AIAA Infotech@Aerospace 2010
20 - 22 April 2010, Atlanta, Georgia

AIAA 2010-3517

Near Real-Time Optimal Prediction of Adverse Events in Aviation Data

Rodney A. Martin* and Santanu Das†

The prediction of anomalies or adverse events is a challenging task, and there are a variety of methods which can be used to address the problem. In this paper, we demonstrate how to recast the anomaly prediction problem into a form whose solution is accessible as a level-crossing prediction problem. The level-crossing prediction problem has an elegant, optimal, yet untested solution under certain technical constraints, and only when the appropriate modeling assumptions are made. As such, we will thoroughly investigate the resilience of these modeling assumptions, and show how they affect final performance. Finally, the predictive capability of this method will be assessed by quantitative means, using both validation and test data containing anomalies or adverse events from real aviation data sets that have previously been identified as operationally significant by domain experts. It will be shown that the formulation proposed yields a lower false alarm rate on average than competing methods based on similarly advanced concepts, and a higher correct detection rate than a standard method based upon exceedances that is commonly used for prediction.

Nomenclature

k	Time index
$\|\cdot\|$	Euclidean norm
$\succeq$	Positive semi-definite
$\mathcal{I}$	Universe of all possible events
$(\cdot)'$	Not (Set complement)
$(\cdot)^\top$	Transpose
$P(\cdot)$	Probability
$E[\cdot]$	Expected Value
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean μ and covariance Σ
$\mathcal{N}(\mathbf{x}; \mu, \Sigma)$	Gaussian distribution evaluated at $\mathbf{x}$ with mean μ and covariance Σ

I. Introduction

DEVELOPING an automated anomaly prediction capability in near-real time for aircraft systems or subsystems is of great importance for the future of aviation safety. It is also important that algorithms providing this predictive capability are capable of isolating the anomalies to specific components or sensors, while providing an algorithmic explanation of the reasons why the anomalies were flagged. One of the objectives we plan to achieve in the process of developing such an algorithm is to allow for the prediction to occur within a 2 second time horizon of an actual adverse event, with a false positive rate less than 5%. The motivation for these specific objectives stems from the need to establish a minimum required time for the crew to respond to a critical event with a high level of confidence, in part driven by a study performed

*Computer Engineer, NASA Ames Research Center, Mail Stop 269-1, Bldg. N-269, Rm. 260-17, P.O. Box 1, Moffett Field, CA 94035-0001.

†Associate Scientist, University Affiliated Research Center, NASA Ames Research Center, Mail Stop 269-2, Bldg. N-269, Rm. 163, P.O. Box 1, Moffett Field, CA 94035-0001.

by Hayden *et al.*[1] The operational significance and meaning of the adverse events will be assessed by domain experts, but can also be assisted by appropriate tools such as limit checks and other well established universally accepted aviation rules.

These types of specifications fit very neatly within a framework introduced in previous work,[2] where we developed the theoretical basis for a method of optimal level-crossing prediction, enabled by Kalman filtering. Optimality is realized here by providing an upper bound on the false alarm probability for a fixed detection probability, and over a fixed prediction horizon. Therefore, it is our aim in this paper to document our initial steps towards achieving the goal of developing a new, state-of-the-art forecasting technology by using these newly conceived theoretical ideas. Our hope is to demonstrate the plausibility of these novel techniques by conducting a preliminary investigation using real aviation data. In previous optimal level-crossing prediction research, we studied a "two-sided" level crossing event that spans a fixed prediction horizon and exceeds upper and lower predefined critical thresholds symmetric about the mean of a stationary linear Gaussian process many times during the timeframe. The two-sided case is practically relevant when monitoring residuals that may be derived from the output of other machine learning algorithms or transformed parameters that relate to system performance. This last point is of paramount importance in the use of this approach. We may connect multiple models in series, each of which is developed with the aid of machine learning techniques in an architecture of the type introduced in previous work (*cf.* Martin[3]). Fig. 1 below illustrates this concept.

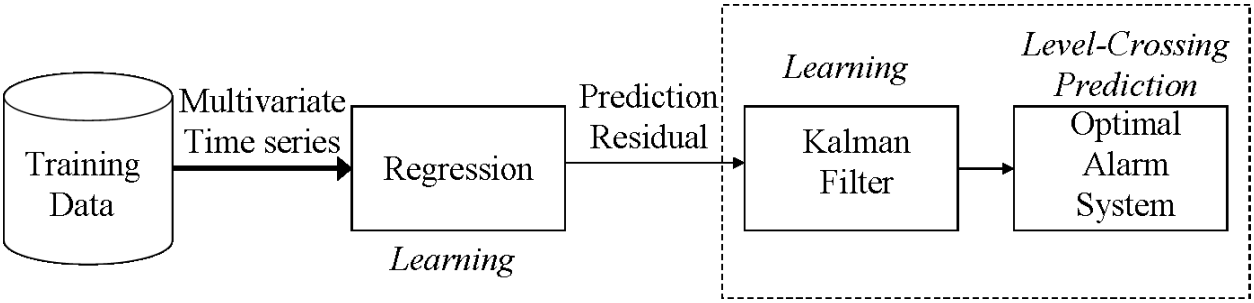


Figure 1. Proposed Functional Architecture

In this investigation we will test the theoretical assumptions made by appealing to the use of the optimal level-crossing predictor and other standard unsupervised machine learning techniques used to parameterize the underlying models shown in Fig. 1. Here we will employ real aviation data sets, where independent time series each represent individual flights. Selected sets of flights from a single aircraft will be chosen to develop a model, and two other distinct sets of flights from this same aircraft will be used for validation and testing purposes. We appeal to the idea of "boosting" in order to allow for the use of the residual generated from the base model by the Kalman filter. In Fig. 1, the regression block represents the base model, which processes a select number of parameters (the "multivariate time series") and maps them to a distinct target parameter. The residual output from this block quantifies the difference between the actual value of the pre-specified target parameter and the value predicted by the base model. Thus, implicitly the use of this architecture extends the domain of this work to systems producing multivariate data, rather than only univariate data as was introduced in the seminal article for this topic. This is primarily implemented with the use of a "base model" to preprocess the multivariate time series, resulting in a univariate output.

Ostensibly, this mapping should have a functional basis specific to the safe operation of the aircraft, for which any reasonably robust machine learning approach can be used (*e.g.* linear regression, quadratic regression, Gaussian process regression, nonlinear kernel-based regression such as is found in MSET[4] (Multivariate State Estimation Technique), bagged neural nets, *etc.*). However, for the purposes of our study, we will use support vector regression (SVR)[5] to provide this mapping. Furthermore, a single target parameter that acts as a global health indicator which represents safe operation of the aircraft rarely exists in reality. In fact, there may be multiple such indicators for each adverse event or anomalous operation that is a candidate for prediction. Thus, we may train as many support vector regressors as there are available to characterize adverse events and target parameters. However, in this study we draw from a pool of candidate target parameters and use only a single target parameter for a single adverse event based upon a selected performance objective.

The residual may hypothetically be distributed in such a way that is amenable to modeling as a Gaussian distribution, and can be used as the basis for learning a linear dynamical system, which can subsequently

be used for design of an optimal level-crossing predictor. As such, we will investigate the two-sided level-crossing event in this paper, and also use a Kalman filter-based approach in an optimal manner relevant for the prediction of level-crossings. Note that the optimal level-crossing predictor in its current incarnation requires use of the underlying linear dynamical system model associated with the Kalman filter (evidenced by the arrow leading from the Kalman filter box to the optimal alarm system box within the dotted line shown in Fig. 1).

The rest of this paper will be organized as follows. In Sec. II we will review the architecture for the algorithmic flow related to Fig. 1, as specifically related to signal flow, and tuning for the best possible performance in generation of the results. Data preprocessing and feature selection necessary for application of SVR, as well as the methods used to aid in the identification of prominent features for tuning will also be discussed. In Sec. III we will provide a brief background on support vector regression. In Sec. IV we will review parameter learning for the linear dynamical system, based upon the SVR residuals. In Sec. V we provide a brief background on the concepts of optimal level-crossing prediction. Finally, in Sec. VI we review the tuning process in detail, which will employ the use of validation and testing data sets to be presented and discussed. Furthermore, we will quantify and compare the results of the optimal level-crossing predictor to a competing method based upon the Sequential Probability Ratio Test (SPRT), and a standard method based upon exceedances, both of which are commonly used for prediction. All of the results will then be summarized in a concluding section, Sec. VII.

II. Signal Flow and Algorithmic Tuning

Fig. 1 is actually a *functional* representation of the architecture, meaning that the arrows in the diagram represent batch data transfer, rather than signal flow or real-time signal processing. This paradigm is used to emphasize the fact that learning takes place in stages, due to the serial nature of the architecture. Otherwise stated, the architecture involves not just one, but two distinct algorithms which both involve machine learning, and are necessary to fulfill the desired objective of recasting the anomaly prediction problem into a form whose solution is accessible as a level-crossing prediction problem.

Fig. 2 depicts a more intuitive signal flow representation of the problem. The linear dynamical system (LDS) shown in the dotted portion of the diagram can be thought of as an alternate representation of the SVR block. More explicitly, the LDS is trained to mimic both the spectral and temporal characteristics of the SVR residual, such that $y_k = z_k - \hat{z}_k$, where y_k represents both the SVR residual and the LDS output, and z_k represents a target parameter relevant to safe operation of the aircraft that characterizes a specifically designated adverse event or anomaly. Note that the SVR block takes as input a multivariate vector of features, $\mathbf{u}_k$, and outputs an estimate of this target parameter, $\hat{z}_k$. Note also that the LDS block takes as input the input or process noise, $\mathbf{w}_k$, and measurement noise, v_k. Loosely, we can intuit that the process noise provides a statistical characterization of the multivariate vector of features, $\mathbf{u}_k$, and the measurement noise provides a statistical characterization of the target parameter, z_k.

The Kalman filter block accepts input from the SVR block as y_k, which processes these residuals as observables, and computes state estimates, $\hat{\mathbf{x}}_{k|k}$, based upon these observations during the correction step. These state estimates can then be used to form forward projected residual value predictions, $\hat{y}_{k+i|k}$, spanning the prediction horizon, $\forall i \in \{1, \ldots, d\}$ which are then used by the optimal level-crossing predictor to initiate an alarm with a minimal false alarm probability as dictated by the alarm design parameter, or border probability, P_b. The level-crossing event is parameterized both by L, the critical level, and d, the prediction horizon. The optimal level-crossing event predictor, Kalman filter and value predictor all fundamentally use the LDS parameters, θ, which have been learned during the training process (see Fig. 2.)

Another functional block diagram of interest is related to algorithmic tuning for the best possible performance in generation of the results. Fig. 3 illustrates such a block diagram, in which the ultimate goal of tuning is to optimize model fidelity while at the same time closing the gap between the prediction performance achieved by the training and validation data. Although prediction performance is typically quantified by the AUC (area under the ROC (Receiver Operating Characteristic) curve), it can be misleading to use this metric for optimization, and is best left for evaluation purposes. For anomaly detection techniques that do not involve extensive modeling, such as ones studied in a comparative analysis by Martin,[6] it may be feasible to use the AUC for optimization, however. The main drawback in using the AUC for optimization here is due to the fact that it does not incorporate a measure of model fidelity. It is possible for a model to have a very poor fit to the data, but for the AUC to be very high. Situations such as this are mainly due to

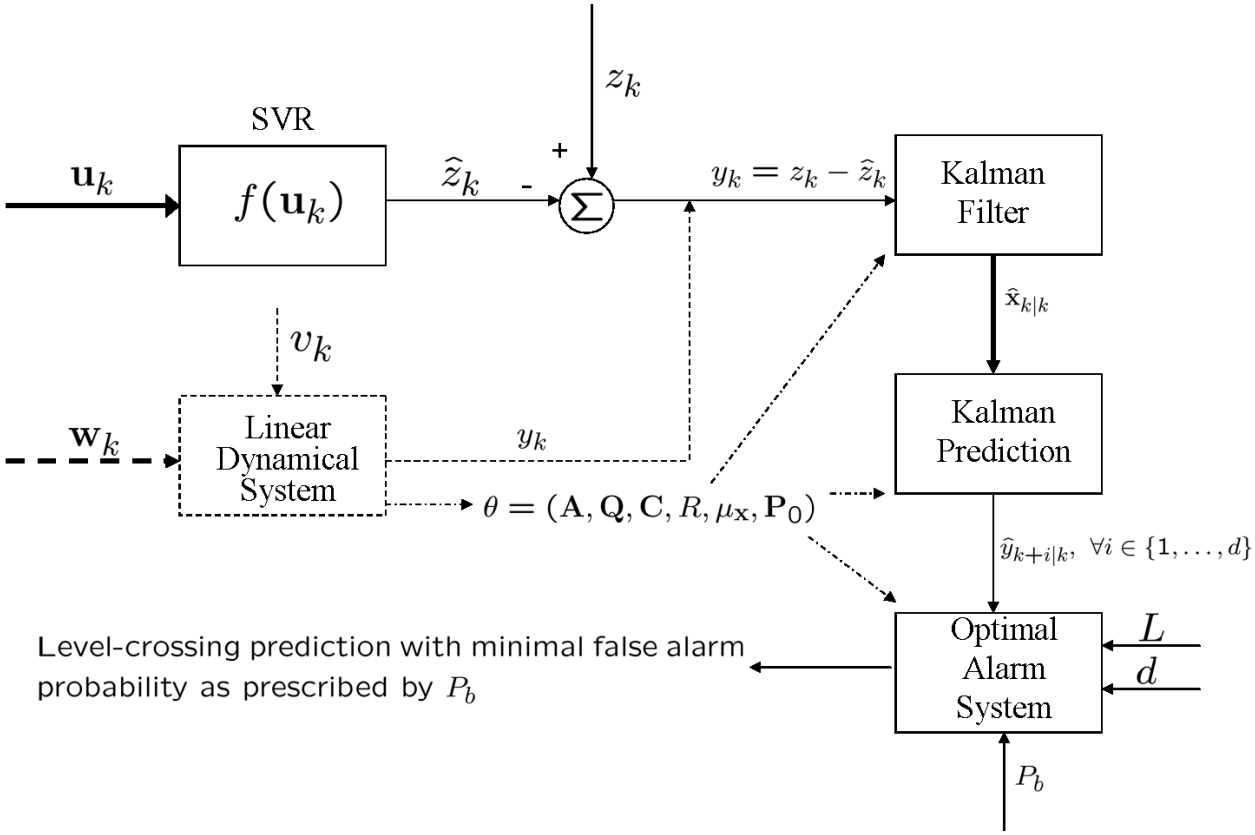


Figure 2. Signal Flow Diagram

ease in discriminability between the two classes of data, which in our case are nominal and anomalous data. Thus, a high AUC can result if the validation (anomalous) data results in a residual that differs greatly than the residual generated from the training (nominal) data, regardless of model fidelity. As such, maximizing the AUC as part of the optimization can be unproductive, and should best be left for evaluation purposes. This discriminability can also be quantified by other means for optimization, using metrics that were designed more for these purposes, such as the small sample corrected AIC (Akaike Information Criterion), (Kolmogorov-Smirnov) KS statistic, and information gain.

The AIC metric is one often used for model order selection in linear dynamical systems, 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 used for the assessment of model fidelity, and has been used in previous related work.[3,7] The KS statistic is fundamentally a measure of Gaussianity of a dataset, by finding the maximum difference between an empirical cumulative distribution function based upon the data and the normal distribution with mean and standard deviation given by the same data. Finally, the information gain, also know as the Kullback-Leibler (KL) divergence can also be used to quantify the discriminability between the nominal and anomalous classes of data. The KL divergence is also use in the AIC computation. The optimization will thus take into account all three metrics, placing the most weight on the KS statistic, as it is valid for all competing methods to be compared, and offers no special advantages to the optimal level-crossing prediction method. The AIC and KL divergence will be used to arbitrate final selection of the optimal kernel width. Overall, the desired objectives are to minimize the KS statistic and AIC, while maximizing the KL divergence.

An additional factor to consider when tuning is selection of the target parameter, shown as the outer loop in Fig. 3. As mentioned previously, there may be multiple target parameter candidates that should be good indicators for the adverse event to be predicted. For the adverse event to be studied in the paper, the most relevant parameters have previously been identified by Das *et al.*,[8] and may act as candidate target parameters. The remaining parameters may act as input or feature parameters. Furthermore, certain features will be better predictors of the candidate targets than others. During the tuning process, candidate

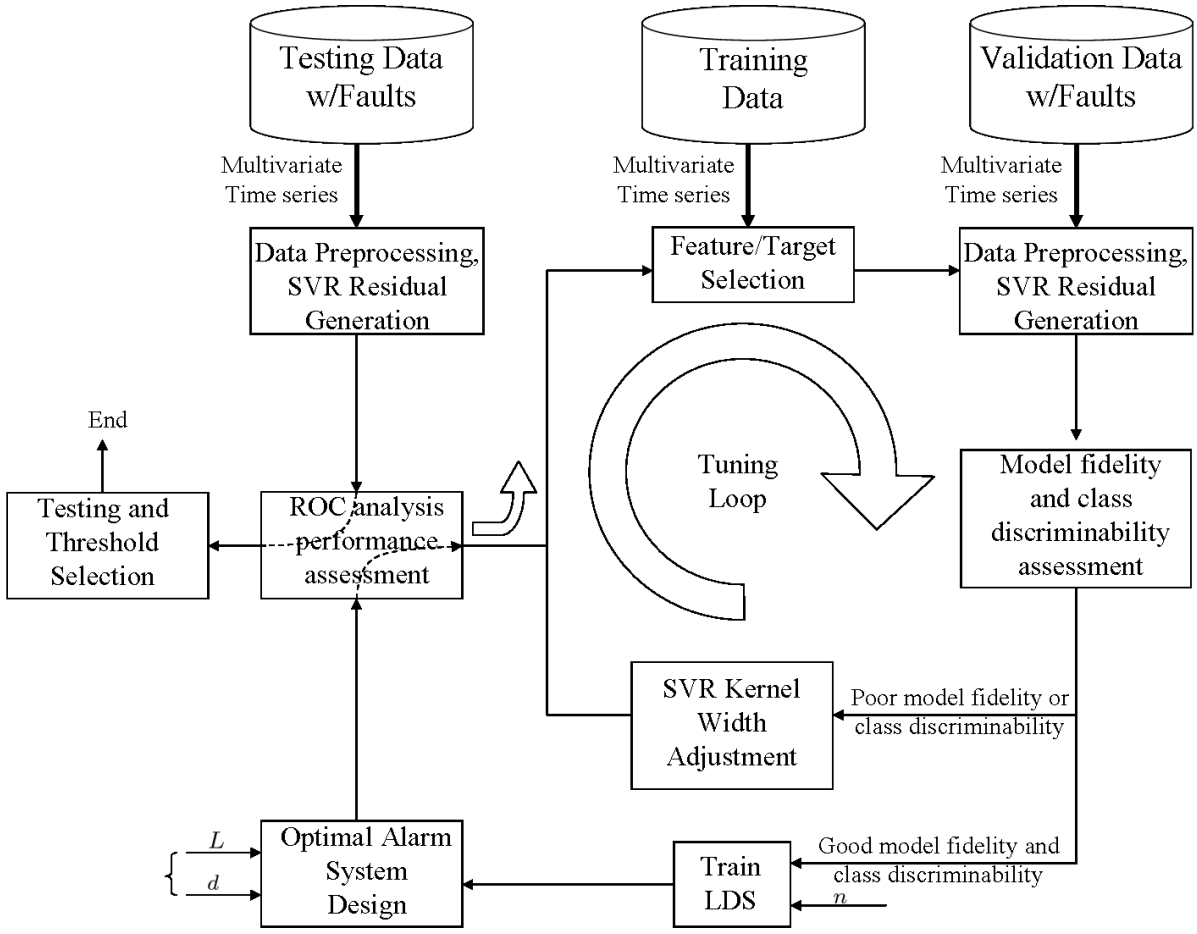


Figure 3. Algorithmic Tuning Diagram

target parameters that are not being tested in that role are then included in the feature parameter set as a predictor of the candidate selected as the target parameter. All candidates will be rotated to the role of target parameter, and the one yielding the best performance as quantified by ROC curve statistics will be selected for final tuning using the inner loop shown in Fig. 3.

One of the many advantages of the optimal alarm system is that ROC curve statistics (the true and false positive rates) can be expressed directly as a function of the model parameters, which are fundamentally an expression of the training data. Therefore, construction of a baseline ROC curve can proceed without the need to observe actual examples of failures, and there is no need to estimate the alarm system metrics empirically. The ROC curve generated from training data serves as an approximation of one that is generated empirically, *i.e.* via simulation. This obviates the need to rely upon having actual available examples of failures for generation of a candidate ROC curve.

As such, it is imperative that the gap between training data-based ROC curve and one based upon actual observations of anomalies be made as small as possible. The level-crossing event must sufficiently characterize an actual adverse event to realize the advantage of expressing the ROC curve as a function of the model parameters. In our case, this translates to ensuring that the correct kernel width from the SVR block is being used to allow for recasting the anomaly detection problem as a level-crossing problem by increasing model fidelity and class discriminability. As seen in Fig. 3, the first step prior to the commencement of any tuning is feature selection. This involves the identification of relevant parameters in the aviation dataset, $\mathbf{u}_k$, that have been considered relevant predictors of the select target parameter, z_k, by experts and the elimination of irrelevant features by other means as documented in Das *et al.*[8]

Subsequently, data preprocessing which includes z-score normalization is performed. A performance assessment is then made by the method previously described, which is based upon the two designated

requirements of good model fidelity and class discriminability. Selection and tuning for an appropriate kernel width value corresponding to these requirements follows. When both requirements have been met, the next step involves training the linear dynamical system, informed by model order selection as a function of the amount of nominal data used for training, and based upon a variety of other indicators to aid in the process. These and other details involved with this step, such as initialization of the model parameters, will be discussed in earnest in Sec. IV. This step is followed by optimal alarm design, which requires selection of L, the critical level, and d, the prediction horizon. L can often be found by selecting an appropriate p-value associated with a reasonable confidence interval for the underlying Gaussian distribution made available by the modeling assumptions. A fixed target value for d can be used to establish the prediction horizon. Both of these parameters have bearing on robustness of prediction performance as described by Martin.[2]

Once the alarm system has been designed, the resulting design parameters can be used for final testing and implementation. This step is seeded by a disjoint hold out dataset which was not used for either training or validation, and presumably contains real examples of adverse events that are similar in nature to those used for validation. It is in this final step that the resulting AUC based upon testing data will be compared to those generated from validation and training. It also implicitly involves the signal flow diagram shown in Fig. 2, where the inputs and outputs are not made explicit in Fig. 3, but can be thought of as taking place within the block labelled "Testing and threshold selection." A threshold will be selected by establishing a maximum allowable false alarm probability, and applying it to the validation ROC curve. The resulting threshold will be used to form a realization using the test data illustrating all observed true and false positives, and missed detections. The same testing regimen will be applied to the two baseline predictive methods described previously for comparison.

III. Support Vector Regression

In this section, we provide a brief description of the $\eta-$Support vector regression algorithm that we use for target prediction.[5] Support vector regression was chosen over other competing regression techniques such as the method(s) used in MSET due to its improved ability to reduce target value prediction error, and built in controls for complexity and numerical stability as found in a nuclear application.[9] Given a finite set of multivariate observations, it is possible to reconstruct an input and target set that takes the form as shown in Eqn. 1, where $\mathbf{U} \triangleq [\mathbf{u}_0 \dots \mathbf{u}_T]^\top$ is an input data matrix of size $(T \times p)$ and the corresponding output is denoted by $\mathbf{z} \triangleq [z_0 \dots z_T]^\top$, termed as the target vector. Thus, there are p parameters and T observations. Once the $\eta-$Support vector regression algorithm is appropriately trained, it is possible to estimate a target function $f(\mathbf{u}_k)$ that has at the most a deviation of η from the actual observed targets $\{z_k\}_{k=0}^T$ for all the input data $\{\mathbf{u}_k \in \mathbb{R}^p\}_{k=0}^T$.

$$\mathbf{z} = f(\mathbf{U}) \tag{1}$$

The target function $f(\cdot)$ is basically a linear combination of weighted similarities between some chosen training points and the test points with an additional offset which is often known as bias, ρ. The chosen training instances with m non-zero weights are termed *support vectors* (SVs, $\mathbf{u}_i$) and they are the representatives of the model. This implies that, given the model, any training points apart from the SVs are of no importance and can be thrown out without changing the performance of the algorithm. The target function is shown in Eqn. 2.

$$f(\mathbf{u}_k) = \sum_{i=1}^{m} (\alpha_i - \hat{\alpha}_i) \langle \mathbf{u}_i, \mathbf{u}_k \rangle + \rho \tag{2}$$

The support vectors and their corresponding weights, α_i and $\hat{\alpha}_i$ result from the solution of a quadratic programming optimization problem in dual form. The expression of the primal problem is shown in Eqn. 3. Further details on the cost function and optimization problem can be found in Smola and Schölkopf.[5]

$$
\begin{aligned}
\text{minimize} \quad & P\left(\mathbf{q}, C, \xi_k^+, \xi_k^-\right) = \frac{1}{2}\mathbf{q}\mathbf{q}^\top + C\sum_{k=0}^{T}\left(\xi_k^+ + \xi_k^-\right) \\
\text{subject to} \quad & \left(z_k - \mathbf{q}^\top\phi(\mathbf{u}_k) - \rho\right) \leq \eta + \xi_k^+ \\
& \left(z_k - \mathbf{q}^\top\phi(\mathbf{u}_k) - \rho\right) \geq \eta + \xi_k^- \\
& \xi_k^+, \xi_k^- \geq 0 \\
& C > 0
\end{aligned} \tag{3}
$$

C and η are user specified regularization and precision parameters respectively. ξ^+, ξ^- are non-zero slack variables, $\mathbf{q}$ is the weight vector normal to the separating hyperplane, ρ is the offset parameter, $\phi(\mathbf{u}_k)$ represents the transformed image of $\mathbf{u}_k \in \mathbb{R}^p$ in the same Euclidean space, and $k \in [0, \ldots, T]$. Throughout this research we have used the RBF (Radial Basis Function) as the mapping function given in Eqn. 4, where σ represents the hyperparameter of the Gaussian function.

$$
\langle \mathbf{u}_i, \mathbf{u}_k \rangle = \exp\left(-\frac{1}{2}\frac{\parallel \mathbf{u}_k - \mathbf{u}_i \parallel^2}{\sigma^2}\right) \tag{4}
$$

The parameter σ controls the overall scale in horizontal variations. Furthermore, for the Gaussian RBF kernel, appropriate values of the hyperparameter, σ, lent itself to distribution of the resulting SVR prediction residual as nearly Gaussian.

IV. Linear Dynamical System

The linear dynamical system will evolve according to Eqns. 5 - 7, demonstrating propagation of both the state, $\mathbf{x}_k \in \mathbb{R}^n$ which is corrupted by process noise $\mathbf{w}_k \in \mathbb{R}^n$, and the covariance matrix, $\mathbf{P}_k$, with time-invariant parameters. The output, $y_k \in \mathbb{R}$ is univariate, and is corrupted by measurement noise $v_k \in \mathbb{R}$.

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{w}_k & (5)\\
y_k &= \mathbf{C}\mathbf{x}_k + v_k & (6)\\
\mathbf{P}_{k+1} &= \mathbf{A}\mathbf{P}_k\mathbf{A}^\top + \mathbf{Q} & (7)
\end{aligned}
$$

where

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

The parameters to be learned are specified in Eqn. 8, as the parameter θ. These parameters are also shown in Fig. 4, which specify them in relation to the probabilistic graphical modeling paradigm which may be used for machine learning purposes. As seen, the number of data points span T time increments.

$$
\theta = (\mu_\mathbf{x}, \mathbf{P}_0, \mathbf{A}, \mathbf{C}, \mathbf{Q}, R) \tag{8}
$$

It is also important to introduce the standard Kalman filter Eqns. 9 - 13, to be used in subsequent learning and prediction formulae.

$$
\begin{aligned}
\hat{y}_{k|k} &= \mathbf{C}\hat{\mathbf{x}}_{k|k} & (9)\\
\hat{\mathbf{x}}_{k|k-1} &= \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1} & (10)\\
\hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{F}_{k|k-1}\varepsilon_k & (11)\\
\mathbf{P}_{k|k-1} &= \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^\top + \mathbf{Q} & (12)\\
\mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{F}_{k|k-1}\mathbf{C}\mathbf{P}_{k|k-1} & (13)
\end{aligned}
$$

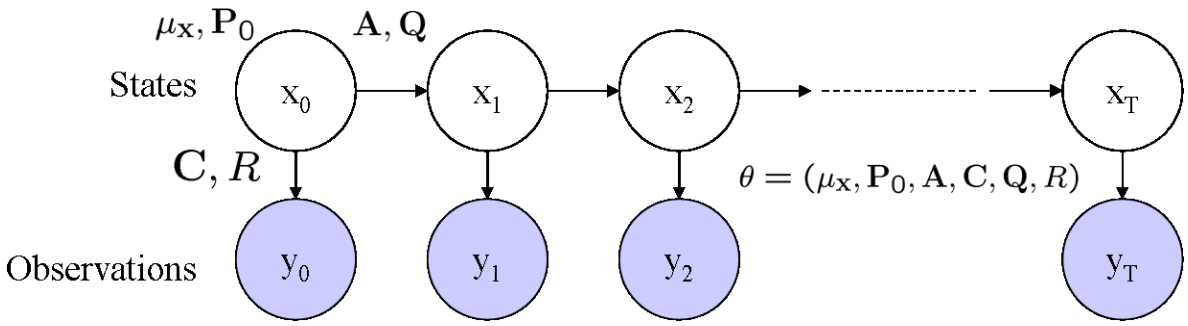


Figure 4. Linear Dynamical System

ε_k represents the *innovation* of the Kalman filter, and the following definitions hold:

$$\begin{aligned}
\hat{\mathbf{x}}_{k|k} &\triangleq E[\mathbf{x}_k|y_0,\ldots,y_k] \\
\mathbf{P}_{k|k} &\triangleq E[(\mathbf{x}_k-\hat{\mathbf{x}}_{k|k})(\mathbf{x}_k-\hat{\mathbf{x}}_{k|k})^\top|y_0,\ldots,y_k] \\
\mathbf{F}_{k|k-1} &\triangleq \mathbf{P}_{k|k-1}\mathbf{C}^\top(\mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}^\top+R)^{-1} \\
\varepsilon_k &\triangleq y_k-\mathbf{C}\hat{\mathbf{x}}_{k|k-1}
\end{aligned}$$

There are two important considerations in learning the parameters that comprise θ. The first consideration is selection of the model order, n, which can be performed using a variety of methods. The second consideration relates to the method of learning the parameters. One such data-driven approach incorporates the use of the EM algorithm, which is an iterative maximum likelihood estimation-based approach that is ultimately a nonlinear optimization problem. As such, it is possible to arrive at a solution which is only a local optimum, and there may be better solutions based upon the location of the initial parameters.

In previous work,[3] the EM algorithm (as implemented by Murphy[10] within the probabilistic graphical modeling context shown in Fig. 4) was used to learn the model parameters using a variety of initialization techniques. Furthermore, the fidelity of resulting models was assessed via AIC-based criteria, such as one presented by Bengtsson and Cavanaugh,[7] in addition to being used for model order selection. In this paper we consider alternative approaches. A report by Derek *et al.*[11] provides an example in which model order selection techniques can be performed with the aid of the SVD (singular value decomposition) of a block Hankel matrix. The blocks of this matrix are constructed by computing sample autocovariance matrices generated from the data. Under certain technical conditions, the rank of the resulting block Hankel matrix will provide an estimate of n, the model order. However, for practical reasons these technical conditions must be relaxed, thus the resulting estimate provides an approximation of the model order. In other work by Overschee and De Moor,[12–14] as well as Favoreel *et al.*,[15] the technical conditions for model order selection vary from those used in the report by Derek *et al.*,[11] and apply almost directly to the system described by Eqns. 5-7. However, they are based upon the SVD of a different matrix, yet are ultimately also constructed by sample data. The number of non-zero singular values will provide an estimate of the model order.

Model order selection can also be assisted with the use of two other supplementary methods. One such method is based upon Canonical Variate Analysis (CVA) introduced by Larimore,[16] in which the principal angle between the row spaces of the past outputs and the future outputs is used to determine model order. Here, the number of principal angles different from 90 degrees provides an estimate of the model order. Intuitively, this makes sense due to the fact that the corresponding row spaces can be considered more linearly dependent as the principle angles approach 90 degrees. As such, including any of the principle angles that approach 90 degrees into the estimate could be considered overfitting the model order by the small margin to linear dependence. The second method is based upon the prediction error as quantified by the *innovation* of the Kalman filter, ε_k, for a variety of candidate model orders. Here, the model order is estimated by selecting the candidate model order above which there is a consistent low value after a jump down from higher values found for lower candidate model orders.

It has thus become clear that model order selection can be somewhat of an art, and based partially upon heuristics, regardless of the method selected. One last heuristic used to determine the final model order involves taking the median or rounded mean of the resulting model orders derived from all three methods presented. Furthermore, the amount of data used to form the resulting model order estimates naturally have some bearing on the outcome of applying these heuristics. Thus, in order to determine the amount of data sufficient to estimate model order based upon select heuristics for all three methods, we have run Monte Carlo simulations as shown in Fig. 5.

The heuristics are applied as described below:

Singular Value Method The model order is estimated as the last instance of consecutive decreases of singular value magnitudes greater than 0.1.

Principle Angle Method The model order is estimated as the last instance of consecutive increases of principal angle magnitudes greater than 1.

Prediction Error Method The model order is estimated as $\arg\min_n \lfloor \epsilon_k^n + \frac{1}{2} \rfloor$, where ϵ_k^n is the simulation error for model order n, where ϵ_k is defined as shown in Eqn. 14, which is also provided in Overschee and De Moor.[12]

$$\epsilon_k = 100 \sqrt{\frac{\sum_{k=1}^{T} \varepsilon_k^2}{\sum_{k=1}^{T} y_k^2}} \tag{14}$$

In Fig. 5 we have computed ensemble averages of model order estimates over 100 runs, shown for increasing data lengths and for increasing target model orders. The target model order was the order used to seed each simulation. The top three panels show the average model order computed using heuristics as a function of target model order and the number of data samples. However, for assessment of how the number of data samples affects the average model order, it is more enlightening to use the bottom three panels, which looks at a "side" view of the top panels, and uses a logarithmic scale for the number of data samples.

There are two observations that can be made from the bottom three panels. The first observation relates to the relative agreement among all three methods for estimating model order when greater than $20,000$ samples are used. Above this limit, there is an apparent convergence to model order estimates between $n = 1$ and $n = 4$ for all three methods, although the convergence is most apparent for the principle angle and prediction error methods, where the range of model order estimates narrows to values between approximately $n = 1$ and $n = 3$. The second conclusion is that even though the target model orders were selected between $n = 1$ and $n = 10$, the average model order estimates are no greater than $n = 4$ for any method, using the given heuristics, and given sufficient data. This indicates that the Monte Carlo simulations generating the data distributed according to the underlying models with fixed orders can be reduced. Model order reduction using balanced realization is a common control theoretic method useful for generating numerically stable reduced order realizations that are both controllable and observable. Intuitively, this phenomenon may be occurring implicitly, as exemplified by this second observation.

When using real data, if we assume that its statistics are fairly well characterized by the underlying distribution of this model, then we can use these observations to provide minimum data requirements ($T_{min} = 20,000$) in guiding model order selection. Furthermore, this minimum data requirement appears to be valid regardless of the *true* model order, as observed by the apparent convergence to a limited range of reduced model orders. Using these model order selection techniques provided for by methods described by Overschee and De Moor,[12–14] parameter estimates for θ_a shown in Eqn. 15 can also be obtained.

$$\theta_a = (\mu_{\mathbf{x}}, \mathbf{P}_0, \mathbf{A}, \mathbf{C}, \mathbf{Q}, R, \mathbf{S}) \tag{15}$$

$\mathbf{S} \stackrel{\triangle}{=} \mathrm{cov}(\mathbf{w}_k, v_k)$ represents the correlation between input and measurement noise, which is not modeled in Eqns. 5-7, such that $\mathbf{S} = \mathbf{0}$. These parameter estimates are based upon a subspace projection method know as N4SID (Numerical Algorithms for Subspace State-Space System IDentification) and are inherently suboptimal due to the error introduced with the subspace projection. However, often the resulting models offer a good fit to the data, and are also often used as initial parameters for susbsequent refinement during iterative nonlinear optimzation using a prediction error minimization (PEM) method. This is equivalent to choosing a cost function similar to Eqn. 14, as documented in the literature by Ljung.[17] Ostensibly, EM

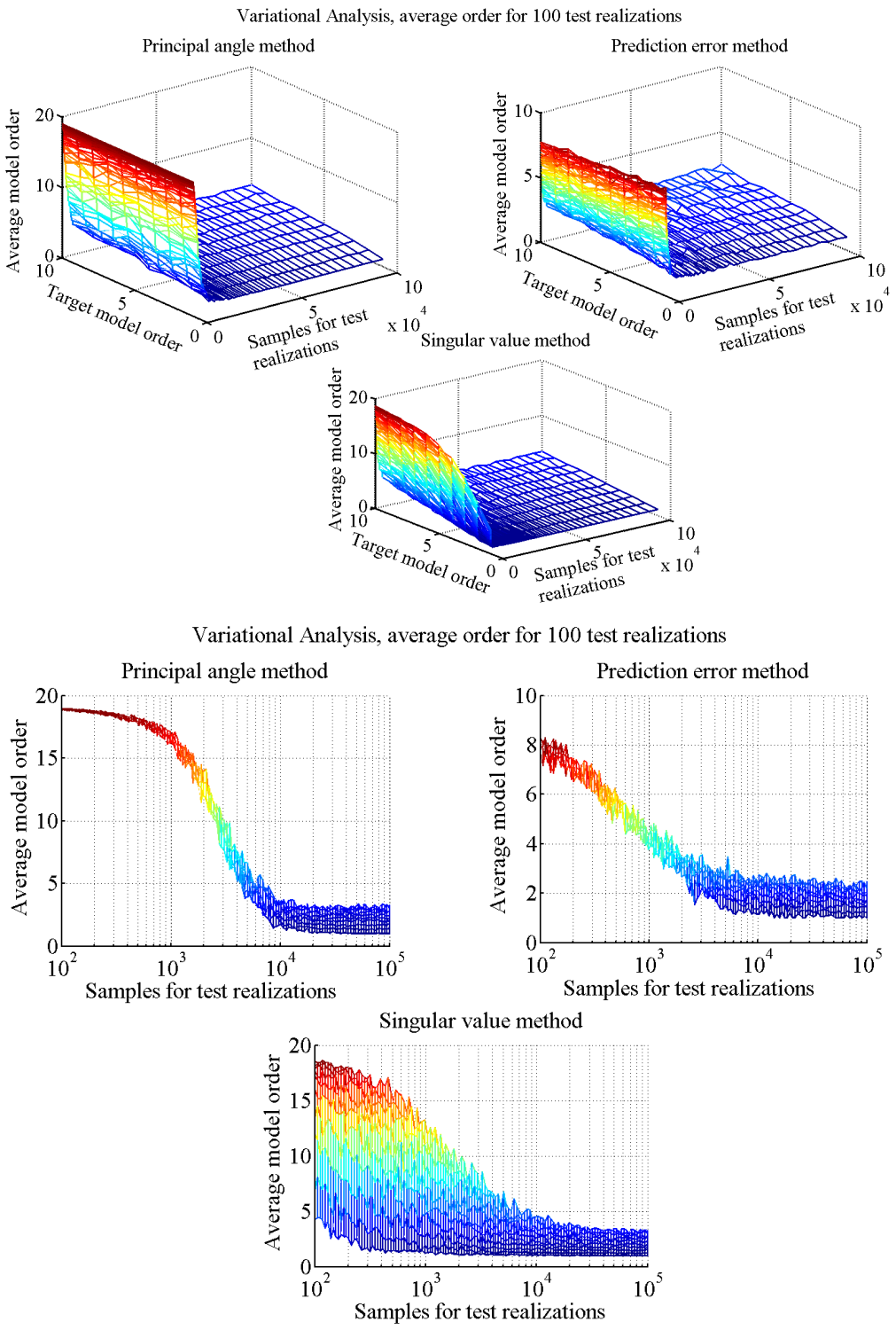


Figure 5. Minimum Data Requirements for Model Order Selection

learning here is equivalent to applying PEM, and although the cost function[10] is distinctly different, the fact that the parameters are learned via iterative nonlinear optimization is the same. Therefore, the N4SID parameters can be used as initial estimates for EM learning, where the model by definition has a non-zero correlation between input and measurement noise. Implicitly, the equivalent state-space model as described

by Eqns. 5-7 is found by holding $\mathbf{S} = \mathbf{0}$ during EM learning.

Another issue is system stability, which is not guaranteed by any of the methods described in this section. Thus, only systems that have been identified with spectral radius $\rho(\mathbf{A}) < 1$ are accepted. Techniques to enforce stability recommended by Overschee and DeMoor[12] (*i.e.* augmenting the extended observability matrix) were applied for systems identified as unstable. However, in these cases the resulting spectral radius was often so close to unity that the conditions for uniqueness of the steady-state solution for Eqn. 7 were violated. In general, low order stable systems can be identified without having to rely on the use of this technique. As such, most of the unstable systems eliminated from consideration were higher order systems. Future research will allow for enforcement of stability throughout the entire identification and learning procedure, as suggested in recent work.[18,19]

V. Optimal Level-Crossing Prediction

Recall that the anomaly detection problem can be cast as an optimal level-crossing prediction problem in the context of Figs. 1-3. In this section, we provide an overview of some of the fundamental theoretical and implementation details of this method. A level-crossing event, C_k, is defined with a critical level, L, that is assumed to have a fixed, static value. The level is exceeded by a critical parameter, y_k, that can be represented by a dynamic process, and is modeled as a zero-mean stationary linear dynamic system driven by Gaussian noise. The theoretical underpinnings of this approach are based upon this standard representation of the optimal level-crossing problem. As such, our underlying assumption is that we can fit all SVR residuals generated in Fig. 2 to a model represented by a stationary linear dynamical system driven by Gaussian noise.

The essence of the optimal alarm system is derived from the use of the likelihood ratio resulting in the conditional inequality: $P(C_k|y_0,\ldots,y_k) \geq P_b$. This basically says "give alarm when the conditional probability of the event, C_k, exceeds the level P_b." Here, P_b represents some optimally chosen border or threshold probability with respect to a relevant alarm system metric. It is necessary to find the alarm regions in order to design the alarm system. The event, C_k, can be chosen arbitrarily, and is usually defined with respect to a prediction window, d, as well as the critical threshold, L. In this paper, the event of interest is shown in Eqn. 16, and represents at least one exceedance outside of the threshold envelope specified by $[-L, L]$ of the process y_k within the specified look-ahead prediction window, d.

$$C_k \triangleq \bigcup_{j=1}^{d} S_{k+j} \tag{16}$$

$$= \bigcup_{j=1}^{d} E'_{k+j} \tag{17}$$

$$= \mathcal{I} \setminus \bigcap_{j=1}^{d} E_{k+j} \tag{18}$$

where

$$E_{k+j} \triangleq \{|y_{k+j}| < L\},\ \forall j \geq 1$$

$$S_{k+j} \triangleq \begin{cases} E'_{k+j} & j = 1 \\ \bigcap_{i=1}^{j-1} E_{k+i}, E'_{k+j} & \forall j > 1 \end{cases}$$

Eqn. 19 represents the unconditional probability of the level-crossing event in its most compact representational form.

$$P(C_k) = 1 - \int_{-L}^{L} \cdots \int_{-L}^{L} \mathcal{N}(\mathbf{y}_d; \mu_{\mathbf{y}_d}, \Sigma_{\mathbf{y}_d}) d\mathbf{y}_d \tag{19}$$

where

$$
\begin{aligned}
\mathbf{y}_d &\triangleq \begin{bmatrix} y_{k+1} \\ \vdots \\ y_{k+d} \end{bmatrix}, \quad \mu_{\mathbf{y}_d} = \mathbf{0}_d \\
\Sigma_{\mathbf{y}_d} &\triangleq \begin{cases} \mathbf{C}\mathbf{P}_k\mathbf{C}^\top + R & \forall i = j \in \{1, \ldots, d\} \\ \mathbf{C}\mathbf{P}_{k+i,k+j}\mathbf{C}^\top & \forall j > i \in \{1, \ldots, d\} \end{cases} \\
\text{and } \mathbf{P}_{k+i,k+j} &\triangleq \mathbf{A}^j(\mathbf{P}_k - \mathbf{P}_{ss}^L)(\mathbf{A}^\top)^i + \mathbf{A}^{j-i}\mathbf{P}_{ss}^L
\end{aligned}
$$

We may approximate $\Sigma_{\mathbf{y}_d}$ by substituting the steady-state version of the Lyapunov equation $\mathbf{P}_{ss}^L$ in place of $\mathbf{P}_k$, given previously as Eqn. 7, which agrees with our assumption of stationarity. As such,

$$
\Sigma_{\mathbf{y}_d} \approx \begin{cases} \mathbf{C}\mathbf{P}_{ss}^L\mathbf{C}^\top + R & \forall i = j \in \{1, \ldots, d\} \\ \mathbf{C}\mathbf{A}^{j-i}\mathbf{P}_{ss}^L\mathbf{C}^\top & \forall j > i \in \{1, \ldots, d\} \end{cases}
$$

This approximation, while it introduces error with regards to the probability of a level-crossing event, $P(C_k)$ at a specific point in time, k, is ostensibly negligible and will provide for a great computational advantage in the design of all alarm systems that it is based upon. Instead of designing an alarm system for each time step, we design a single alarm system based upon the limiting statistics that are reached at steady-state, greatly reducing the computational burden. It is these specific types of restrictions that have been lifted in work by Antunes *et al.*,[20] but also incur much greater computational effort.

Previous work[2] provides the mathematical underpinnings for the optimal alarm condition corresponding to the level-crossing event, shown here as Eqn. 20. Alternatively, the optimal alarm condition derived in the cited paper can be expressed in terms of the subevents E_{k+j}, as shown in Eqn. 21.

$$
\begin{aligned}
P(C_k|y_0, \ldots, y_k) &\geq P_b &(20)\\
\Leftrightarrow P(\bigcap_{j=1}^{d} E_{k+j}|y_0, \ldots, y_k) &\leq 1 - P_b &(21)
\end{aligned}
$$

Recall from Fig. 2 that state estimates generated from Kalman filtering Eqns. 9 - 13 were to be used in subsequent prediction formulae. As such, here they can be used to form forward projected residual value predictions, $\hat{y}_{k+j|k}$, spanning the prediction horizon, $\forall j \in \{1, \ldots, d\}$ for the level-crossing event. Therefore, relevant predictions, covariances and cross-covariances must be computed and are given below as Eqns. 22-26, respectively.

$$
\begin{aligned}
\hat{y}_{k+j|k} &= \mathbf{C}\mathbf{A}^j\hat{\mathbf{x}}_{k+j|k} &(22)\\
\mathbf{P}_{k+j|k} &= \mathbf{A}^j(\mathbf{P}_{k|k} - \mathbf{P}_{ss}^L)(\mathbf{A}^\top)^j + \mathbf{P}_{ss}^L &(23)\\
&\approx \mathbf{A}^j(\hat{\mathbf{P}}_{ss}^R - \mathbf{P}_{ss}^L)(\mathbf{A}^\top)^j + \mathbf{P}_{ss}^L &(24)\\
\mathbf{P}_{k+i,k+j|k} &= \mathbf{A}^j(\mathbf{P}_{k|k} - \mathbf{P}_{ss}^L)(\mathbf{A}^\top)^i + \mathbf{A}^{j-i}\mathbf{P}_{ss}^L &(25)\\
&\approx \mathbf{A}^j(\hat{\mathbf{P}}_{ss}^R - \mathbf{P}_{ss}^L)(\mathbf{A}^\top)^i + \mathbf{A}^{j-i}\mathbf{P}_{ss}^L &(26)\\
\hat{\mathbf{P}}_{ss}^R &= \mathbf{P}_{ss}^R - \mathbf{F}_{ss}\mathbf{C}\mathbf{P}_{ss}^R &(27)\\
\mathbf{F}_{ss} &= \mathbf{P}_{ss}^R\mathbf{C}^\top(\mathbf{C}\mathbf{P}_{ss}^R\mathbf{C}^\top + R)^{-1} &(28)
\end{aligned}
$$

$\mathbf{P}_{ss}^R$ is the combined steady-state version of Eqns. 12 and 13 given previously, or the discrete algebraic Riccati equation, and $\hat{\mathbf{P}}_{ss}^R$ is the steady-state *a posteriori* covariance matrix given in Eqn. 27. Eqn. 28, which is the steady-state version of the Kalman gain from Eqn. 14 is also used in Eqn. 27.

The approximations shown in Eqns. 24 and 26, while suboptimal with regards to the optimality of the Kalman filter as cited by Lewis,[21] will provide for a great computational advantage in design of the optimal alarm system and its corresponding approximations. Instead of designing an optimal alarm system for each time step, we design a single optimal alarm system based upon the limiting statistics that are reached at

steady-state, greatly reducing the computational burden. The assumption of stationarity thus required for the design of an optimal alarm system holds here as well.

A more formal representation of the optimal alarm region, A_k, is shown in Eqn. 29, which essentially defines a sublevel set of $g(\hat{\mathbf{y}}_d) \stackrel{\triangle}{=} P(\bigcap_{j=1}^{d} E_{k+j}|y_0,\ldots,y_k)$ as a function of $\hat{\mathbf{y}}_d$.

$$\begin{aligned} A_k &\stackrel{\triangle}{=} \{\bigcap_{i=1}^{d} \hat{y}_{k+i|k} : P(C_k|y_0,\ldots,y_k) \geq P_b\} \\ &\stackrel{\triangle}{=} \{\bigcap_{i=1}^{d} \hat{y}_{k+i|k} : P(\bigcap_{j=1}^{d} E_{k+j}|y_0,\ldots,y_k) \leq 1-P_b\} \end{aligned} \tag{29}$$

As was discussed in seminal work in this topic,[2] it is not possible to obtain a closed-form representation of the parametrization for the optimal alarm region shown in Eqn. 29. To allay the computational burden of using Monte Carlo simulation to generate ROC curve statistics to be estimated empirically, we can generate ROC curve statistics by numerically integrating expressions for the computation of relevant multivariate normal probabilities whose domains of integration serve as approximations for Eqn. 29. In the seminal article,[2] there are two such approximations that were studied. In this paper we use the "closed-form" approximation that requires the least computational effort which was shown not to lose any appreciable accuracy as compared to the other approximation, as quantified by the AUC.

The approximation to the optimal alarm region in "closed-form," is shown in Eqn. 30.

$$\bigcup_{j=1}^{d}\bigcup_{i\in\mathcal{B}} A_k^{i,j} = \bigcup_{j=1}^{d} |\hat{y}_{k+j|k}| \geq L + \sqrt{V_{k+j|k}}\Phi^{-1}(P_b) \equiv L_{A_j} \tag{30}$$

where

$$\begin{aligned} A_k^{i,j} &= \{\hat{y}_{k+j|k} : P(E_{k+j}^{i\,'}|y_0,\ldots,y_k) \geq P_b\} \\ V_{k+j|k} &\stackrel{\triangle}{=} \mathbf{C}\mathbf{P}_{k+j|k}\mathbf{C}^\top + R \\ i &\in \mathcal{B} \equiv \{\ell,\upsilon\} = \{\text{lower limit, upper limit}\} \\ E_{k+j}^{\upsilon} &= \{y_{k+j} < L\} \\ E_{k+j}^{\ell} &= \{y_{k+j} > -L\} \end{aligned}$$

$\Phi^{-1}(\cdot)$ represents the inverse cumulative normal standard distribution function, and $L_{A_j} \forall j \in \{1,\ldots,d\}$ represent the limits of integration. It is possible to generate formulae for the true and false positive rates as a function of L_{A_j} by using Eqns. 31-32, where in place of A_k its approximation $\bigcup_{j=1}^{d}\bigcup_{i\in\mathcal{B}} A_k^{i,j}$ may be used.

True positive rate:

$$P_d = P(C_k|A_k) = \frac{P(C_k,A_k)}{P(A_k)} \tag{31}$$

False positive rate:

$$\begin{aligned} P_{fa} = P(A_k|C_k^{'}) &= \frac{P(C_k^{'},A_k)}{P(C_k^{'})} \\ &= \frac{P(A_k)-P(C_k,A_k)}{1-P(C_k)} \end{aligned} \tag{32}$$

Because we have already introduced the formula for $P(C_k)$ in Eqn. 19, which holds regardless of the alarm system being used, we must only find expressions for $P(C_k,A_k)$ and $P(A_k)$.

$$\begin{aligned} P(A_k) &= \begin{cases} P(\bigcup_{j=1}^{d}\bigcup_{i\in\mathcal{B}} A_k^{i,j}) & P_b > P_{b_{crit}} \\ 1 & P_b = P_{b_{crit}} \end{cases} \\ &= \begin{cases} 1 - P(\bigcap_{j=1}^{d}\bigcap_{i\in\mathcal{B}} A_k^{i,j}) & P_b > P_{b_{crit}} \\ 1 & P_b = P_{b_{crit}} \end{cases} \end{aligned} \tag{33}$$

$$P(C_k, A_k) = \begin{cases} P(C_k) - P(A'_k) + P(C'_k, A'_k) & P_b > P_{b_{crit}} \\ P(C_k) & P_b = P_{b_{crit}} \end{cases} \tag{34}$$

where

$$P(A'_k) = P(\bigcap_{j=1}^{d} \bigcap_{i \in \mathcal{B}} A_k^{i,j}) = P(\bigcap_{j=1}^{d} |\hat{y}_{k+j|k}| < L_{A_j})$$
$$= \int_{-L_{A_1}}^{L_{A_1}} \cdots \int_{-L_{A_d}}^{L_{A_d}} \mathcal{N}(\hat{\mathbf{y}}_d; \mu_{\mathbf{y}_d}, \hat{\Sigma}_{\hat{\mathbf{y}}_d}) d\mathbf{y}_d$$

and

$$\begin{aligned} \hat{\Sigma}_{\hat{\mathbf{y}}_d} &\triangleq \Sigma_{\mathbf{y}_d} - \hat{\Sigma}_{\mathbf{y}_d} \\ &= \mathbf{O}(\mathbf{P}_{ss}^{L} - \hat{\mathbf{P}}_{ss}^{R})\mathbf{O}^{\top} \\ \mathbf{O} &\triangleq \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^d \end{bmatrix} \\ \hat{\mathbf{y}}_d &\triangleq E[\mathbf{y}_d | y_0, \ldots, y_k] = \begin{bmatrix} \hat{y}_{k+1|k} \\ \vdots \\ \hat{y}_{k+d|k} \end{bmatrix} \\ \hat{\Sigma}_{\mathbf{y}_d} &\triangleq \begin{cases} V_{k+i|k} & \forall i = j \in [1, \ldots, d] \\ \mathbf{CP}_{k+i,k+j|k}\mathbf{C}^{\top} & \forall i \neq j \in [1, \ldots, d] \end{cases} \end{aligned}$$

Furthermore,

$$P(C'_k, A'_k) = P(\bigcap_{j=1}^{d} E_{k+j}, \bigcap_{j=1}^{d} \Omega'_{A_j})$$
$$= \int_{-L}^{L} \cdots \int_{-L}^{L} \int_{-L_{A_1}}^{L_{A_1}} \cdots \int_{-L_{A_d}}^{L_{A_d}} \mathcal{N}(\mathbf{z}; \mu_{\mathbf{z}}, \Sigma_{\mathbf{z}}) d\mathbf{z}$$

where

$$\begin{aligned} \mathbf{z} &\triangleq \begin{bmatrix} \mathbf{y}_d \\ \hat{\mathbf{y}}_d \end{bmatrix} \\ \mu_{\mathbf{z}} &\triangleq \begin{bmatrix} \mu_{\mathbf{y}_d} \\ \mu_{\mathbf{y}_d} \end{bmatrix} \\ \Sigma_{\mathbf{z}} &\triangleq \begin{bmatrix} \Sigma_{\mathbf{y}_d} & \hat{\Sigma}_{\hat{\mathbf{y}}_d} \\ \hat{\Sigma}_{\hat{\mathbf{y}}_d} & \hat{\Sigma}_{\hat{\mathbf{y}}_d} \end{bmatrix} \end{aligned}$$

The domain of feasibility for this approximation is shown in Eqn. 35, whose derivation is provided in the seminal article.[2]

$$P_b \geq P_{b_{crit}} = \Phi\left(\frac{-L}{\sqrt{V_{k+d|k}}}\right) \tag{35}$$

As far as runtime implementation details are concerned, we have the option of using the "closed-form" approximation which amounts to limit checking of the type governed by Eqn. 30. As mentioned previously, the advantage of doing so is that the computational burden is much less than that of the other approximation or the exact alarm condition shown in Eqn. 20, especially as the prediction horizon increases. Furthermore,

the implemented alarm system will exhibit the same characteristics as the designed alarm system. However, the approximation is inherently suboptimal. As such, we may alternatively use the exact alarm condition for runtime implementation. However, doing so will only guarantee optimality if the alarm system were designed by using Monte Carlo simulation to generate ROC curve statistics, in the limit of infinite samples. It must be emphasized that we've chosen to use the "closed-form" approximation for design of the alarm system rather than simulating an unknown and infeasibly large number of samples to approximate the limiting statistics. Any error that is introduced during this design process as characterized in the seminal article[2] will translate at runtime implementation. Hence, the approximation error cannot be avoided as a result of this choice. As such, the best choice will be to perform design of the alarm system with knowledge of how the suboptimality manifests itself at design time. Ultimately, the lesser of two evils is to pay now at design time rather than later at runtime.

When assessing the performance of the alarm system via the ROC curve, either for validation during tuning, or for final testing, it will be necessary to obtain both ground truth data and a score representing the degree of anomalousness of each monitored data point. In our case, the score is quantified by the conditional probability of the level-crossing event, which represents the basis of the optimal alarm condition without any approximations. Unlike run-time implementation, the "closed form" approximation cannot be used to substitute for the exact optimal alarm condition here in the context of the conditional level-crossing probability. Approximation of the alarm region occurs in vector space and does not translate to a single equivalent probability in probability space. As such, we must rely on the actual optimal alarm conditional level-crossing probability for validation and final testing when constructing the ROC curve. This implies that an error will be present throughout the tuning process due to the fact that the closed form approximation will be used at design time and the exact optimal alarm condition will be used for tuning. However, under certain technical conditions (see Martin[2]) this error can be kept to a minimum, to the extent that it is on par with standard sampling error.

VI. Results

Prior to formal validation, testing, and evaluation of the optimal level-crossing predictor performance on real aviation data and real anomalies, we will conduct a preliminary set of experiments. One experiment will use both simulated data and simulated faults, and the other will use real data and simulated faults. Conducting both of these experiments will ensure that we learn how much the model fidelity affects the ability of the optimal level-crossing predictor to outperform a standard method based upon exceedances. This is used for comparison because it is one that is commonly used for prediction. The experiments will also involve a comparison to a baseline prediction method that is based upon the Sequential Probability Ratio Test (SPRT), a central part of MSET that has met with quite favorable results and is used ubiquitously in nuclear applications, as well as aviation and space applications.[22]

The SPRT test statistic is a cumulative log-likelihood ratio between the probability distributions characterizing anomalous and nominal behavior, respectively, as shown in Eqn. 36. The denominator represents the null hypothesis, $\mathcal{H}_0$ shown as Eqn. 38 and characterizes nominal behavior, while the numerator represents the alternative hypothesis, $\mathcal{H}_j$ and characterizes anomalous behavior. There are four distinct alternative hypotheses shown as Eqns. 39-42, which are tested using the SPRT statistics that are computed with MSET to provide comprehensive coverage. These statistics test the hypotheses for both positive and negative mean drifts, as well as nominal and inverse variance shifts. The optimal stopping rule is provided by Eqn. 37, where P_d and P_{fa} are preselected target values.

$$S_k = S_{k-1} + \sum_{i=1}^{k} \log \frac{p(\varepsilon_i|\mathcal{H}_j)}{p(\varepsilon_i|\mathcal{H}_0)}, \quad j \in \{1, \ldots, 4\} \tag{36}$$

$$S_k > \log \frac{P_d}{P_{fa}} \tag{37}$$

$$p(\varepsilon_i|\mathcal{H}_0) = \mathcal{N}(\varepsilon_i; 0, \mathbf{C}\mathbf{P}_{ss}^R\mathbf{C}^\top + R) \tag{38}$$

$$p(\varepsilon_i|\mathcal{H}_1) = \mathcal{N}(\varepsilon_i; M, \mathbf{C}\mathbf{P}_{ss}^R\mathbf{C}^\top + R) \tag{39}$$

$$p(\varepsilon_i|\mathcal{H}_2) = \mathcal{N}(\varepsilon_i; -M, \mathbf{C}\mathbf{P}_{ss}^R\mathbf{C}^\top + R) \tag{40}$$

$$p(\varepsilon_i|\mathcal{H}_3) = \mathcal{N}(\varepsilon_i; 0, V(\mathbf{C}\mathbf{P}_{ss}^R\mathbf{C}^\top + R)) \tag{41}$$

$$p(\varepsilon_i|\mathcal{H}_4) = \mathcal{N}(\varepsilon_i; 0, \frac{\mathbf{C}\mathbf{P}_{ss}^R\mathbf{C}^\top + R}{V}) \tag{42}$$

Table 1. Simulation Experiments

Alternate Hypothesis	**Simulation Parameters**	**Testing Criteria**
Standard Exceedance	$p = 0.01$	$p = 0.01$
Optimal Level-Crossing Prediction	$d = 5$, $p = 0.01$	$d = 5$, $p = 0.01$
SPRT Positive Mean Shift	Ramp to nominal $\max_k y_k$ at random time	$M = 0.01$
SPRT Negative Mean Shift	Ramp to nominal $-\max_k y_k$ at random time	$M = 0.01$
SPRT Nominal Variance Shift	Apply constant multiplier $M = 5$ to nominal y_k at random time	$V = 1.5$
SPRT Inverse Variance Shift	Apply constant multiplier $M = \frac{1}{5}$ to nominal y_k at random time	$V = 1.5$

These hypotheses along with the level-crossing alternate hypothesis will all be tested via simulation. In order to allow for a fair comparison of the alternate level-crossing hypothesis to the remaining four SPRT-based hypotheses, we will implement a prewhitening filter for the latter. This is required in order to meet the assumptions implicit in using the SPRT, which is that the residuals coming from the estimation part of MSET (which in our case are the SVR residuals) are white (*i.e.* not serially correlated), and Gaussian. Gaussianity has already been provided for by application of the tuning procedure discussed in Sec. II, illustrated by Fig. 3. Furthermore, the state estimation procedure involved with MSET is specifically designed to yield residuals that are white.[4] Since we are not using the same state estimation procedure as MSET, this prewhitening step is compulsory. It is important to note that LDS parameters $\mathbf{C}$, $\mathbf{P}_{ss}^R$, and R which were trained using the

methods described in Sec. IV are the same ones used in the tests corresponding to Eqns. 38-42. Conveniently, prewhitening occurs naturally via the Kalman filter residual, and thus the associated filter parameters are based upon the same θ subsequently used for optimal level-crossing prediction. This lends itself nicely to a well balanced basis for comparison. The failure simulations will be seeded by inserting candidate faults into nominal data to generate ground truth data corresponding to all five of the alternate hypotheses, and tested accordingly as shown in Table 1. The nominal real data is derived from the concatenation of 11 real flights from regional jets previously identified as nominal, all of which come from the same commercial aviation dataset.

The positive and negative shifts were respectively simulated with linearly increasing and decreasing ramps as the DC frequency component. This simulation type was selected in order to more closely approximate the gradual onset of an adverse event rather than an immediate shift which would be less amenable to the assessment of predictive capability. The metric used to assess performance will be the AUC, which quantifies the probability of correctly ranking two randomly selected nominal and anomalous data points respectively.[23] In order to evaluate the AUC metric with respect to the desired 2 second prediction time horizon, the binary ground truth vector used to compute it is advanced by the number of indices corresponding to this duration. Eqns. 43 and 44 provide a summary of the performance of all of the methods as confusion matrices $\mathbf{M}_{sr}$ and $\mathbf{M}_{ss}$, respectively. The subscript sr is meant to indicate that real data is seeded with simulated faults, and the subscript ss is mean to indicate that simulated data is seeded with simulated faults. The rows correspond to the simulations of the alternate hypotheses shown in Table 1 (in the same order), and columns correspond to tests of those alternate hypotheses (again, in the same order as shown in Table 1). Thus, the performance of a specific alternate hypothesis being tested for all 6 simulated anomalies can be determined by looking at all of the row entries for the corresponding column.

$$\mathbf{M}_{sr} = \begin{bmatrix} 0.9712 & 0.9709 & 0.3471 & 0.3835 & 0.3859 & 0.5896 \\ 0.9013 & 0.9047 & 0.3538 & 0.3776 & 0.4015 & 0.5604 \\ 0.8457 & 0.8551 & 0.5319 & 0.0030 & 0.2452 & 0.5769 \\ 0.9215 & 0.9271 & 0.0000 & 0.99997 & 0.9290 & 0.0065 \\ 0.8773 & 0.8856 & 0.8884 & 0.9615 & 0.9865 & 0.8902 \\ 0.3199 & 0.3537 & 0.0002 & 0.0001 & 0.0000 & 0.999992 \end{bmatrix} \tag{43}$$

As we can see for the case of real data with simulated faults, the optimal level-crossing predictor and standard exceedance measure both perform well for all of the simulated anomalies with the notable exception of the inverse variance shift. The SPRT tests for the negative mean, nominal and inverse variance shifts yield almost perfect performance for robust detection of their respective fault simulations. The SPRT test for the positive mean shift yields a less desirable result for robust detection of the respective fault simulation. With certain exceptions, the SPRT tests perform poorly on robustly detecting fault simulations other than the ones they were hypothesized to detect. All tests, however seem to be able to robustly detect the nominal variance shift fault simulation. The SPRT tests have difficulty robustly detecting the level-crossing and exceedance events, although the inverse variance SPRT test performs better than random guessing for both.

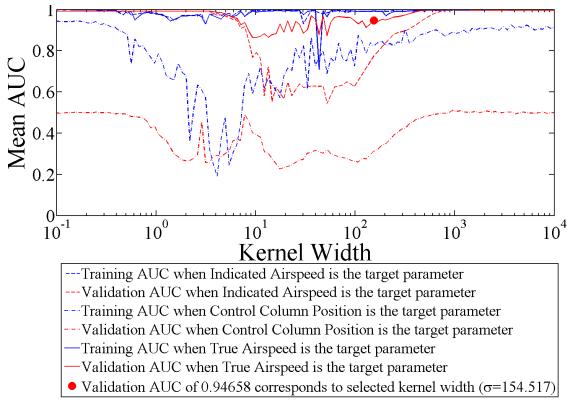
Even though three out of the four SPRT tests yield nearly perfect performance for robust detection of their respective fault simulations, the thresholds corresponding to the optimal stopping rule for these tests never result in the lowest possible false positive and missed detection rates. We also note that the results reported in Eqns. 43 and 44 depend heavily on the choice of SPRT testing criteria (*i.e.* M and V), which provide a measure of sensitivity for the tests. Thus, in general both the optimal level-crossing predictor and standard exceedance measure provide broader coverage than the SPRT-based tests for hypothesized anomalies outside of the class of anomalies they were designed to detect.

$$\mathbf{M}_{ss} = \begin{bmatrix} 0.9918 & 0.9933 & 0.5341 & 0.5205 & 0.5382 & 0.5263 \\ 0.9381 & 0.9537 & 0.5366 & 0.4936 & 0.5323 & 0.4975 \\ 0.7288 & 0.7388 & 0.4154 & 0.0223 & 0.0236 & 0.0007 \\ 0.8976 & 0.9054 & 0.0000 & 0.999996 & 0.4906 & 0.0000 \\ 0.8726 & 0.8732 & 0.7777 & 0.0026 & 0.99995 & 0 \\ 0.1840 & 0.1985 & 0.0015 & 0.0004 & 0.0019 & 0.9876 \end{bmatrix} \tag{44}$$

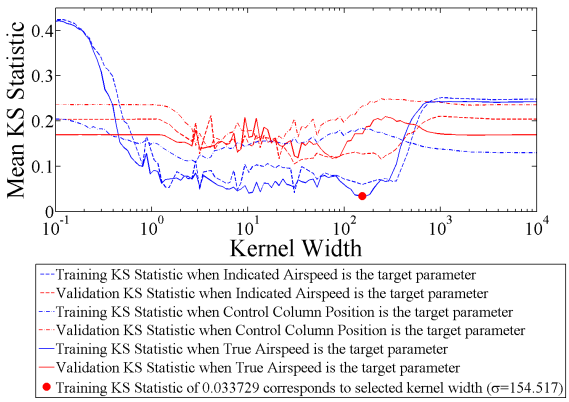
We note that the primary difference between $\mathbf{M}_{ss}$ and $\mathbf{M}_{sr}$ lies in the fact that the optimal level-crossing

predictor outperforms the standard exceedance measure when exposed to simulated data produced by a model that it is designed to be optimal for. This can be ascertained by finding that there is a wider margin between the AUC values in the leading 2×2 submatrix of $\mathbf{M}_{ss}$ than that of $\mathbf{M}_{sr}$. Thus, the more closely the model describes the real data generating source (*i.e.* higher model fidelity), the wider the margin of optimality between the optimal level-crossing predictor and standard exceedance methods.

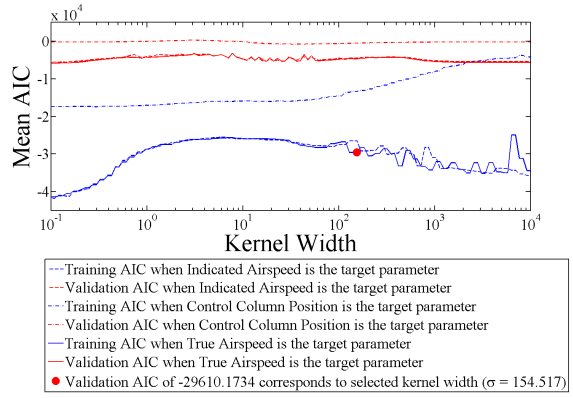
The results presented thus far meet with intuition, however, real performance should be dictated by the evaluation of *true* anomalous events as opposed to the ones that are simulated to mimic the hypotheses that have been selected to represent them. As such, we apply a formal validation and testing procedure to generate results for the anomalies previously identified by experts. In testing both real data and real anomalies using the method shown in Fig. 3, we will be able to discern not only how model fidelity affects the ability of the optimal level-crossing predictor to outperform the standard exceedance method, but also how well both compare to the SPRT methods when exposed to real faults. In this study, model fidelity can be positively influenced through a variety of methods, namely, using a sufficient amount of data as identified in Sec. IV or adjusting the kernel width to provide a Gaussian character to the residual. Increasing the amount of training data to provide for an increase in model fidelity will prove to be computationally prohibitive. Therefore, we will focus on the method previously proposed from Fig. 3, by using a tuning loop and adjusting the kernel width for SVR.



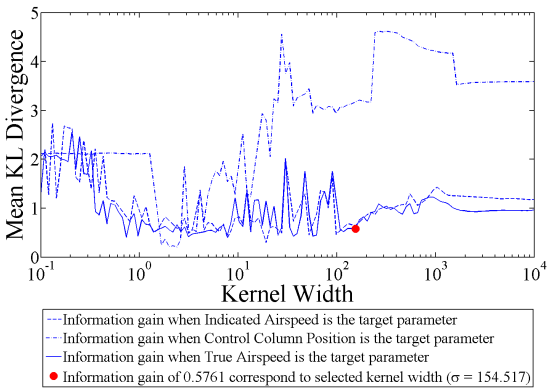
(a) Mean AUC vs. σ for Speed Violation



(b) Mean KS Statistic vs. σ for Speed Violation



(c) Mean AIC vs. σ for Speed Violation



(d) Mean Information Gain vs. σ for Speed Violation

Figure 6. Feature and Kernel Width Tuning and Selection

For our experiment we will target an adverse event that is a speed violation in which the pilot must slow down the aircraft to an indicated speed below 250 knots, upon descending below an altitude of 10,000 ft, according to FAA regulations. Thus, to establish ground truth for both validation via the tuning loop shown in Fig. 3 and testing, we will use these criteria. 6 nominal flights previously identified not to experience this event were used as training data to form the basis of generating both SVR and LDS model parameters. In applying SVR, 29 non-redundant continuous parameters were identified to be used as predictors of a selected target parameter. 4 different flights which contain examples of the speed violation event were used

as validation data to compute relevant metrics to aid in target parameter selection and assessment of model fidelity and class discriminability (*cf.* Fig. 3).

Out of all of the candidate target parameters under consideration identified in previous work,[8] (indicated airspeed, true airspeed, and control column position), the true airspeed parameter yielded the best performance as measured by the worst case mean validation AUC (over all 4 flights). This can be seen in panel (a) of Fig. 6, which illustrates the validation AUC as a function of kernel width (σ) for the three candidate target parameters, all shown in red with differing line styles. The solid red line corresponds to the mean validation AUC when using true airspeed as the target parameter. It is clear that out of the three candidate target parameters shown, the worst case mean validation AUC was much higher for the true airspeed parameter than when using any of the other target parameter candidates. One of the reasons for this observation may be due to the fact that the 29 parameters were indeed good predictors of the true airspeed, which is of interest for this specific event. Recall also from Sec. II that candidate target parameters not being used in that role are included in the feature parameter set as a predictor of the candidate selected as the target parameter. Thus, indicated speed and control column position are good predictors of the true airspeed, which meets with intuition.

The model parameters previously trained were used for validation, as well as the same parameters that were used to parameterize and detect critical events for all methods to be compared, shown in Table 1. Fig. 6 illustrates the results of kernel width tuning and selection, depicting how the KS statistic, mean information gain, and AIC vary over various values of the kernel width for the all candidate target parameters. The results corresponding to the selected target parameter: true airspeed, however, is now the focus of kernel width selection. The best kernel width value is highlighted in Fig. 6 with a red circle in each panel, again by minimizing the training KS statistic and training AIC (shown as blue solid lines in panels (b) and (c)), while maximizing the mean information gain over all 4 flights (shown as a blue solid line in panel (d)). Recall also that the optimization places the most weight on the KS statistic, whereas the AIC and KL divergence metrics are used only to arbitrate final selection of the optimal kernel width. As is clearly shown in panel (b) of Fig. 6, the minimum KS statistic for training data is highlighted with a red circle. The corresponding kernel width value minimizer of $\sigma = 154.5$ does not minimize the training AIC, nor maximize the mean, however we will nonetheless use this value since neither alternate metric provides sufficient arbitration. Furthermore, the validation AUC value corresponding to this value of σ is a very acceptable 0.95.

Table 2. Speed Violation Prediction Experiment

	AUC	$P_{fa}{}^{\dagger}$	$P_d{}^{\dagger}$	$T_d{}^{\dagger,\S}$	$P_{fa}{}^{\ddagger}$	$P_d{}^{\ddagger}$	$T_d{}^{\ddagger,\S}$
Standard Exceedance	0.9857	N/A	N/A	N/A	0	0.57	71
Optimal Level-Crossing Prediction	0.9874	N/A	N/A	N/A	0	0.62	63
SPRT Positive Mean Test	0.796	0.62	0.978	5	0.69	1	0
SPRT Negative Mean Test	0.998	0	0	N/A	0	0	N/A
SPRT Nominal Variance Test	0.94	0	0.18	0	0.1	0.8	0
SPRT Inverse Variance Test	0.39	0	0	N/A	0.28	0.01	161

† Based upon theoretically selected threshold

‡ Based upon empirically selected threshold

§ Time elapsed between onset of speed violation and first correct detection in seconds

Thus, the associated model parameters will be applied for the test data, which consists of a single flight that contains an example of the same event that was identified for the 4 validation flights. The testing results are summarized in Table 2. Alarm system design for all methods shown in this table are based upon 4 averaged validation ROC curves corresponding to the 4 validation flights, and choice of an upper bound of $P_{fa} \leq 0.01$. Thus, it is clear that the optimal level-crossing predictor outperforms the standard exceedance method for correct detections. Both methods yield a perfect zero P_{fa} false positive rate. Note that there are both theoretically and empirically selected threshold-based results for all SPRT tests. The reason for this is due to the fact that the false alarm and correct detection rate results are quite poor when using Eqn. 37 to establish the theoretically optimal threshold, as shown in the second two columns of Table 2. Better results are achieved by using the same ROC curve-based alarm design technique as was used for the competing methods, resulting in an empirically selected threshold. Thus, the basis for comparison with other methods

is well grounded, since alarm design is conducted in the same fashion.

Taking all of the empirically selected threshold-based SPRT tests into account simultaneously, the results are mixed due to poor false alarm rates for the positive mean and inverse variance shift tests, and poor correct detection rates for the negative mean and inverse variance shift tests. The most competitive out of the four SPRT tests is therefore the nominal variance shift test, which yields a correct detection rate of 0.8, however it achieves this at the expense of a false positive rate of 0.1. For the standard exceedance and optimal level-crossing predictor methods, the resulting false positive rates are both 0, while the optimal level-crossing predictor method yields a higher correct detection of 0.63 in comparison to 0.57 for the standard exceedance method. Although both detection rates are lower than the one achieved by the SPRT nominal variance shift test, the optimal level-crossing predictor yields the highest detection rate for methods having the lowest possible false alarm rate.

Using empirically selected thresholds, it was also found that the first correct detection of the optimal level-crossing predictor came 8 seconds prior to the first correct detection for the standard exceedance method, and 98 seconds prior to the inverse variance shift SPRT test. However, the first correct detection for the positive mean and nominal variance SPRT tests came 63 sec prior to the optimal level-crossing predictor, at first onset of the violation. We have also determined that the optimal level-crossing predictor runs in less than a second during the testing phase, which computationally far exceeds our objective of running in near real-time, due to the 11 min period spanned by the test data. This does not include training time that involved building the kernel function matrix.

VII. Conclusion

This paper has provided preliminary results for the optimal prediction of adverse events in aviation data, demonstrating that an anomaly prediction problem can be recast as an optimal level-crossing prediction problem. We have demonstrated that the optimal level-crossing predictor yields a better correct detection rate and real-time advance predictive capability than using a method based upon standard exceedances. Even though these results are less desirable than one of the SPRT tests normally associated with MSET, overall it is competitive with the SPRT method. This stems from the fact that when the SPRT tests are *all* considered in tandem to provide broader coverage than testing a single hypothesis, the results can be mixed and thus uninformative without subsequent arbitration. However, the optimal level-crossing predictor has no such drawback.

Another conclusion can be drawn from the apparent relative advance predictive capability reported in Sec. VI and the results presented in Table 2. The test AUC results, although they quantify the probability of correctly ranking two randomly selected nominal and anomalous data points respectively for each method, are not necessarily good indicators of which method yields the lowest false alarm rate and highest correct detection rate. This is primarily due to the fact that the thresholds were empirically chosen based upon *validation* data. Furthermore, neither the AUC nor the false alarm or correct detection rates provide good indications of which method will provide the earliest correct indication of the anomalous event. This is due to the fact that the AUC, false alarm, and correct detection rates are aggregate indicators, and do not represent a measure of real-time predictive capability. More research on the optimization of advance predictive capability as related to the optimal level-crossing predictor will be conducted in the future.

In future work we will also extend the scope of the investigation to incorporate multivariate rather than univariate state estimation which will aid in event localization and isolation efforts. We will also consider multiple adverse events in parallel by selecting appropriate target parameters that are relevant to aviation safety. Specifically, we will target those adverse events that provide for a fairer basis for comparison in the future. The adverse event studied here yielded violations that were always initiated at $k = 0$. Thus, advance prediction of the onset of the event in real-time was not feasible and reliance on the reporting of false dismissals and alarms were necessary in order to demonstrate superior performance.

The effect of adjusting critical event parameterization and the art of model order selection both affect final performance and will also be studied in future work. Namely, the free alarm design parameters L (the critical threshold), d (the prediction horizon) have been fixed in this paper. These will both be investigated in depth in future studies, in addition to the model order, n, which is selected heuristically here. Lastly, for future comparisons, implementation of the baseline SPRT detection portion of the MSET will be patterned more closely after methods used in practice.[4] Specifically, repeated application of the SPRT test (also known as the CUSUM algorithm), will be used as an enhancement to allow for continuous restarting of the tests,

setting the SPRT indices to 0 after each restart.

Acknowledgments

We would like to thank the Integrated Vehicle Health Management Project of the Aviation Safety Program for providing support during the period of time that this research was conducted. We would also like to thank Dr. Nikunj Oza and Dawn McIntosh for reviewing the paper, Dr. Nhan Nguyen for encouraging submission of the work to this venue, and Robert Lawrence for his invaluable domain expertise.

References

[1]Hayden, S., Oza, N., Mah, R., Mackey, R., Narasimhan, S., Karsai, G., Poll, S., Deb, S., and Shirley, M., "Diagnostic Technology Evaluation Report For On-Board Crew Launch Vehicle," Tech. Rep. 214552, National Aeronautics and Space Administration, 2006.

[2]Martin, R. A., "A State-Space Approach to Optimal Level-Crossing Prediction for Linear Gaussian Processes," *IEEE Transactions on Information Theory (preprint, currently under review)*, 2010.

[3]Martin, R., "An Investigation of State-Space Model Fidelity for SSME Data," *Proceedings of the International Conference on Prognostics and Health Management*, IEEE, October 2008.

[4]Bickford, R., "MSET Signal Validation System Final Report," Technical report, NASA Contract NAS8-98027, August 2000.

[5]Smola, A. J. and Schölkopf, B., "A Tutorial on Support Vector Regression," Tech. rep., Statistics and Computing, 2003.

[6]Martin, R. A., *Aerospace Technologies Advancements*, chap. Evaluation of Anomaly Detection Capability for Ground-Based Pre-Launch Shuttle Operations, IN-TECH, January 2010, pp. 141–164.

[7]Bengtsson, T. and Cavanaugh, J. E., "An Improved Akaike information criterion for state-space model selection," *Computational Statistics and Data Analysis*, Vol. 50, No. 10, 2006, pp. 2635–2654.

[8]Das, S., Matthews, B. L., Srivastava, A. N., and Oza, N. C., "Multiple Kernel Learning for Heterogeneous Anomaly Detection: Algorithm and Aviation Safety Case Study," *The* 16^{th} *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Vol. To appear, 2010.

[9]Zavaljevski, N. and Gross, K. C., "Support Vector Machines for Nuclear Reactor State Estimation," Tech. rep., Argonne National Laboratory, 2000.

[10]Murphy, K. P., "The Bayes' Net Toolbox for MATLAB," *Computing Science and Statistics*, Vol. 33, 2001.

[11]Derek, M., Isaacs, K., McElfresh, D., Murguia, J., Nguyen, V., Shao, D., Wright, C., and Bremer, M., "Anomaly Detection with Multi-dimensional State Space Models," Tech. rep., San Jose State University, February 19, 2010.

[12]Overschee, P. V. and Moor, B. D., *Subspace Identification for Linear Systems, Theory, Implementation, Applications*, Kluwer Academic Publishers, 1996.

[13]Overschee, P. V. and Moor, B. D., "Subspace algorithms for the stochastic identification problem," *Automatica*, Vol. 29, No. 3, 1993, pp. 649–660.

[14]Overschee, P. V. and Moor, B. D., "N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems," *Automatica*, Vol. 30, No. 1, January 1994, pp. 75–93.

[15]Favoreel, W., Moor, B. D., and Overschee, P. V., "Subspace state space system identification for industrial processes," *Journal of Process Control*, Vol. 10, No. 2-3, 2000, pp. 149 – 155.

[16]Larimore, W. E., "Canonical variate analysis in identification, filtering, and adaptive control," *Proceedings of the 29th Conference on Decision and Control*, Vol. 2, IEEE, December 1990, pp. 596 –604.

[17]Ljung, L., *System Identification: Theory for the user*, Prentice Hall, 2nd ed., 1999.

[18]Siddiqi, S., Boots, B., and Gordon, G., "A Constraint Generation Approach to Learning Stable Linear Dynamical Systems," *Advances in Neural Information Processing Systems 20*, edited by J. Platt, D. Koller, Y. Singer, and S. Roweis, MIT Press, Cambridge, MA, 2008, pp. 1329–1336.

[19]Lacy, S. L. and Bernstein, D. S., "Subspace identification with guaranteed stability using constrained optimization," *IEEE Transactions on Automatic Control*, Vol. 48, No. 7, July 2003, pp. 1259 – 1263.

[20]Antunes, M., Turkman, A. A., and Turkman, K. F., "A Bayesian Approach to Event Prediction," *Journal of Time Series Analysis*, Vol. 24, No. 6, November 2003, pp. 631–646.

[21]Lewis, F., *Applied Optimal Control & Estimation: Digital Design & Implementation*, Prentice Hall, Inc., 1992.

[22]Hines, J. W. and Seibert, R., "Technical Review of On-Line Monitoring Techniques for Performance Assessment Volume 1: State-of-the-Art," Tech. Rep. NUREG/CR-6895, University of Tennessee, January 2006.

[23]Bradley, A. P., "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, Vol. 30, No. 7, 1997, pp. 1145 – 1159.