

A novel Architecture for on-line Failure Prognosis using Probabilistic Least Squares Support Vector Regression Machines

Taimoor Khawaja¹, Dr. George Vachtsevanos²

^{1,2} *Intelligent Control Systems Laboratory, Georgia Institute of Technology, Atlanta, GA, 30332, USA*

taimoor@gatech.edu

gjv@ece.gatech.edu

ABSTRACT

The ability to forecast machinery failure is vital to reducing maintenance costs, operation downtime and safety hazards. Recent advances in condition monitoring technologies have given rise to a number of prognostic schemes (both model-based and data-driven) that attempt to forecast machinery health by constructing health propagation models for the underlying systems. In particular, algorithms that use the data-driven approach learn models directly from the data, rather than using a hand-built model based on human expertise. This paper introduces a novel architecture for data-driven Failure Prognosis of complex non-linear systems using Least Squares Support Vector Regression Machines (LSSVR). An adaptive recurrent LSSVR machine is proposed and augmented with a Bayesian Inference scheme that allows probabilistic estimates of future health deterioration. Extensions for efficient multi-step long-term prognostics and Remaining Useful Life (RUL) calculation are suggested. Data from a seeded fault test for a UH-60 planetary gearbox plate is used to test the online performance of the prognostics algorithm.

1. INTRODUCTION

Rapid growth in modern day industry, such as nuclear power plant, automobile and shipbuilding, industry, has resulted in the development of increasingly complex non-linear systems with complicated controls and feedback loops. Today's complex and advanced machines demand highly sophisticated and costly maintenance strategies leading to much improved instrumentation and monitoring capabilities. Current maintenance strategies have progressed from breakdown maintenance, to preventive maintenance, then to condition-based maintenance (CBM) managed by experts, and lately towards a futuristic view of intelligent predictive maintenance systems in effect extending lifecycles, lowering downtimes and improving

mission management capabilities (Heng 2009). This just-in-time predictive maintenance approach is referred to as system Prognostics in the reliability industry.

Essentially, the approaches for prognostic reasoning are based on one of the following three methodologies: (1) physical modeling, (2) model-driven statistical learning techniques and (3) data-driven statistical learning modeling (Kothamasu 2006), (Jardine, 2006). Physical models are designed by experts and validated using large sets of data. Generally, they have higher accuracy, but have characters of higher operation costs and inflexibility, which can only be applied to specific types of components (Brotherton, 2000). The model-driven statistical learning methods assume that both the operational data and a mathematical model are available. In comparison, the data-driven statistical learning models are developed from collected input/output data. They are based on the selected features that correlate with the failure progression and produce the desired output prediction of the time-to-failure (TTF) based on a training process. Hybrid approaches combining data-driven approaches with model-based techniques have also been suggested.

Most real-world systems are non-linear in nature. Prognosis for the prediction of chaotic or non-linear time-series signals remains a huge challenge. Moreover, prognostic schemes must contend with multiple sources of error such as noise, modeling inconsistencies and degraded sensor fidelity which contributes to the uncertainty associated with prognosis. Any viable solution must therefore be able to model non-linear systems and incorporate

methods to effectively deal with the issue of uncertainty.

Many existing approaches to data-driven prognosis use artificial neural networks (Wang, 2003), (Brotherton, 2000), (Parker, 1993). Artificial neural networks are a type of model, based loosely on the neural structure of the brain, in which the weights of the connections among neurons are automatically adjusted to maximize the fit of the model to the data. Although ANNs have been shown to successfully model many non-linear systems, they also have their limitations. One of the main issues with ANNs is the high number of free parameters that need to be tuned in order to get good results. Secondly, ANNs are not guaranteed to converge to the optimal solution and are prone to the issue of local optima. Finally, with the exception of a few ANN prognostic schemes, most black-box ANN strategies output point predictions incapable of representing process uncertainties. In order to overcome these limitations many researchers study Support Vector Machines (SVM) as an alternate to ANNs. Support Vector Machines (SVM) are an elegant classification and regression scheme that employ Structural Risk minimization (SRM) principles as opposed to Empirical Risk Minimization (ERM) used by ANNs which makes them less prone to overfitting. Additionally, whilst ANNs can suffer from multiple local minima, the solution to an SVM is global and unique (Chen, 2005). It is thus that SVMs are often applied to prediction of time-varying nonlinear systems in the form of Support Vector Regression Machines (SVRs).

In this paper, we introduce a prognostics architecture for real-time Prognostics and Health Management (PHM) systems based on the Least Squares formulation of SVR (LSSVR) framework. The core LSSVR algorithm is an efficient reformulation of the classical SVR scheme and is therefore suitable for online real-time prognostics systems (Suykens, 1999). A Bayesian Inference System is associated with the LSSVR (B-LSSVR) thus providing a natural framework for probabilistic interpretation of prediction uncertainty (Gestel, 2002). The rest of the paper is organized as follows. Section 2 discusses some related background on LSSVR formulation. The Bayesian Inference Model associated with LSSVRs is also discussed briefly. In section 3, the proposed prognostics architecture is introduced while its individual modules are discussed at length in Section 4. Section 5 addresses the evaluation of Remaining useful life (RUL) based on the proposed scheme. In section 6, we present prognostics results obtained by applying the scheme to a progressing crack on a planetary gear plate on-board the UH-60 BlackHawk aircraft and complete the discussion with concluding remarks and future research in section 7.

2. LSSVR AND BAYESIAN INFERENCE METHODS FOR LSSVR

SVM (and its regression counterpart SVR) is based on the principles of statistical learning theory, or VC theory (Vapnik, Chervonenkis) developed over the last several decades. Given training data $\{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\} \subset X \times \mathcal{R}$, where X denotes the space of input patterns, the goal is to find a function $f(x)$ which has at the most ϵ deviation from the actually obtained targets y_i for all the training data, and at the same time is as flat as possible. That function $f(x)$ can be described according to Equation 2-1:

$$f(x) = \sum_{i=1}^l \langle w, \varphi(x_i, x) \rangle + b \quad 2-1$$

Flatness in this case implies that we seek a small w . Mathematically, the primal SVR problem can be defined according to Equation 2-2.

$$\begin{aligned} \min & \left\{ \frac{1}{2} w^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \right. \\ \text{s.t.} & \left. \begin{cases} y_i - \langle w, \varphi(x_i) \rangle - b \leq \epsilon + \xi_i \\ \langle w, \varphi(x_i) \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \right. \end{aligned} \quad 2-2$$

where the slack variables ξ_i, ξ_i^* are introduced to allow for some errors in order to cope with otherwise infeasible constraints of the optimization problem. The constant $C > 0$ determines the trade-off between the flatness of f and the amount up to which deviations larger than ϵ are tolerated.

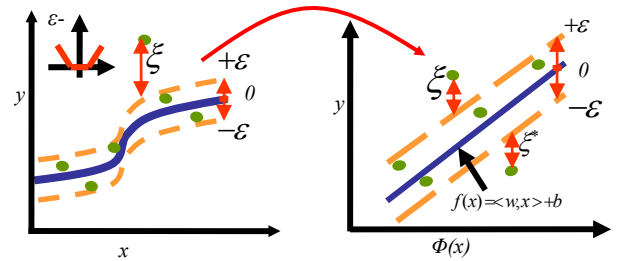


Figure 2-1 Non-Linear extension of SVR using the kernel mapping

Non-linear regression for SVRs is achieved by preprocessing the training patterns x_i by a map $\varphi: X \rightarrow \mathcal{T}$ that transforms the input space into a higher dimensional feature space \mathcal{T} , where linear SVR regression can be applied to estimate the underlying function (Nillson, 1965). Figure 2-1

illustrates how the use of a kernel can simplify the regression process.

SVR, being a derivative of SVM, is a quadratic programming (QP) problem and its solution is very time consuming. The LS-SVM is an efficient reformulation of the SVM because it is solved through a set of linear equations (Sollich et. Al., 2001) and is therefore very well suited for solving on-line real-time regression problems. The primal LSSVR problem can be defined according to Equation 2-3.

$$\begin{aligned} \min_{w,b} I(w,b) &= \frac{\mu}{2} \|w\|^2 + \frac{1}{2} \sum_{i=1}^l \zeta_i e_i^2 \\ &= \mu E_W + \sum_{i=1}^l \zeta_i E_{D,i} \quad 2-3 \\ \text{s.t.} \quad y_i &= \langle w, \varphi(x_i) \rangle + b + e_i \end{aligned}$$

where the tradeoff between regularization and training error is determined by the ratio $\gamma_i = \frac{\zeta_i}{\mu}$ ($i = 1, 2, \dots, N$).

The LSSVR problem in the dual space corresponds to solving the following linear system.

$$\begin{bmatrix} 0 & \bar{1}^T \\ \bar{1} & Q \end{bmatrix} \begin{bmatrix} b \\ \bar{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad 2-4$$

where $y = [y_1, \dots, y_N]$, $\bar{1}_v = [1; \dots; 1]e = [e_1; \dots; e_N]$ and

$$D_\gamma = \text{diag}(\gamma_1, \dots, \gamma_N), Q = \Omega + D_\gamma^{-1} \text{ and } \Omega = K(x, x_i)$$

By dividing the linear system above into two linear systems with positive definite data matrices as in (Suykens, 2000), iterative methods such as the Hestenes-Stiefel Conjugate Gradient algorithm can be applied to solve large scale problems efficiently.

The Bayesian Inference model for LSSVR is defined similar to its SVM counterpart. Given N data points $D = \{(x_i, y_i)\}_{i=1}^N$ and hyper-parameters μ and $\zeta_{1:N}$ for the model, a probabilistic interpretation of Equation 2-3 is obtained by applying Baye's rule as follows:

$$\begin{aligned} P(w,b|D, \log \mu, \log \zeta_{1:N}, H) &= \\ \frac{P(D|w,b, \log \mu, \log \zeta_{1:N}, H) P(w,b| \log \mu, \log \zeta_{1:N}, H)}{P(D| \log \mu, \log \zeta_{1:N}, H)} \quad 2-5 \end{aligned}$$

Where the model H corresponds to the kernel function K, possibly with kernel parameters. Assuming that the prior is independent of the hyperparameters $\zeta_{1:N}$, w and b are independent variables and that the noise is Gaussian

distributed additive noise e_i ($i = 1, \dots, N$) with zero mean and variance ζ_i^{-1} , the posterior can be redefined as follows:

$$\begin{aligned} P(w,b|D, \log \mu, \log \zeta_{1:N}, H) &\propto \\ \exp(-\frac{\mu}{2} \langle w, w \rangle) \exp(-\sum_{i=1}^N \frac{\zeta_i}{2} e_i^2) \quad 2-6 \end{aligned}$$

The maximum a Posteriori model parameters w_{MP} and b_{MP} are obtained by taking the negative logarithm of Equation 2-6 which corresponds to solving the optimization problem in Equation 2-4. The moderated output (which is also the mean of the posterior) is obtained using the Equation below:

$$z_{MP} = \langle w_{MP}, \varphi(x) \rangle + b_{MP} = \sum_{i=1}^N \alpha_i K(x, x_i) + b_{MP} \quad 2-7$$

Since z is a linear transformation of the Gaussian distributed model parameters w and b , the variance σ_z^2 in the feature space is given as follows:

$$\begin{aligned} \sigma_z^2 &= E\{(z - z_{MP})^2\} \\ &= E\left\{\left[\langle w, \varphi(x) \rangle + b - \left(\langle w_{MP}, \varphi(x) \rangle + b_{MP}\right)\right]^2\right\} \quad 2-8 \\ &= \psi(x)^T H^{-1} \psi(x) \end{aligned}$$

with $\psi(x) = [\varphi(x); 1]$. Notice that the computation is carried out without explicit knowledge of the mapping $\varphi(x)$. The interested reader is referred to (Gestel, 2001) for detailed derivations. The Bayesian Inference scheme allows us to represent the prognosis results in the form of a posterior probability. In addition, this framework establishes an adaptive scheme which continuously tunes model hyperparameters and helps in model estimation (Gestel, 2001).

We employ the B-LSSVR algorithm as a basic building block for the prognosis algorithm. For long-term prediction, the posterior probability distribution with mean z_{MP} and variance σ_z^2 is sampled using Monte Carlo methods in order to determine possible failure progression trajectories in a way that manages the curse of dimensionality always associated with prognosis.

3. PROGNOSIS ARCHITECTURE

Figure 3-1 proposes an architecture for fault diagnosis and failure prognosis of complex non-linear systems.

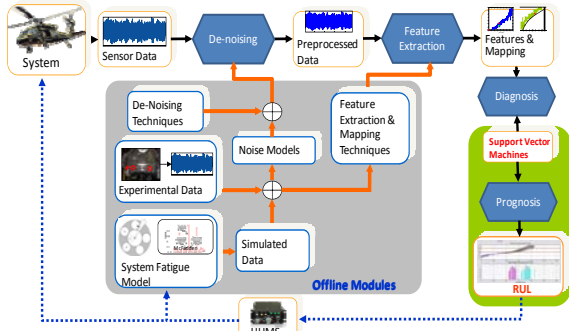


Figure 3-1 Architecture for fault diagnosis and failure prognosis

The architecture groups modules as offline and online modules. Offline analysis provides critical parameters integrated into online operations of signal de-noising and feature-to-fault mapping besides other possibilities. Online modules include de-noising/signal preprocessing, feature extraction, fault diagnosis and failure prognosis. In this paper, we focus on one particular module - the online prognostics module.

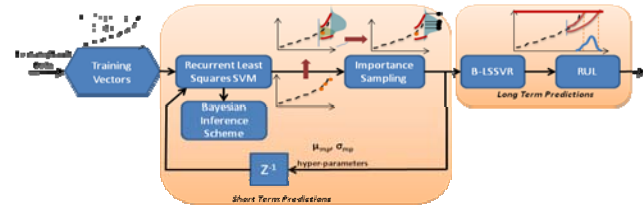


Figure 3-2 A real-time prognosis scheme based on Bayesian LSSVR machines

An overview of the proposed prognosis architecture is presented in Figure 3-2. At each time step, the stream of incoming feature/fault progression data is used to generate a sequence of training vectors used as input to a recurrent LSSVR predictor. Tied together with the Bayesian Inference scheme introduced in Section 2, the short-term predictor estimates a one-step ahead (possibly) multi-modal distribution representing fault progression. An importance sampling step following that uses the estimated distribution to determine a finite set of most-likely next-step predictions for the evolving fault. The scheme is employed recursively in order to infer multi-step prediction trajectories along with associated multi-modal distributions and uncertainty bounds.

Although this prediction scheme can be employed to generate long-term predictions, this process may require significant computational time especially when time-horizons are large and near-real-time performance is

expected. This paper proposes a scheme to circumvent this problem by splitting the prognostics scheme into a short-term and a long-term prognostics problem as highlighted in Figure 3-2. Accurate trajectories are predicted by using the more complex, short term prediction scheme based on the recurrent BLSSVR. On the other hand, the long-term prognosticator can have multiple forms. One solution would be to use the same trained recurrent LSSVM without further adaptation of the hyper-parameters for long term predictions. However, recurrent LSSVMs over multiple trajectories would still bear a considerable overhead. Alternatively, we propose to use a simple and more computationally efficient scheme that implements a BLSSVR machines with an exponential kernel function. Here we assume that the prognostics module is activated after the occurrence of a fault and that evolving faults are monotonic in nature, i.e. the system goes from bad to worse. The short-term and long-term predictors are discussed at length in Section 4.

4. BAYESIAN LSSVR FOR REAL-TIME LONG-TERM PROGNOSIS

Consider a nonlinear time series with samples $\{x_1, x_2, \dots, x_n\}$ sequentially collected from a deteriorating system. In order to predict the future health $\{x_{n+1}, x_{n+2}, \dots\}$ of the system using LSSVRs, one trains a recurrent model of the system with the following training data:

$$x = \begin{bmatrix} x_1 & x_2 & \dots & x_p \\ x_2 & x_3 & \dots & x_{p+1} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x_{n-p} & x_{n-p+1} & \dots & x_{n-1} \end{bmatrix}, y = \begin{bmatrix} x_{p+1} \\ x_{p+2} \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \quad 4-1$$

The corresponding LSSVR machine is similar to the one described in Equation 2-4 where the matrices are defined as follows:

$$\begin{bmatrix} 0 & \bar{1}^T \\ \bar{1} & Q_{n-p} \end{bmatrix} \begin{bmatrix} b_{n-p} \\ \bar{\alpha}_{n-p} \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad 4-2$$

where

$$Q_{n-p} = \Omega + D_y^{-1}$$

$$= \begin{bmatrix} K(x_1, x_1) + 1/\gamma & \dots & K(x_{n-p}, x_1) + 1/\gamma \\ \vdots & \ddots & \vdots \\ K(x_1, x_{n-p}) + 1/\gamma & \dots & K(x_{n-p}, x_{n-p}) + 1/\gamma \end{bmatrix} \quad 4-3$$

The LS-SVM regression model for nonlinear time-series prediction is then expressed using Equation 4-4:

$$y(x_{test}) = \sum_{i=1}^{n-p} \alpha_i K(x_i, x_{test}) + b \quad 4-4$$

If $x_{test} = \{x_{n-p+1}, \dots, x_n\}$, then $y(x_{test}) = \hat{x}_{n+1}$ is the one-step ahead predicted value. This recurrent predictor is combined with the Bayesian Inference model for robust prognostics as discussed below.

At $T = n$, the B-LSSVR algorithm approximates the posterior distribution (defined by the mean z_{MP} and variance σ_z^2) representing the health of the system. In order to generate multiple trajectories from this single distribution, one needs to sample the posterior. Obviously, the choice of the sampling method influences the accuracy of predicted future paths. The importance of an intelligent sampling method is further motivated by the fact that it can act as an important tool in managing the *curse of dimensionality* generally associated with long term prediction. Obviously, if we take each sample in the estimated distribution as a possible future prediction, we will end up with an NP complex problem (Cooper, 1990). We take a page out of the *Markov Chain Monte Carlo (MCMC) methods* for approximate inference and use an Importance Sampling algorithm that takes the information available in the distribution and redistributes it proportionally on the state space as illustrated in

Figure 4-1 by using an acceptable number of samples, N_s (Optimal N_s is still an open subject to research) in the process (Doucet, 2005). Thus, a set of most likely one-step ahead prediction points is obtained as represented in Figure 4-2.

Consequently, these prediction points are combined with the training data vectors to create multiple prediction paths originating from a single progression as shown in Figure 4-2 Bayesian LS-SVR based multi-step prediction distribution and IS

Each trajectory is retrained using the same B-LSSVR algorithm to produce additional distributions representing possible predictions for time $T + 2$ which are combined to

derive the posterior distribution as a mixture of Gaussians. Note that the mixture of Gaussians produces a multi-modal posterior prediction distribution. It must also be noted that although the underlying process for posterior estimation for each predicted trajectory is indeed entirely Gaussian in nature, the nature of the cumulative posterior distribution as presented in Figure 4-2 is a function of both the means and variances of the individual Gaussian distributions. Indeed, it has been shown that any arbitrarily complex distribution can be estimated using a mixture of Gaussians with appropriate bandwidths (Turlach, B.A., 1993). Additionally, a certain amount of randomness is also built in the Importance Sampling algorithm in order to facilitate exploration and also to mitigate the chances of artificial convergence. The combined B-LSSVR and Importance Sampling step is employed iteratively in order to predict fault evolution in the long-term.

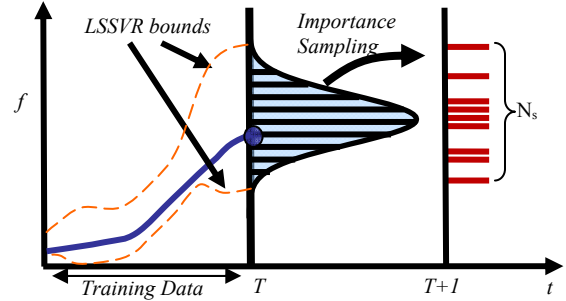


Figure 4-1 B-LSSVR based one-step ahead prediction distribution and IS

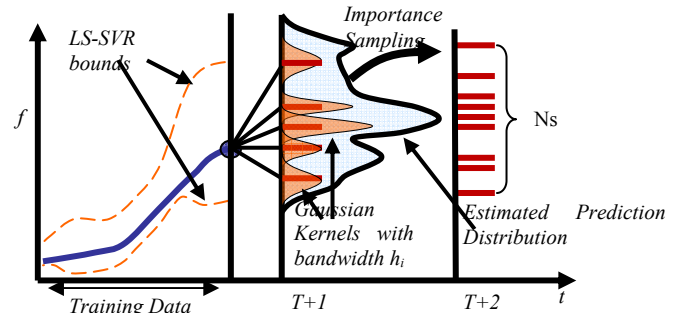


Figure 4-2 Bayesian LS-SVR based multi-step prediction distribution and IS

In summary, the proposed implementation considers non-linear processes with non-gaussian noise where the time varying parameters are fine tuned iteratively with new incoming measurements. We start with an initial model H (the kernel parameter) and initial hyper-parameters μ and $\zeta_{1:N}$. At the first level, model H is combined with the incoming measurements D to provide an estimated distribution of the predicted fault growth using

Equation 2-7 and Equation 2-8. This distribution is sampled using an Importance Sampling algorithm to derive possible future paths for fault growth. Each path is then retrained using the B-LSSVR model. The second inference level tunes the hyperparameters; μ and ζ_i . For multistep predictions, the same algorithm is iterated. It is thus that the algorithm is *adaptive* in nature while the fact that the estimated distribution is sampled using no more than N_s samples allows us to manage the *curse of dimensionality* associated with long-term prognosis.

5. REMAINING USEFUL LIFETIME (RUL)

Besides characterizing the time evolution of impending failures, the second important task for machine prognostics is estimating the Remaining Useful Life (RUL) of the system. RUL is the time left for the normal operation before a breakdown occurs or machine condition reaches the critical failure threshold. At a given time instance, the algorithm described in Section 3 accurately forecasts a set of prediction paths that describe the possible fault growth trends over the long-term prediction horizon as shown in Figure 5-1.

Due to the fact that probabilistic predictive schemes are very time consuming, many prognosis schemes resort to limited steps of model refinement in essence accounting for modeling uncertainties and various noise sources contaminating the signal. This refined model is used for accurate short term predictions which are then projected over long-term prediction horizons using simpler models. We propose to use a two-level scheme by first ascertaining high confidence prediction paths using the core recurrent B-LSSVR algorithm over the short-term horizon. Instead of using simpler regression schemes, the filtered future trajectories are used to train regression models using the core LSSVR algorithm. The efficient combination of recurrent B-LSSVR for short term prediction and LSSVR for long term prediction results in a set of intelligently filtered most-likely paths which in combination with the definition of critical thresholds is used to estimate the RUL pdf, also referred to as the Time-to-Failure (TTF) pdf.

In statistics, kernel density estimation (KDE) (or Parzen window method, named after Emanuel Parzen) is a non-parametric way of estimating the probability density function of a random variable. Given some i.i.d. data $x = \{x_1, x_2, \dots, x_n\}$ about a sample of a population, kernel density estimation makes it possible to extrapolate the data to the entire population (Parzen, 1962) according to Equation 5-1.

$$f_h(x) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x-x_i}{h}\right) \quad 5-1$$

where K is some kernel and h is a smoothing parameter called the bandwidth. In order to estimate the RUL distribution based on long-term predicted trajectories, we employ KDE with a gaussian kernel whereby the variance is controlled indirectly through the parameter h .

Once the probability density function of the RUL is estimated, other important attributes – such as expectations and 95% confidence intervals – may be computed. We apply our prognostics scheme on data collected from a seeded fault test for progressively degrading planetary gear from a UH-60 Blackhawk aircraft in the following section.

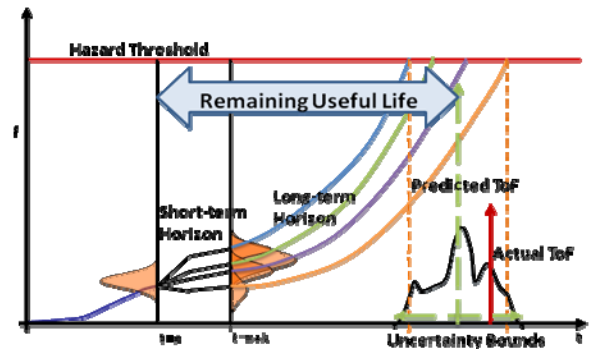


Figure 5-1 Long-term Prediction, Time of Failure (ToF), Uncertainty Bounds and Remaining Useful Life (RUL)

6. CASE STUDY: CRACK GROWTH ON A UH-60 PLANETARY GEARBOX

A UH-60 Blackhawk gearbox with a growing axial crack fault on the gear plate was chosen as a real-world test case (Saxena, 2005). The research team designed a Finite Element ANSYS model of the plate, generated artificial vibration data based on the model and inferred several features from it which could reflect the growth pattern of a simulated fault. An overview of the system Finite Element Model (ANSYS) and its equivalent mechanical layout is shown in Figure 6-1.

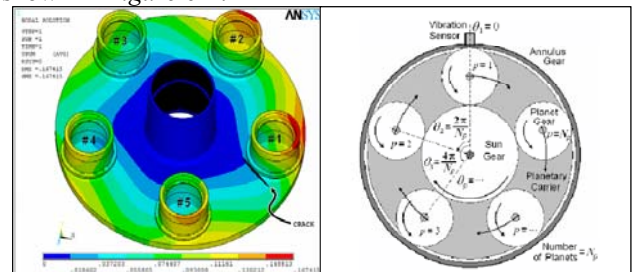


Figure 6-1 (a) ANSYS model of the gearbox plate (b) Mechanical layout

A developing crack close to one of the planetary gears as shown in *Figure 6-1(a)* can lead to a critical failure condition in the aircraft. With the purpose of testing the feasibility and efficiency of diagnostics and prognostics algorithms, a seeded fault test was conducted to collect fault data under a fixed known loading profile. Data is collected in terms of Ground-air-Ground (GAG) cycles representing time. Raw vibration signals were first denoised to get rid of artifacts and environmental noise. Features identified during the modeling phase were used to extract fault growth patterns from the denoised data (Wu, 2004). One feature, in particular, referred to as the Side-Band Ratio (SBR) correlates remarkably well with the actual fault evolution and is used for the purpose of this research. *Figure 6-2* shows how the feature varies as the crack grows with time till the gear-plate breaks into half; a stage considered catastrophic. Subsequently, the hazard threshold for the feature is represented by the red line. The figure also highlights the initial training data required for the recurrent LSSVR algorithm. A set of 85 samples were used to initialize the algorithm.

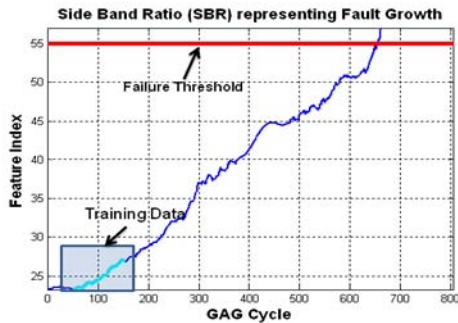


Figure 6-2 Plot of Side Band Ratio (SBR) feature representing fault growth with GAG cycles representing time

After a cold start, the algorithm tunes its parameters according to the incoming feature data. Results for the short-term prediction horizon at GAG cycle 164 are presented in *Figure 6-3*.

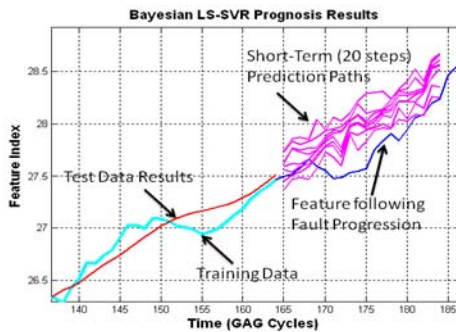


Figure 6-3 Prognostics results for short-term horizon (20 steps). The multiple trajectories successfully track the actual feature progression

This step onwards, the trajectories serve as training data for the long-term prognosticator which uses an exponential kernel function for training purposes. The results from the long-term predictor are shown in *Figure 6-4*.

Based on these projected trajectories and the user-defined failure thresholds, the algorithm estimates the RUL distribution, most probable Time of Failure (ToF_{mp}) and the confidence bounds associated with ToF_{mp} . An RUL distribution drawn from results compiled at the same GAG-cycle is presented in *Figure 6-5*.

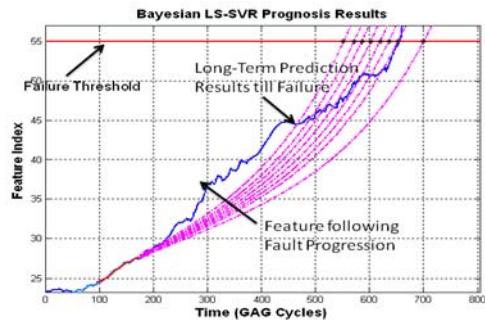


Figure 6-4 Prognostics results for long-term prediction till failure. The results are based on data gathered till GAG Cycle 164

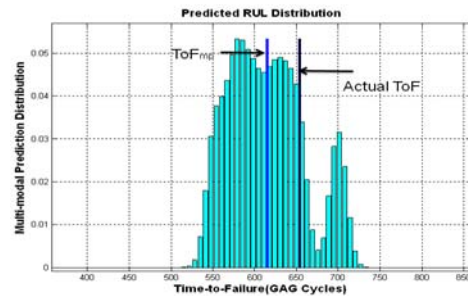


Figure 6-5 Probability Distribution for RUL. The actual failure time lies within the estimated distribution and is reasonably close to the predicted time of failure

7. CONCLUSION

An intelligent architecture for data-driven failure prognosis has been presented based on a recurrent treatment of the Least Squares SVR machines. It is shown that a Bayesian Inference model coupled with the LSSVRs can be used to derive failure evolution trajectories and also to adaptively tune the parameters of the LSSVR model. Further, sampling the output of the recurrent B-LSSVR using an Importance Sampling technique solves the curse of dimensionality associated with Prognostics while preserving prediction accuracy. Finally, for real-time on-line application, the prognostics architecture is extended using an efficient second stage LSSVR

based regression scheme that allows us to construct long-term prediction trajectories for fault evolution, evaluate a posterior prediction distribution and extract statistical information such as the most likely ToF and *uncertainty bounds* associated with it. Test results performed on data set from a progressing crack in a gear-plate suggest that the architecture can be successfully applied to complex non-linear systems experiencing non-gaussian noise. A more comprehensive statistical characterization based on performance metrics suggested for prognostics systems is in the works.

REFERENCES

- Aiwina Heng, Sheng Zhanga, Andy C.C. Tana and Joseph Mathew (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing, Volume 23, Issue 3*.
- R. Kothamasu, S.H. Huang, W.H. Ver Duin (2006), System health monitoring and prognostics—a review of current paradigms and practices, *International Journal of Advanced Manufacturing Technology 28*
- A.K.S. Jardine, D. Lin, D. Banjevic (2006), *A review on machinery diagnostics and prognostics implementing condition-based maintenance, Mechanical Systems and Signal Processing 20*
- T. Brotherton, G. Jahns, J. Jacobs, D. Wroblewski (2000), Prognosis of faults in gas turbine engines, *IEEE Aerospace Conference Proceedings 6*
- Wang, P., & Vachtsevanos, G. (2003), Fault prognostics using dynamic wavelet neural networks. *Artificial Intelligence for Engineering Design and Manufacturing, 15*.
- Brotherton, T., Jahns, J., Jacobs, J., & Wroblewski, D. (2000), Prognosis of Faults in Gas Turbine Engines. *Proceedings of the IEEE Aerospace Conference*, IEEE, New York, Vol. 6, pp. 163–171
- Parker, B.E., Jr., Nigro, T.M., Carley, M.P., Barron, R.L., Ward, D.G., Poor, H.V., Rock, D., & DuBois, T.A. (1993), Helicopter gearbox diagnostics and prognostics using vibration signature analysis. *Proc. SPIE—The International Society for Optical Engineering, Vol. 1965, pp. 531–542*.
- Chen, P.-H., C.-J. Lin, and B. Schölkopf (2005), A tutorial on v-support vector machines. *Applied Stochastic Models in Business and Industry*, 21(2): p. 111-136
- Suykens, J.A.K. and J. Vandewalle (1999), Least Squares Support Vector Machine Classifiers, *Neural Processing Letters*, 9(3): p. 293-300
- Gestel, T.V., et al. (2002), Bayesian Framework for Least-Squares Support Vector Machine Classifiers, Gaussian Processes, and Kernel Fisher Discriminant Analysis. *Neural Computation 14(5): p. 1115–1147*.
- J.A.K., S. and V. J. (2000), Least Squares Support Vector Machines for Classification and Non Linear Modeling, in *Neural Network world.* p. 20-48.
- Van Gestel, T., et al. (2001), Bayesian interpretation of least squares support vector machines for financial time series prediction, in *Proc. of the 5th World Multiconference on Systemics, Cybernetics and Informatics.* p. 254-259.
- Cooper G.F.(1990), The computational complexity of probabilistic inference using Bayesian belief networks, *Artificial Intelligence 42 (2–3) 393–405*.
- Doucet, A. and X. Wang (2005), Monte Carlo methods for signal processing: a review in the statistical signal processing context. *Signal Processing Magazine, IEEE.* 22(6): p. 152 - 170.
- Saxena, A., B. Wu, and G. Vachtsevanos (2005), A methodology for analyzing vibration data from planetary gear systems using complex Morlet wavelets, in *American Control Conference, 2005. Proceedings of the 2005.* p. 4730 - 4735.
- Nilsson, N.J. (1965), *Learning Machines: Foundations of Trainable Pattern-Classifying Systems.* New York: McGraw-Hill.
- Aizerman, M.A., E.M. Braverman, and L.I.R. er. (1964), Theoretical foundations of the potential function method in pattern recognition and learning, in *Automation and Remote Control.* p. 821–837.
- Parzen E. (1962). On estimation of a probability density function and mode, *Ann. Math. Stat. 33, pp. 1065–1076*.
- Wu, B., et al. (2004), Data Analysis and Feature Selection for Fault Diagnosis of Helicopter Planetary Gears, in *IEEE Autotestcon*.
- Vachtsevanos, G. et al. (2006), *Intelligent Fault Diagnosis and Failure Prognosis for Engineering Systems, New Jersey: John Wiley & Sons, Inc.*
- Turlach, B.A. (1993), "Bandwidth selection in kernel density estimation: A review", Discussion Paper 9317, *Institut de Statistique, Voie du Roman*.