbf{P}_0| &= n^2 \\
|R| &= 1
\end{aligned}
$$

Therefore, a formula for the total number of parameters to be learned is shown in Eqn. 4.

$$|\theta| = |\mu_{\mathbf{x}}| + |\mathbf{P}_0| + |\mathbf{A}| + |\mathbf{C}| + |\mathbf{Q}| + |R| = 3n^2 + 2n + 1 \quad (4)$$

The notation for the equations is shown as in [17] for simplicity and generality. However, if we consider the special case of control system error as the output, y_k, the notation becomes trickier to bookkeep, as was performed in [18]. Furthermore, the details on discretization of the continuous-time LDS and initialization using basic assumptions are also as presented in [18]. A brief summary is provided here, as these details relate to one of various initialization/learning strategies we will investigate.

The state dynamics of an open-loop plant, $\mathcal{P}$, as shown in Fig. 1 can be expressed by equations 5-6[1]

$$
\begin{aligned}
\dot{\mathbf{x}}(t) &= \mathbf{A_c}\mathbf{x}(t) + \mathbf{B_c}\mathbf{u}(t) + \mathbf{\Gamma}_c w(t) \quad (5) \\
y(t) &= \mathbf{C_c}\mathbf{x}(t) + v(t) \quad (6)
\end{aligned}
$$

where

$$
\begin{aligned}
w(t) &\sim \mathcal{N}(0, Q_c) \\
v(t) &\sim \mathcal{N}(0, R_c)
\end{aligned}
$$

The controllable canonical form shown in Eqns. 7-9 is used to allow for a mapping to intuitive canonical parameters: the natural frequency, ω_n, and the damping ratio, ζ. Constraining ourselves to this basic canonical form is not only intuitively appealing, but it may also allow for us to ultimately appeal to the control research community. State-space models of this form offer the building blocks, however primitive, to parameterize more elaborate control system architectures that include PI controllers, use state feedback, or use even more sophisticated techniques from control theory. Furthermore, enforcing these constraints during learning implicitly reduces the dimension of the parameter space.

We can estimate the natural frequency by making an assumption of $e(t)$ to be represented by a zero-mean stationary Gaussian random process. In this case, we can use Rice's formula for the level-upcrossing rate [15] [27], as shown in Eqn. 10, to compute the natural frequency, $\omega_n = \frac{\sigma_{\dot{e}}}{\sigma_e}$. This formula can be derived very easily [20], and is used in similar studies [8][9][21].

$$
\begin{aligned}
\mathbf{A_c} &= \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \quad (7) \\
\mathbf{\Gamma_c} &= \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} \quad (8) \\
\mathbf{C_c} &= \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (9) \\
\nu_e^+ &= \frac{\sigma_{\dot{e}}}{2\pi\sigma_e} e^{-\frac{1}{2}\left(\frac{L-\mu_e}{\sigma_e}\right)^2} \quad (10)
\end{aligned}
$$

The number of zero-upcrossings, ν_e^+, of $L = 0$ by the sample data, and the 2^{nd}-order statistics: μ_e, and σ_e can all easily be obtained in order to find ω_n by using Rice's formula. During the learning procedure for the linear dynamic system, the EM algorithm is used to find the parameters shown in Fig. 5. Details of this procedure are provided in Ghahramani and Hinton [11] as well as Digalakis et. al. [7], and it is implemented using Murphy's BNT (Bayes' Net Toolbox) [24].

A. Initialization and Learning Strategies

One method of initialization/learning is to set $\zeta = 1$ and "clamp" ω_n during training. Initial values for $\mathbf{A}_c$ and $\mathbf{\Gamma_c}$ can then be derived as a function of ζ and ω_n. $\mathbf{C_c} = \begin{bmatrix} 1 & 0 \end{bmatrix}$ is also fixed during learning, and R is initialized by making a random guess at the SNR (signal to noise ratio), so that $R = \frac{\sigma_e^2}{SNR}$.

$$
\begin{aligned}
\mathbf{A}_c\mathbf{X}_{ss} + \mathbf{X}_{ss}\mathbf{A}_c^T &= -\mathbf{\Gamma}_c\mathbf{\Gamma}_c^T \quad (11) \\
\mathbf{Q}_c &= \frac{\sigma_e^2 - R_c}{\mathbf{C_c}\mathbf{X}_{ss}\mathbf{C_c}^T} \quad (12)
\end{aligned}
$$

Using these assumptions, we apply the steady-state continuous-time Lyapunov equation (w/ solution $\mathbf{X}_{ss}$) in order

[1] All coefficients are subscripted by c for "continuous" in order to disambiguate between this and the unsubscripted discrete analogue shown in Eqns. 1-2.

to find an adequate initialization for $\mathbf{Q}_c$, as is performed in [20],[21], and shown in Eqns. 11-12. We then discretize all parameters using the sampling interval T_s using the procedure outlined in [21], allowing us to form Eqns. 1 - 2 (details omitted for clarity). Furthermore, we use the solution of the discrete algebraic Lyapunov equation (Eqn. 13) as an initialization for $\mathbf{P}_0$, and initialize $\mu_{\mathbf{x}} = 0$, After learning, we can then back out the learned value of the damping ratio ζ and the signal to noise ratio.

$$\mathbf{P}_0 = \mathbf{P}_{ss} = \mathbf{A}\mathbf{P}_{ss}\mathbf{A}^T + \mathbf{Q} \tag{13}$$

Another initialization/learning technique involves the same initialization strategy, but relaxation of the "clamping" of ω_n during training. In this way the value of ω_n can also be learned in addition to the damping ratio ζ, and the signal to noise ratio. However, we still enforce the canonical form constraint throughout the learning process. Finally, we will investigate a constraint-free learning process, although the initialization strategy remains identical to the two previous cases, and enforcement of the canonical form constraint is applied after the learning process.

For comparison, two alternative initialization strategies will be tested as well, one of them which is least desirable of all, that being a semi-random approach. Here we will randomly initialize $\mathbf{A}$, $\mathbf{C}$, and $\mathbf{Q}$, using Eqn. 13 again to find $\mathbf{P}_0$. In order to ensure that $\mathbf{Q}$ is a Hermitian matrix, we apply the following transformation: $\frac{\mathbf{Q}+\mathbf{Q}^T}{2}$. Finally, we make a random guess at the SNR (signal to noise ratio) in order to choose R, and initialize $\mu_{\mathbf{x}} = 0$, as before.

Factor analysis is the final alternate initialization strategy to be tested. To use an analogy from within the probabilistic graphical modeling paradigm, factor analysis is to LDS (or Kalman filter) as the GMM is to the HMM (Hidden Markov Model) [14]. That is, the main assumption shared by the GMM and factor analysis is the lack of causal conditional dependencies among the hidden variables. In the GMM/HMM domain the hidden variables are discrete/multimonial random variables, and in the factor analysis/linear dynamic system domain the hidden variables all have continuous Gaussian distributions. As such, the same machinery for learning the parameters (the EM algorithm) is used for factor analysis, which provides us with an informed set of initialization parameters, and has been performed in relevant machine learning venues [12]. Table IV summarizes and assigns case numbers to all initialization and learning strategies discussed thus far and to be investigated in the subsequent section.

We will also quantify the levels of suboptimality introduced by using these various initialization schemes in the subsequent section. We enforce the constraints during learning, or after learning so that the model structure will adhere to the canonical form. A more technically sound approach to initialization may be to apply a random perturbation technique of the kind often seen in algorithms such as stochastic local search [23]. A more technically sound approach to learning would be to derive a modified M-step so that the parametric updates are an explicit function of desired parameters (the intuitive canonical parameters such as SNR, ω_n, and ζ, and potentially even control gains).

The only other research that addresses a similar control theoretic approach to using probabilistic graphical modeling is Deventer et al. [6] and Deventer [5], whose work will also be considered for comparison in future studies. However, in Deventer's work, controller design is part of the research problem. We will assume that the controller has already been designed and attempt to learn its dynamical structure and that of the plant. All of these methods will be investigated in future studies.

Furthermore, all of the initialization and learning strategies discussed thus far consider only the control system error as the primary data source. If we alternatively use the transformation of the Orca or SVM score as the data source, we may use a context free initialization strategy such as factor analysis, so as not to cast the data in a particular domain (i.e. intuitive canonical parameters for control system error). We may also lift all restrictions during training, and have more flexibility with the order of the model. As such, we will consider model orders ranging from $n = 2$ to $n = 10$ for the Orca and SVM scores, in addition to models trained by using the control system error as a data source.

B. Model Likelihood and Kalman Filter Equations

As of yet, we have not discussed the metric with which we will ascertain model fidelity. As a precursor, let us discuss the log-likelihood function of our model which is maximized during each iteration of the M-step. The likelihood of the data given our model parameters can be expressed as follows:

$$\begin{aligned} p(y_0, \ldots, y_T|\theta) &= \prod_{k=0}^{T} \mathcal{N}(\varepsilon_k; 0, \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + R) && (14) \\ \varepsilon_k &\triangleq y_k - \hat{y}_{k|k-1} && (15) \end{aligned}$$

where T is the total number of observed samples, and ε_k in Eqn. 15 is the white noise innovation process. Other definitions are provided below.

$$\begin{aligned} \hat{\mathbf{x}}_{k|k} &\triangleq E[\mathbf{x}_k|y_0, \ldots, y_k] \\ \mathbf{P}_{k|k} &\triangleq E[(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})(\mathbf{x}_k - \hat{\mathbf{x}}_{k|k})^T|y_0, \ldots, y_k] \end{aligned}$$

Furthermore,

$$\begin{aligned} \hat{y}_{k|k} &= \mathbf{C}\hat{\mathbf{x}}_{k|k} && (16) \\ \hat{\mathbf{x}}_{k+1|k} &= \mathbf{A}\hat{\mathbf{x}}_{k|k} && (17) \\ \mathbf{F}_{k+1|k} &\triangleq \mathbf{P}_{k+1|k}\mathbf{C}^T(\mathbf{C}\mathbf{P}_{k+1|k}\mathbf{C}^T + R)^{-1} && (18) \\ \mathbf{P}_{k+1|k} &= \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}^T + \mathbf{Q} && (19) \\ \mathbf{P}_{k+1|k+1} &= \mathbf{P}_{k+1|k} - \mathbf{F}_{k+1|k}\mathbf{C}\mathbf{P}_{k+1|k} && (20) \end{aligned}$$

Eqn. 18 represents the dynamically updated Kalman gain, and combining the two equations 19 and 20, we may obtain the following:

TABLE IV
INITIALIZATION AND LEARNING STRATEGIES FOR CONTROL SYSTEM ERROR

Case Label	Training Constraints	Parameter Clamping	Initialization Type
Case #1	Canonical	Natural Frequency	Data-Driven, Canonical
Case #2	Canonical	None	Data-Driven, Canonical
Case #3	None	None	Data-Driven, Canonical
Case #4	None	None	Semi-Random
Case #5	None	None	Factor Analysis

$$\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_{k|k-1}\mathbf{A}^T - \mathbf{A}\mathbf{F}_{k|k-1}\mathbf{C}\mathbf{P}_{k|k-1}\mathbf{A}^T + \mathbf{Q} \quad (21)$$

Thus far, all equations have been introduced under the assumption that y_k is zero mean process, without loss of generality. This was allowed for the sake of mathematical convenience. We must make a distinction between the control system error and the SVM or Orca composite anomaly scores providing the basis for the training dataset. The control system error is close enough to a zero mean process qualitatively and quantitatively to allow for the mathematical representation introduced thus far to be used (cf. Fig. 3). However, when using the SVM or Orca composite anomaly scores as the basis for the dataset, the zero-mean assumption clearly fails, as evidenced in Fig. 4. As such, we will briefly highlight how to use the current mathematical formulation without loss of generality. There are a number of ways to handle data with a non-zero mean, however in our case we will add a term to the state and output equations as shown in Eqns. 22- 23.

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}u_k + \mathbf{w}_k & (22) \\ y_k &= \mathbf{C}\mathbf{x}_k + v_k & (23) \end{aligned}$$

where

$$\begin{aligned} \mathbf{B} &= \begin{bmatrix} 1 \\ \mathbf{0}_{(n-1)\times 1} \end{bmatrix} \\ u_k &= u_{ss}, \quad \forall k \in 1, \ldots, T \\ &= \frac{\mu_y}{\mathbf{C}(\mathbf{I}_n - \mathbf{A})^{-1}\mathbf{B}} \end{aligned}$$

We assume $u_k \in \mathbb{R}$ is a scalar whose steady-state value can easily be determined by the use of Eqn. 22, and $\mathbf{B}$ is chosen to be a fixed coefficient out of convenience. This is practical because $\mathbf{B}$ is not updated during the learning process, and as such does not require clamping which might otherwise introduce suboptimality. μ_y is empirically determined from the training data set, and is used for validation as well. Propagating this extra term through the Kalman filter will result in the updates shown in Eqn. 24

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}\hat{\mathbf{x}}_{k|k} + \mathbf{B}u_k \quad (24)$$

Alternatively, we could have introduced an additive constant to Eqn. 23 in order to account for the non-zero mean. However, the formulation shown above allows us to appeal to supervised learning problems or dynamic systems which may actually require the use of an driving input that changes with time, as is the case in control theory.

C. Akaike Information Criterion

An expression for the log-likelihood function follows easily from [10], shown in Eqn. 26. The more general expression is proved in [16], and the expression shown in Eqn. 26 is derived by using the assumptions and notation introduced thus far, in addition to Eqn. 25.

$$\sigma_k \triangleq \mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^T + R \quad (25)$$

$$\log p(y_0, \ldots, y_T|\theta) = -\frac{T}{2}\log 2\pi - \sum_{k=0}^{T} \log \sigma_k^2 + (\varepsilon_k \sigma_k)^2 \quad (26)$$

However, it is well known that even though this is implicitly the objective function of the MLE (Maximum Likelihood Estimation) problem performed iteratively during the M-step of the EM learning algorithm[2], we cannot use this as an unbiased indicator for the assessment of model fidelity [30]. The form of the log-likelihood function itself would otherwise seem to indicate some measure of how well the data $(y_0, \ldots, y_T)$ fit the model (θ). The "log" part of the log-likelihood function is introduced for mathematical convenience and has no bearing on the result of the MLE problem since it is a monotonic operator.

The bias stems from the fact that the learned model parameters that comprise θ are used to obtain the value for the log-likelihood. Intuitively, any metric used to assess how well the data fit the model should not contain any parameters used to train that model. The bias introduced can also be derived with more mathematical rigor, using a more precise definition. However, we will forgo those derivations here and speak about bias in a more qualitative sense.

This bias can be accounted for by using the Akaike information criterion (AIC), which is based upon the Kullback-Leibler (KL) divergence. The KL divergence is a measure of the distance between the modeled distribution and the true distribution. Therefore, the AIC is often used to guide model selection due to its inherent capability to assess the fit of the data to the model based upon the number of model parameters. As such, it can be use for the assessment of model fidelity.

The biased term shown in Eqn. 27 acts as a proxy for the precise definition of KL divergence. The AIC adjusts for the

[2] More precisely, the expected complete log-likelihood function is the objective function under consideration for MLE, in place of the log-likelihood function. This involves the use of sufficient statistics represented as the expected value of the hidden variables $\mathbf{P}_{k|k}$ and $\hat{\mathbf{x}}_{k|k}$ computed during the E-step.

inherent bias by adding a bias correction term as shown in Eqn. 28, which is an approximation of the *expected* difference between the KL divergence and the computed bias term. The AIC defined in Eqn. 29 is the sum of the biased term and the bias correction term, where θ_0 represents the parameters of the true model.

$$\begin{aligned}
\mathcal{T}_b(y_0,\dots,y_T,\theta) &= -2\log p(y_0,\dots,y_T|\theta) && (27)\\
\mathcal{T}_c(\theta_0) &\approx 2|\theta| && (28)\\
\text{AIC} &= \mathcal{T}_b(y_0,\dots,y_T,\theta)+\mathcal{T}_c(\theta_0) && (29)
\end{aligned}$$

We will explore the use of a method recently introduced by [2] which is a variant of the AIC explicitly derived for state-space models, denoted AICi. The AICi criteria is a revised and improved approximation to the bias correction term, which is robust for small sample settings. It also tends to report less deviation from the true model when the trained model is overparameterized. Because the bias correction term represents the *expected* difference between the KL (Kullback-Leibler) divergence and the computed biased term, we can estimate it via Monte Carlo simulation. We can therefore replace Eqn. 28 with Eqn. 30 which uses an ensemble average of this expected difference by using a convenient sampling distribution.

The sampling distribution, $y_k \sim \mathcal{N}(0,1)$ is used to generate M distinct T-sample training cases, and M independently sampled additional T-sample test cases. Essentially, there are $2M$ sets of data generated. The first M sets of data are used as training data to generate corresponding sets of new model parameters, and the second M sets of data are used as test cases. The constant coefficient, $\mathbf{C}(j)$, shown indexed by j corresponds to the j^{th} model generated by the j^{th} set of training data. $\mathbf{P}_{k|k}(j)$ and $\hat{x}_{k|k-1}(j)$ correspond to the j^{th} set of test data using the j^{th} model.

$$\begin{aligned}
\mathcal{T}_c(\theta_0) \approx\; & -T+\frac{1}{M}\sum_{j=1}^{M}\left(\sum_{k=0}^{T}\text{trace}\left(\mathbf{P}^{-1}_{k|k}(j)\right)+\dots\right. && (30)\\
& \left.\sum_{k=0}^{T}\hat{\mathbf{x}}^T_{k|k-1}(j)\mathbf{C}^T(j)\mathbf{P}^{-1}_{k|k}(j)\mathbf{C}(j)\hat{\mathbf{x}}_{k|k-1}(j)\right)
\end{aligned}$$

Because the AIC and AICi criteria both provide a measure of disparity between the true and fitted model, we would like to minimize either metric to as low a value as possible. In the next section we will evaluate both the quantitative and computational differences between using both methods and use the method that demonstrates the best performance with respect to these requirements. As such, we will also provide an interpretation of the results for the chosen metric using all initialization and learning strategies discussed thus far, and for both training and validation sets shown in Table II.

IV. Results

We begin the presentation of our results for the case in which we train on control system error. It is used both for training and validation based upon the data shown in Table II.

TABLE V
AIC Results

Case Label	AIC			
	Initial	**Final**	**Adjusted**	**Validation**
Case #1	27432	27349	27349	57805
Case #2	1046211	27084	27084	57743
Case #3	27432	26954	27341	57444
Case #4	43421	26957	30522	57301
Case #5	27440	26954	28478	57198

The model order in this case is restricted to $n = 2$ in order to allow for adherence to canonical form.

Thorough testing was performed on the quantitative and computational differences between the AIC and AICi metrics. It was found that the resulting AICi metric values were on par with the AIC values, but at an added computational burden of 1615:1 when using $M = 10$ as the number of training and test cases to use for the ensemble average shown in Eqn. 30. Case # 1 was used for testing purposes, and on average the AICi values were fractions of a percent different from the values listed on the first row in Table V for AIC. This may inherently be due to the modest dimension of the parameter space ($n = 2 \Rightarrow |\theta| = 17$), and due to the fact that the approximation for the AIC criterion is sufficient due to the large sample size ($T = 6507$). Furthermore, the same canonical parameters resulted when using both methods, corresponding to the first row of Table VI. As such, unless otherwise stated, we will apply the AIC metric for our measure of model fidelity.

In Table V, we see that the lowest AIC values are the learned models (in the "final" column) for Cases #3 and #5. This corresponds to unconstrained learning, using data-driven canonical initialization and factor analysis-based initialization, respectively. Intuitively, this result makes sense due to the freedom from constraints, and the use informative initialization schemes. The final AIC score for Case #4 in which the less informative semi-random initialization was used does not yield a much different result than Cases #3 and #5. However, it does appear that the AIC score for Case #4 prior to training was well above the other cases, with the notable exception of Case #2.

Case #2 requires adherence to the canonical form during training, however neither of the canonical parameters are clamped, as opposed to Case #1 in which only ω_n was required to be clamped. As such, the same initialization strategy used for Case #1 failed due to non-convergence of the learning process. Therefore an alternate set of parameters as shown in the corresponding row in Table VI was used. Table VI also illustrates that the damping ratio and natural frequency values are often pathological (i.e. Cases #4 and #5). However, realizations of the resulting linear dynamic system suffice to qualitatively represent the training data for the control system error. At any rate, a coarse grid search was used to find the parameters corresponding to Case #2, which yielded a successfully convergent learning regime.

The initial AIC score shown in Table V for Case #3 was

TABLE VI
CANONICAL PARAMETER RESULTS

	Natural Frequency (ω_n)		**Damping Ratio (ζ)**		**Signal-to-Noise Ratio (SNR)**	
Case Label	**Initial**	**Final**	**Initial**	**Final**	**Initial**	**Final**
Case #1	34.06	34.06	1	0.614	100	72.2
Case #2	34.06	94.79	0.001	1.48	10000	128.34
Case #3	34.06	37.38	1	2.04	100	59.2
Case #4	34.06	0.005	1	21769	100	57.78
Case #5	34.06	0.073	1	216	1.56	42.02

hence understandably very high due to the extreme parametric settings required for convergence. However, the final AIC score obtained after the learning process were on the same order of magnitude as for the remaining cases. The final AIC scores were higher for Cases #1 and #2 that imposed constraints during learning, and understandably highest for Case #1, in which the natural frequency parameter, ω_n, was clamped.

Table V also lists the AIC scores for canonical adjustments that occur after training, in the fourth column labelled "Adjusted." For Cases #1 and #2, in which the constraints were applied throughout the learning process, the AIC score clearly does not change. However, for the remaining Cases #3 - #5, a modest level of suboptimality is introduced, as evidenced by the higher AIC scores. Intuitively, even though no constraints were placed on the form of the parameters during training in Case #3, an adjustment to enforce the canonical constraint introduces less suboptimality than the other Cases # 4 and #5 since they were not initialized in canonical form.

Therefore, it is clear that if a strong requirement for the restriction to canonical form exists, the best method to use is Case #2. This case actually enforces the canonical form during training, but does not clamp any of the parameters. However, if there is no strong requirement for adherence to the canonical form, the cases that do not enforce constraints during training, specifically Case # 3 and #5, which do not use random initialization demonstrate superior performance.

The final column in Table V show the AIC score for the validation data as applied to either the final or "adjusted" model, whichever has the lowest AIC score. As might be expected, the validation scores are much higher than the learned AIC scores due to the fact that more than half of the validation data contain anomalies (cf. Table I). However, it is evident that the AIC score decreases as the case number increases.

Case #1 is the most restricted, in which both clamping and constraints are enforced during learning, leading to the highest reported validation score. Case #2 does not clamp parameters during learning, although the canonical constraint is enforced, resulting in a slightly lower reported AIC score for the validation data. The remaining cases are constraint-free during the learning process, with Case #5 demonstrating the most favorable AIC score for the validation data. As such, we will use factor analysis as the default initialization method to study the serial architecture. This involves using the SVM and Orca composite anomaly scores as training and validation data, in addition to control system error for comparison. The results will be presented graphically as a function of model order.

We conclude this section with a discussion of the results for the cases related to the use of the serial architecture. One of the most ubiquitous observations in this part of the investigation is that the learning phase often fails, due to matrix singularities that arise when using model orders above various thresholds for different techniques. This occurs even when using factor analysis for initialization, which is a far superior method than randomization. Using the "data-driven canonical" approach to initialization is valid only for a model order of $n = 2$ for the framework chosen in this paper. However, it is certainly possible to augment the state-space in canonical form to allow for a more generic parametric representation that is less intuitively appealing. Alternatively, we may even explore the use of a linearized physics-based model that is defined as a function of physical parameters for the prior distribution (i.e. initialization). Hence, we can implicitly integrate physics-based and data-driven methods in a Bayesian context, while augmenting our model order in an informative manner.

Using control system error and SVM anomaly scores for the training data, the learning phase failed to converge for model orders of $n = 4$ or higher, and for model orders of $n = 3$ or higher when using Orca anomaly scores for the training data. This may imply overparametrization, or a fundamental identification of the maximum specified model order, which is not known apriori. However, in order to provide a sanity check on these results, we may use the AICi criterion in addition to the AIC criterion. This is done in order to account for the fact that the AIC criterion often yields a biased lower score for models that are overparameterized due to higher model orders. However, the model learning part of the procedure outlined in association with Eqn. 30 failed. This was due again to matrix singularities that arose when using higher model orders. For the lowest model orders that did converge, the AICi criteria did serve to corroborate that the AIC criteria was fairly unbiased and accurate. Again, on average the AICi values were fractions of a percent different from the AIC values.

Even though certain models cannot be successfully trained for all model orders, we may still investigate their fidelity. Both training and validation data can be used to generate AIC scores that are shown as function of model order. The model based upon the initialization using factor analysis can easily be used to compute the AIC for both training and validation data. The results are shown in Figs. 6 - 8, for all three cases illustrated

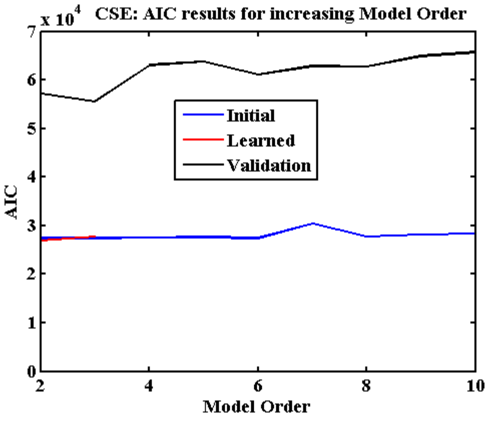

Fig. 6. AIC Result for Control System Error

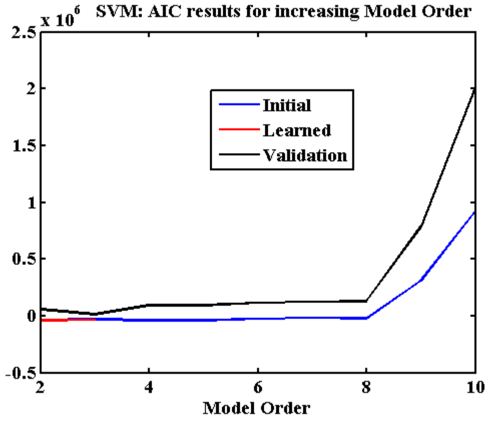

Fig. 7. AIC Result for SVM

in the serial architecture. Where possible, the AIC score is also shown for models with lower orders that successfully completed the training phase.

As seen in Fig. 6, the validation AIC score is nearly double that of the training AIC score for all model orders shown, as might be expected. Also, there is a slight rise in both training and validation AIC scores with model order. This may be harder to discern for the training AIC score due to an outlier at $n = 7$, however, there is a general upward trend which meets with intuition. Furthermore, as previously alluded to, the AIC score is shown for the models with lower orders $(n = 2, 3)$ that successfully completed the training phase.

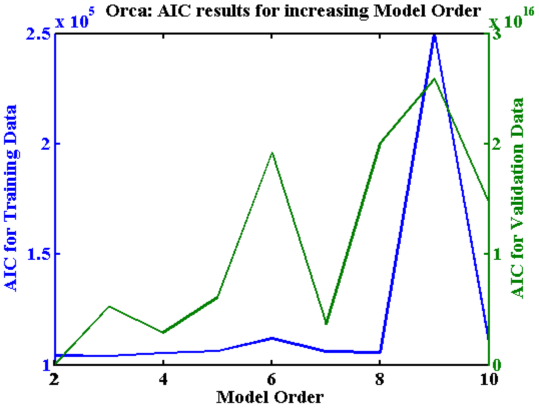

Fig. 8. AIC Result for Orca

In comparing these results to those shown in Fig. 7, we first draw the reader's attention to the discrepancies between the scales of both Fig.6 and 7. In general, the AIC scores are much more extreme for the SVM-based AIC scores shown in Fig. 7 than in the control system error-based AIC scores shown in Fig. 6. The AIC score for the training data in Fig. 7 rises from a very small, even negative AIC score, to one that is much higher than in Fig. 6 for higher models order. This exaggerated rise in AIC with model order speaks to the idiosyncracies of fitting an overparameterized LDS model to the SVM score. A similar characteristic is evident for the validation data, however, for lower model orders the AIC score is a much higher positive value than was found for the training data. In fact, for $n = 2$, the AIC scores based on the validation SVM data was on par with the validation control system error data, indicating a similar fit. In this case it may be more prudent to choose the SVM model using the serial architecture due to the limitation of using control system error cited earlier.

The results of using Orca in the serial architecture are shown in Fig. 8. Due to the drastic difference between the scales of the AIC scores for training and validation data, the data labels are segregated and shown opposite of each other (training in the left in blue, and validation on the right in green). It is clear that the AIC scores exhibited here are orders of magnitude greater than for either of the previous cases shown in Figs. 6-7. Neither exhibits the expected monotonic increase in AIC with model order. All of these observations are consistent with the fact that the model is not fit well to the data. As such, we can certainly look more at this issue in future work using the strategies outlined previously that are shared with the SVM approach.

The lowest validation AIC score out of all models tested was for the SVM composite anomaly score, when using a model order of $n = 3$, having a value of 13473. As such, this should certainly be a candidate for application to subsequent anomaly detection studies. This should be preceded by a more thorough examination of the subtleties for any potential bias of the reported AIC score by using AICi for corroboration if possible. Furthermore, remedies to the matrix singularity issue need to be investigated in more depth, as well as other issues introduced earlier. One such example is to exploit the fact that the SVM score is a distance-based metric in order to achieve a better qualitative fit of the data to the model.

V. Conclusions

In this paper, we have examined the model fidelity of several competing data transformation techniques as measured by the AIC (Akaike information criterion). When using control system error as the sole source of data, we have found that the use of canonical constraints during training decreases model fidelity. However, if the canonical constraint is deemed compulsory, no clamping should be used in order to achieve the lowest AIC score. Furthermore, when using the canonical constraint in future studies, suboptimality may be avoided by deriving a modified M-step during learning. Lifting the canonical requirement implies constraint free learning, and the use of either data-driven initialization or initialization based upon factor analysis yields the best model fidelity.

We have also found that the SVM composite anomaly score yields the lowest validation AIC score, as is such a candidate for future study for application to subsequent anomaly detection studies. Part of this study should include the investigation of matrix singularities that appear during learning when initialized by factor analysis. This may be remedied by using more principled initialization techniques such as stochastic local search.

Finally, by using the serial architecture in lieu of the control system error alone, we are implicitly reducing the entire feature space into a univariate signal while retaining salient operational signatures. This is potentially a far more effective approach than using only a small fraction of the feature space by using the control system error alone. As such we may potentially allow for many more anomalies to be detected by using this paradigm.

ACKNOWLEDGMENT

The author would like to extend thanks to John Wallerius and Nikunj Oza for their help in reviewing this paper, and to various NASA funding sources for supporting this research.

REFERENCES

[1] Stephen D. Bay and Mark Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *KDD '03: Proceedings of The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 29–38, New York, NY, 2003. ACM Press.

[2] Thomas Bengtsson and Joseph E. Cavanaugh. An improved Akaike information criterion for state-space model selection. *Computational Statistics and Data Analysis*, 50(10):2635–2654, 2006.

[3] James C. Boyd and David A. Lacher. A multi-stage gaussian transformation algorithm for clinical laboratory data. *Clinical Chemistry*, 28(8):1735–1741, 1982.

[4] Gilles Cohen, Melanie Hilario, and Christian Pellegrini. One-class support vector machines with a conformal kernel. a case study in handling class imbalance. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 850–858, 2004.

[5] Rainer Deventer. *Modeling and Control of Static and Dynamic Systems with Bayesian Networks*. PhD thesis, Universität Erlangen–Nürnberg, 2004.

[6] Rainer Deventer, Joachim Denzler, and Heinrich Niemann. Control of Dynamic Systems Using Bayesian Networks. In Leliane Nunes de Barros et. al, editor, *Proceedings of the IBERAMIA/SBIA 2000 Workshops*, pages 33–39, Atibaia, São Paulo, Brazil, 2000. Tec Art Editora, São Paulo.

[7] Vassilios V. Digalakis, Jan Robin Rohlicek, and Mari Ostendorf. ML estimation of a stochastic linear system with the EM algorithm and its application to speech recognition. *IEEE Transactions on Speech and Audio Processing*, 1(4):431–442, 1993.

[8] Clifford C. Federspiel, Rodney A. Martin, and Hannah Yan. Thermal comfort models and "call-out" (complaint) frequencies. Technical report, University of California, Berkeley, Center for the Built Environment, 2003.

[9] Clifford C. Federspiel, Rodney A. Martin, and Hannah Yan. Recalibration of the complaint prediction model. *International Journal of HVAC&R Research*, 10(2), April 2004.

[10] Joe Frankel. *Linear dynamic models for automatic speech recognition*. PhD thesis, University of Edinburgh, 2003.

[11] Zoubin Ghahramani and Geoffrey E. Hinton. Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, Department of Computer Science, University of Toronto, 1996.

[12] Zoubin Ghahramani and Sam Roweis. Learning nonlinear dynamical systems using an EM algorithm, 1999.

[13] David L. Iverson. Inductive system health monitoring. In *Proceedings of The 2004 International Conference on Artificial Intelligence (IC-AI04)*, Las Vegas, Nevada, June 2004. CSREA Press.

[14] Michael I. Jordan. An introduction to probabilistic graphical models. Manuscript used for Class Notes of CS281A at UC Berkeley, Fall 2002.

[15] Benjamin Kedem. *Time Series Analysis by Higher Order Crossings*. IEEE Press, 1994.

[16] Helmut Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, 2006.

[17] Rodney Martin. Investigation of optimal alarm system performance for anomaly detection. In *National Science Foundation Symposium on Next Generation of Data Mining and Cyber-Enabled Discovery for Innovation*, Baltimore, MD, October 2007.

[18] Rodney Martin. Unsupervised anomaly detection and diagnosis for liquid rocket engine propulsion. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, March 2007.

[19] Rodney Martin, Mark Schwabacher, Nikunj Oza, and Ashok Srivastava. Comparison of unsupervised anomaly detection methods for systems health management using space shuttle main engine data. In *Proceedings of the 54th Joint Army-Navy-NASA-Air Force Propulsion Meeting*, Denver, CO, May 2007.

[20] Rodney A. Martin. Optimized response to thermal sensation complaints in buildings. Master's thesis, University of California, Berkeley, December 2000.

[21] Rodney A. Martin. *Optimal Prediction, Alarm, and Control in Buildings Using Thermal Sensation Complaints*. PhD thesis, University of California, Berkeley, 2004.

[22] Rodney A. Martin. Approximations of optimal alarm systems for anomaly detection. *IEEE Transactions on Information Theory (preprint)*, 2007.

[23] Ole J. Mengshoel. Understanding the role of noise in stochastic local search: Analysis and experiments. *Artificial Intelligence*, 172(8–9):955–990, 2008.

[24] Kevin P. Murphy. The Bayes' Net Toolbox for MATLAB. *Computing Science and Statistics*, 33, 2001.

[25] Han Park, Ryan Mackey, Mark James, Michail Zak, Michael Zynard, John Sebghati, and William Greene. Analysis of Space Shuttle Main Engine Data Using Beacon-based Exception Analysis for Multi-Missions. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, March 2002.

[26] Ronald K. Pearson. *Discrete-time Dynamic Models*. Oxford University Press, USA, December 1999.

[27] S. O. Rice. Mathematical analysis of random noise. *Bell System Technology Journal*, 24:46–156, 1945.

[28] Mark Schwabacher. Machine learning for rocket propulsion health monitoring. In *Proceedings of the SAE World Aerospace Congress*, volume 114-1, pages 1192–1197, Dallas, Texas, 2005. Society of Automotive Engineers.

[29] Mark Schwabacher and Robert Waterman. Pre-launch diagnostics for launch vehicles. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, March 2008.

[30] Neil Thacker. Tutorial: Beyond Likelihood. Technical report, Tina Internal Memo No. 2005-008, Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, March 2007.