Proceedings of the 7th Annual Conference on Information Fusion, Fusion 2005, Vol. 2., pp. 1247 – 1255, 2005

Prognostic Information Fusion for Constant Load Systems

Kai Goebel
GE Global Research
K1-5C4A, One Research Circle,
Niskayuna, NY 12309
USA
goebelk@research.ge.com

Piero Bonissone
GE Global Research
K1-5C32A, One Research Circle,
Niskayuna, NY 12309
USA
bonissone@research.ge.com

Abstract *–This paper describes a process for aggregating different information sources to estimate remaining equipment life. Specifically, the approach presents a rigorous chain of preprocessing, modeling and post-processing steps that arrive at the desired prognostic result. The preprocessing steps deal with data reduction, filtering, and signature amplification. The prediction model applies Adaptive Neuro-Fuzzy Inference System (ANFIS) to the data. The post-processing steps include recursive trending which implicitly forces the prognostic trend to be confirmed before updated estimates are reported. Prognostic false positives and false negatives are introduced as innovative measures that help in assessing the performance of the approach. The method is illustrated using real-life data from industrial web paper breakage prediction.*

Keywords: Prognostic fusion, prognostics, prognosis, decision fusion

1 Introduction

Finding synergy in using different information sources to assess system states has a long tradition within the fields of multivariate statistics and pattern recognition. Recently, the field of information fusion, and more specifically multi-classifier fusion has been recognized as a research area in its own right. Fusing information for prognostic purposes is a fairly new endeavor and will likely lead to the development of new techniques that are specialized to perform related tasks. Prognosis is a key element in equipment health management by providing an estimate for remaining equipment life. Accurate prognosis is difficult to achieve for a number of reasons. These range from the availability of suitable data sets for training of algorithms to the unsolved issue of validating prognostic approaches to the incorporation of future usage information to the need to manage uncertainty to the need to find technology that aggregates diverse information sources and returns a continuous output reflective of remaining life. This paper will address some aspects of the latter issue. Generally, remaining life estimates are conditional on future usage. That is, the estimate must change for real-time estimates as new information about intended usage becomes available. For many systems, future usage is less certain the farther out the prediction is. This means that remaining life will also have a much wider range of possible values and the uncertainty about any specific estimate increases. The situation is slightly better for constant load processes where few process changes are carried out or where the changes are largely the same.

The work reported herein – prediction of the next failure at the wet end of a paper mill – faced the challenge that none of the dozens of observable process and control variables provided a single strong indication for an impending failure. Indeed, there seemed to be not even an apparent consistent weak indicator. Some variables looked as if they could provide a trend for particular failure trajectories, but that was not true for many other failure trajectories. While this may be due to different failure root causes, it might also point to a lack of single variable predictions. With a lack of access to further domain knowledge, the problem could be posed as a fusion problem where the task was how to fuse dozens of process variables to arrive at a reliable failure prediction.

2 Prognostics

Using the definition where prognostics is the estimation of remaining useful component life, the task of providing that estimate can be accomplished using three fundamental different means.

1. The first is extrapolation from past data. Most statistical approaches fall into this category. These include
   - Linear extrapolation
   - Non-linear curve fit (extrapolation)
   - Multi-variable regression
   - Auto-Regressive Moving Average (ARMA, ARIMA) Models
   - Multi-Step Adaptive Kalman Filter (Frelicot, 1996)
   - Variance Analysis (Fuh and Wu, 1995)
   - Particle Filter (Bauer at al., 2002)
   - Least angle regression (LARS) (Efron et al., 2004)
   - Shrinkage methods (Ojelund et al., 2002)
   - Least Absolute Shrinkage and Selection Operator (LASSO). (Tibshirani, 1996).

2. Another means is the use of a model that mimics the dynamics of the system. This involves an in-depth understanding of the processes and can be difficult for very complex systems. However, good models can use anticipated usage conditions and potentially provide an estimate with high confidence. Models are typically built using 1^{st} principles dynamic equations.

3. The third means requires the availability of time series examples from which prognostic patterns can be learned. These methods include

- Dempster-Shafer regression (Petit-Renaud and Denoeux, 2004)
- Fuzzy regression trees (Haskell, 1999)
- ANFIS (Jang, 1993)
- NN (+ CPNN) (Vachtsevanos, 2003)
- Dynamic Wavelet NN (Wang and Vachtsevanos, 2001)
- Forecasting by Pattern and Cluster (Sfetsos and Magro, 2004)

Some classification techniques (if the problem were rephrased as the classification of an impending failure with an associated estimated time to failure) are not as readily applicable in the prognostic context because they provide only categorical output which would need to be artificially converted back into the continuous domain. However, if the answer sought is within the resolution of the answer bins provided, this may be an acceptable approach after all. Another issue is the uncertainty associated with any estimate. If the uncertainty bounds are very wide, it does not make sense to demand a high resolution output. Rather, the output resolution should be commensurate with the residual uncertainty.

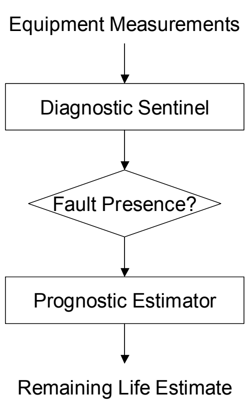

Figure 1 – Interaction of Diagnostics and Prognostics

A condition-based prognostic process is typically described as a two-pronged approach. First, a diagnostic sentinel determines the onset of a fault condition using common fault classification methods. Prior to the fault detection, the prognostic component is largely dormant and remaining life estimates are merely based on equipment statistics (but can be refined with overall wear information). Next, when a fault indication is provided, the prognostic estimator is triggered to supply remaining life estimates that are refined as new information about the condition of the equipment arrives. Figure 1 illustrates this process.

While it is by no means trivial to come up with a good diagnostic system, we will, for the purpose of this paper, assume that a well performing diagnostic function exists. The duality of this approach assumes that there are both diagnostic and prognostic indicators that support both functions. That is, while the information from the diagnostic indicators may be helpful for the prognostics, they may not be sufficient to provide a complete picture. It is desirable to have prognostic evidence from which prognostic indicators can be derived. By prognostic indicators we mean features that exhibit trending characteristics that are ideally proportional with the proximity to the to be predicted event. In real-world applications, prognostic indicators are not easy to come by. Often times, the solution is to use indicators that correlate with accumulated damage and then derive remaining life from this measure, assuming that an upper damage level exists. The derivation of these estimates may involve models that try to capture the impact of changing equipment load conditions on the remaining life. These models may be either based on first principles, using rules, or some other representation. Figure 2 gives an example for diagnostic indicators, and prognostic evidence and indicators.

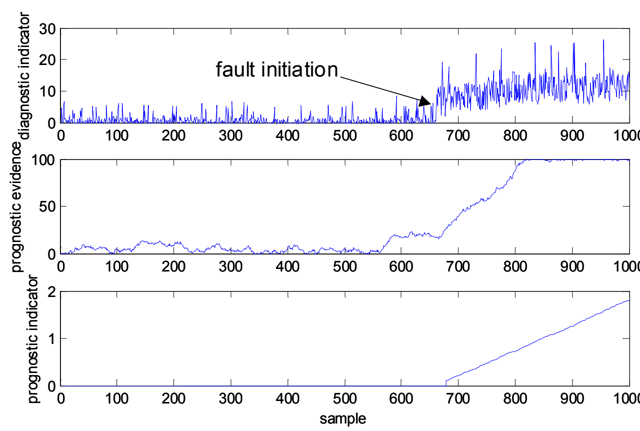

Figure 2 – Input Data: (a) Diagnostic Indicators; (b) Prognostic Evidence; (c) Prognostic Indicators

3 Prognostic Fusion Approach

Fusion can be employed at all levels, data, feature, and decision level. This paper will describe a feature level fusion approach. Approaches suitable for prognostic fusion at the feature level must be able to aggregate different information sources and to give back a continuous output, ideally amended by a confidence value.

The envisioned prognostic process uses 3 major modules: 1.) A preprocessing module that uses appropriate principal component analysis (PCA), filtering, smoothing, normalization, and transformation techniques; 2.) A prognostic model that fuses different information sources and produces a remaining life prediction; and 3.) A post-processing module that recursively confirms the prediction. In run-time mode, the process uses the tuned PCA and ANFIS models that were established during training, using a performance evaluation as a guideline. In the following, we will describe some of the salient features of this process. The process is illustrated in Fig. 3. The individual modules are further described below.

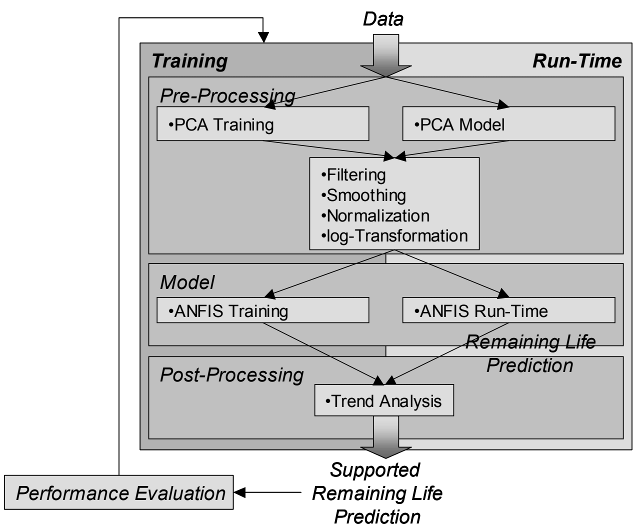

Figure 3 – Prognostic Process

3.1 Preprocessing

Typically, the complexity of a model increases in a nonlinear way with the number of inputs used by a model. High complexity models tend to be excellent in training mode but rather brittle in testing mode. Usually, these models tend to overfit the training data and do not generalize well to new situations. Moreover, some models are very hard to train because the number of model parameters may increase exponentially. On top of that, using input variables that provide little discrimination power may lead to a poor model. It is therefore important to manage the number of input variables carefully. Techniques that help to keep the number of variables at a desired level (which may vary from case to case) include feature selection and feature space reduction techniques. One of the latter is the principal component approach (Singh and Harrison, 1985) where the use of a (small) number of top principal components of a principal component transformation often times allows to explain a very high degree of data variability. The principal component approach will be used for the proposed prognostic model after it was determined that feature selection would not lead to desired performance because the information content was small in each of the features from the input feature set.

To remove noise, reduce data size by compression, and smooth the resulting time series to identify and highlight their general patterns (velocity, acceleration, etc.) further value transformations can be performed using median or rectangular filters. To reduce the response variable's variability we found it advantageous to take the natural logarithm transformation. This affords the learning algorithms with greater sensitivity the closer the variables are towards the target relative to the original response variable. We assumed that the underlying function mapping of input states obtained from sensor readings to desired output can be learned in a static way and not dynamic (involving time changes of these values).

3.2 Model

Soft Computing provides a paradigm in terms of representation and methodologies, which facilitates the integration of uncertain, imprecise knowledge in data-driven methods or in making use of somewhat unreliable data in a knowledge-driven approach. Soft Computing provides a rich repertoire to represent a model structure, to tune model parameters, and to iterate this process. In traditional neural networks, for example, the topology represents the model structure and the links' weights represent the model parameters. In a Mamdani- type fuzzy knowledge base (Mamdani and Assilian, 1975) the structure of the underlying model is the ruleset, while the model parameters are the scaling factors and termsets. The inference obtained from such a system is the result of interpolating among the outputs of all relevant rules. The inference's outcome is a membership function defined on the output space, which is then defuzzified to produce a crisp output. In a Takagi-Sugeno-Kang (TSK) type of fuzzy system (Takagi and Sugeno, 1985) representational power is increased by allowing the use of a first-order polynomial, defined on the state space, to be the output of each rule in the ruleset. This enhanced representational power (which comes at the expense of local legibility (Babuska et al., 1994) results in a model that is equivalent to radial basis functions (Bersini et al., 1995). The same model can be translated into a structured network, such as the adaptive neural fuzzy inference systems (ANFIS) proposed by Jang (Jang, 1993). ANFIS is a representative hybrid system in which NNs are used to tune a fuzzy logic rule base. In ANFIS the ruleset determines the topology of the net (model structure), while dedicated nodes in the corresponding layers of the net (model parameters) define the termsets and the polynomial coefficients.

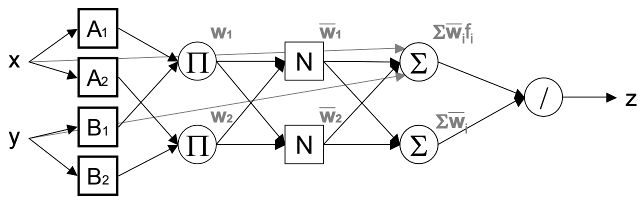

Figure 4 – ANFIS Architecture

ANFIS consists of a five-layer generalized network. As conceptually illustrated in Figure 4 for a system with input x and y, the first layer defines the fuzzy partitions (A_1, A_2, B_1, and B_2) on the input space, the second layer performs a differentiable T-norm operation Π (such as the product or the soft minimum) which gives us the firing strength of the rules, w_i. The third layer normalizes the evaluation of the left-hand-side of each rule, $\overline{w}_i$, so that their degrees of applicability will add up to one. The fourth layer computes the polynomial coefficients in the right-hand-side of each Takagi-Sugeno rule where the function $f_i = p_i x + q_i y + r_i$ is learned with parameters p_i, q_i, and r_i. Lastly, the fifth layer computes the overall output z. Jang's approach is based on a two-stroke optimization process. During the forward stroke the termsets of the second layer are kept equal to their previous iteration value while the coefficients of the fifth layer are computed using a least mean square method. At this point ANFIS produces an output that is compared with the one from the training set to produce an error. ANFIS tries to minimize the mean squared error between the network outputs and the desired answers as the data points in the training set are presented. The RMSE is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2} \qquad (1)$$

where Y and $\hat{Y}$ are the actual and predicted responses, respectively, and n is the total number of predictions. The error gradient information is then used in the backward stroke to modify the fuzzy partitions of the second layer. This process is continued until convergence is reached.

To test the appropriateness of the approach, we simulated data similar to the data shown in Figure 2 where the output was a remaining life estimate covering a range of different prediction horizons and different slopes. The model was trained and then tested on an independent data set. Figure 5 shows the result for 20 training sets (Figure 5 a) and 10 test sets (Figure 5 b) where the actual remaining is superimposed with the remaining life estimates.

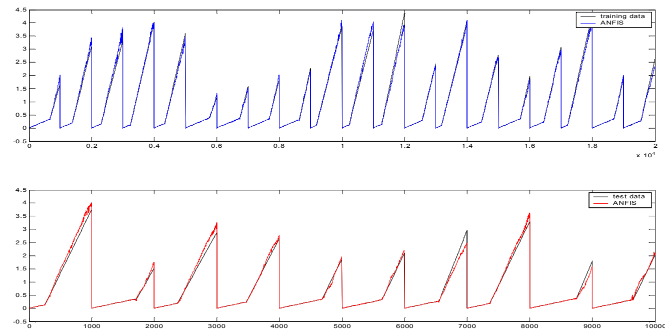

Figure 5 – ANFIS Results for Training and Testing

3.2.3 Post-Processing

The motivation for trend analysis is to take advantage of the correlation between consecutive remaining life estimates. Ideally, all updated predictions are monotonically decreasing (recall that we assume constant load conditions).. The slope of the line that connects all these remaining life points should be *slope* = -1 (assuming that X-axis and Y-axis are time and time-to-failure, respectively). The same argument can be applied to the predicted values of time-to-failure. That is, the slope of an imaginary line that connects predicted time-to-failure should be close to -1 (if we have a perfect predictor). This line is denoted as the prediction line.

In reality, predictions are almost never perfect due to noise, faulty sensors, etc. Hence we would never get a prediction line with *slope*=-1. Nevertheless, the slope of the prediction line would get closer to the target if outliers –predictive data points that are far away from the prediction line – are recursively removed and if the slope of the prediction line is recursively re-estimated.

Even more importantly, predictions will be inconsistent when the open-loop assumption is violated. An abrupt change in the slope indicates a strongly inconsistent prediction. These inconsistencies can be caused, among other things, by a control action applied to correct a perceived problem. We are interested in predicting time-to-failure in open-loop (if no control action is taken). However, often times data are collected with the process in closed-loop (controlled by the operators). Therefore we need to be able to detect when the application of control actions have changed the trajectory's trend. In such cases we suspend the current prediction and reset the prediction history. This step eliminates many false positives.

As an example, let a moving window be of size ten. Slope and the intercept of the prediction line can be estimated by least mean squares. After that, a set number of outliers (say, three) to the line are removed. Next, slope and intercept of the prediction line with the remaining seven data were re-estimated. Then the window is advanced in time and the above slope and intercept estimation process is repeated. The results are time-series of slopes and intercepts. The two consecutive slopes were compared to see how far they were away from *slope*=-1. If they were within a pre-specified tolerance band, e.g. 0.1, we took averages of the two intercepts. In this way, predictions were continuously adjusted according to the slope and intercept estimation. Figure 6 shows the conceptual prediction result.

3.3 Performance Analysis

To guide development of a prognostic approach (and that of prognostic fusion), methods to assess prognostic performance are needed.

Whereas in diagnostics the aim is to classify a fault or precursor of a fault, a prognostics problem tries to make a judgment about the remaining life of a component. This has repercussions for the performance criteria used to measure the goodness of a tool and confusion matrices have not typically been used for prognostic evaluation. We will argue in the following that prognostic confusion matrices can be established. While it would seem a reasonable assumption to assess the performance by whether the estimate was on target or not, there will rarely be an estimate that is completely on the mark. However, this is in most cases not required anyhow. The question then is what is the acceptable tolerance for the problem at hand? We need to keep in mind that the utility of the error is oftentimes not symmetric with respect to zero (where the error is defined as the difference between actual remaining life and estimated remaining life). For instance, if the prediction is too early, the resulting early alarm forces more lead-time than needed to verify the potential for failure, monitor the various process variables, and perform a corrective action. On the other hand, if the failure is predicted too late, it means that this error reduces the time available to assess the situation and take a corrective action. The situation deteriorates completely when the failure occurs before a prediction is made that advises of critical system state. Therefore, given the same error size, it is in most situations preferable to have a positive bias (early prediction), rather than a negative one (late prediction). Of course, one needs to define a limit on how early a prediction can be and still be useful.

Therefore, two different boundaries for the maximum acceptable late prediction and the maximum acceptable early one can be established. Any prediction outside of the boundaries will be considered either a false positive or a false negative.

We define the prediction error (Bonissone and Goebel, 2002) as

$E(t)$=[*Actual time to failure*(t) – *Predicted time to failure*(t)]
and we will report prediction results in terms of a histogram of the prediction error $E(t)$.

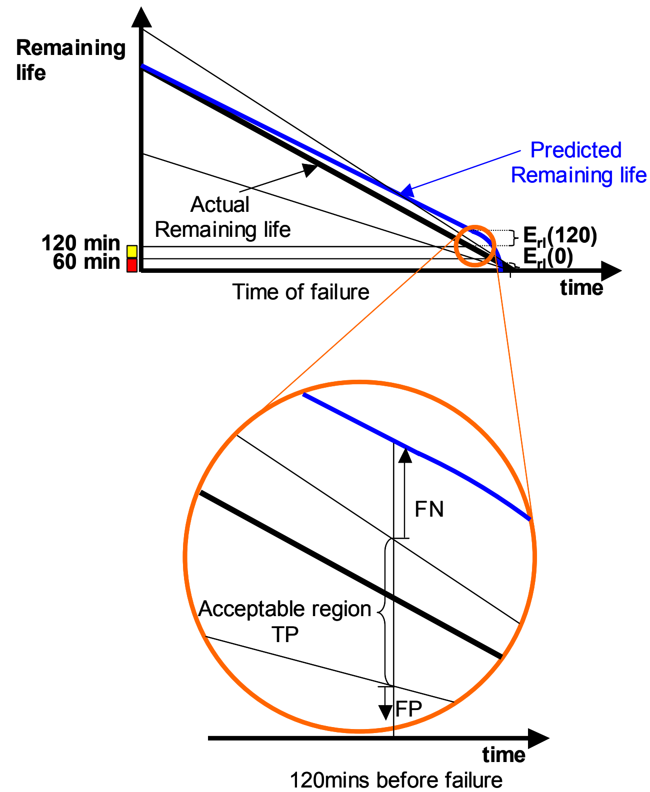

Figure 6 - Conceptual prediction results and error assessment

In particular, focus will be on two instances of $E(t)$:

- $E(t_r)$ - prediction error at the time when the critical zone (for example, within the next mission) is reached, and
- $E(t_0)$ - prediction error at the time when the failure occurs.

Incorrect classifications are typically classified as false negatives (FN) and false positive (FP). In the context of late or early predictions, these categorizations are based on the magnitude of deviation from true time of failure. Therefore, we will define the following limits as the maximum allowed deviations from the origin:

False Negatives A prediction is considered a false negative if one fails to correctly predict a failure more than t_{fn} time units later than the actual time to failure, i.e., $E(t_r) < -t_{fn}$ time units. Note that a prediction that is late more than t_r time units is equivalent to not making any prediction and having the failure occurring.

False Positives A prediction is considered a false positive if we fail to correctly predict a failure if the prediction is more than t_{fp} time units earlier than the actual time to failure, i.e., $E(t_r) > t_{fp}$ time units. We consider this to be excessive lead time, which may lead to unnecessary corrections.

4 Prognostic Fusion Applied to Breakage Prediction

We applied the methods described above to the breakage of the paper web in a paper-making machine at the wet end, specifically at or near the site of the center roll (Bonissone and Goebel, 2002). With great simplification, paper making can be described as turning wood fiber and bonding agents into paper through a series of pressing and drying operations. In the early stages, the product is referred to as the paper web. When the paper web breaks, the machine has to be shut down, cleaned and started up again. Web breaks can result in the loss of 5%-12% of production, with rather big impact on revenue. The paper mill considered had an average of 35 wet-end breaks every month on a machine, with a peak value of as much as 15 in a single day. The average production time lost as a result of these breaks is 1.6 hours/day. Considering that each paper machine works continuously (24 hours a day, every day of the year), this downtime translates to 1.6/24 = 6.66% of its annual production. With an installed base of hundreds of paper machines, producing world-wide revenues of about $45 billions, this translates to loss revenue of $3 billions every year.

While dry-end breaks are relatively well understood, the causes for wet-end breaks are harder to explain and are harder to predict and control. In part this has to do with the time it takes to process the paper material starting from the pulp until it ends up as paper on the final roll versus the warning limits. The latter are considerably longer than the paper processing time. That means that a prognostics system can only deal with system changes that have long transients such as material build-up on drums, etc. It means also that the prognostics has relatively little opportunity to react to material variability because there is by design not enough time to warn against breaks related to these conditions. The aim of this project, then, was to design a web break predictor that will output margin of breaks, i.e., how much time left to a web break. This will help engineers to better anticipate the breaks and take remedial action. Specific requirements were to issue the warning at least 60 minutes before the breakage and potentially up to 90 minutes prior to the breakage. In addition, high priority was placed on avoiding false positive warnings.

As mentioned before, there were dozens of process variables for which time series information was recorded. The variables consisted of flow variables, temperatures, pressures, etc. as shown in Table 1. Both process variables as well as control variables were considered. None of these variables provided a strong indication (or even a consistent weak indication, for that matter) of impending failure. After exhaustive evaluation of the information content of any single variable, it was then attempted to fuse the different information sources.

Table 1: Input Variables Used

Variable Name	Variable Type
Flatbox 3-7 Vaccum	Process
FBHI Vacuum	Process
Couch Vacuum	Process
Kraft Flow	Control
GWD Flow	Control
BGWD Flow	Control
C Broke Flow	Control
Broke Flow	Control
CTMP Flow	Control
Retention Aid	Control
Ash Consistency	Process
R Sash Scan Avg	Process
First Press Uhle	Process
Second Press Uhle	Process
Total Head	Control
Horizontal Slice Pos	Control
Headbox Slice Lb	Control
Rush Drag	Control
Hbox Dilution	Control
Tray Consistency	Process
Grade Number	Control
RSBW Scan Average	Process
RSBW Spread CD	Process
First Main 3rd Press	Control
First Press Pickup Roll	Control
Second Press 1st Press	Control
Third Press 2nd Press	Control
Pickup Roll Wire	Control
First Press Load	Control
Second Press Load	Control
Third Press Load	Control
Couch Load	Control
Wire Load	Control
Large Particle Concentration	Process
Couch speed	Control
Third Press speed	Control
Stock Flow	Control
Total Virgin Stock Flow	Control

First, training data were scrubbed and only appropriate time series were included into the training set (break trajectories with causes that were out of scope for prediction were eliminated), and removal of trajectories with bad data (such as missing or zero values recorded for long period of time). Trajectories of the last 180 minutes before breakage were assembled. Principal components from the data set were computed as the only features, assessing both the top 3 and

4 principal components thus reducing the number of data by an order of magnitude. Next, a median filter with window size of 3 was employed to smooth the data.

The diagnostic mode (as described in section 2) was realized by a classification tree that operated on some features extracted from the principal components such as first and second derivatives. With the fault detection in place, the prognostic model was designed next.

We compared results for different ANFIS configurations. In particular, we tested 3 and 4 membership functions in conjunction with 4 and 3 principal components, respectively. Each input has generalized bell-shaped membership functions (MF). For the three principal component case, there were 292 modifiable parameters for the specific ANFIS structure.

The training of ANFIS was stopped after 25 epochs and the corresponding training and testing root mean squared error (RMSE) were 0.0658 and 0.0975, respectively. For the four principal component case, there were 441 modifiable parameters and the corresponding training and testing root mean squared error (RMSE) were 0.0213 and 0.0424, respectively. Table 1 summarizes ANFIS training for the two training and testing conditions (with the 4 principal component case in parentheses).

Table 1: Summary of ANFIS training

Condition	Setting/Result
# of trajectories	25
# of training data	997
# of testing data	503
# of inputs	3 (4)
# of MFs	4 (3)
Type of MF	Generalized bell-shaped
# of modifiable parameters	292 (441)
# of epochs	25
Training RMSE	0.0658 (0.0213)
Testing RMSE	0.0975 (0.0424)

The model was tested on the data withheld during the shuffling step. It was furthermore validated against independent data sets, both break positive data (BPD) and break negative data (BND).

Shown in figure 7 are a.) the histograms for the error before the 60 minute alert, b.) final error E(0), c.) earliest valid prediction, and d.) maximum absolute error for the test data set. The most important histograms are the histograms 7.a.) and 7.b.) showing the distribution of E(60) and E(0), i.e., the distribution of the prediction error at the time of the 60 minute alert and at the time of the break. The model tends to slightly underestimate the time-to-breakage. This is a desired feature because it provides a more conservative estimate which does not lead to an incorrect sense of time available for the operator. The mean of the distribution of the final error E(0) is around 20 minutes, (i.e., we tend to predict the break 20 minutes earlier) From the histogram of the earliest final prediction, one can see that reliable predictions can be made, on average, about 150 minutes before the break occurs.

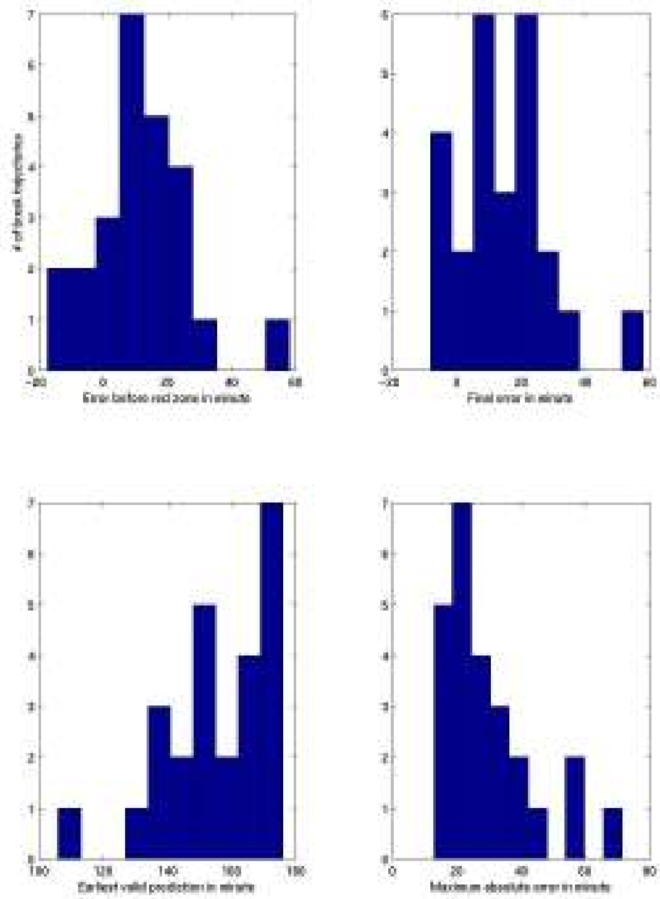

Figure 7 – Performance distributions for test set

Figure 8 summarizes the salient performance metrics FP, Correct Predictions, and FN for the validation. For the train set, similar behavior of the error between time to break = 60 and time to break = 0 can be observed.

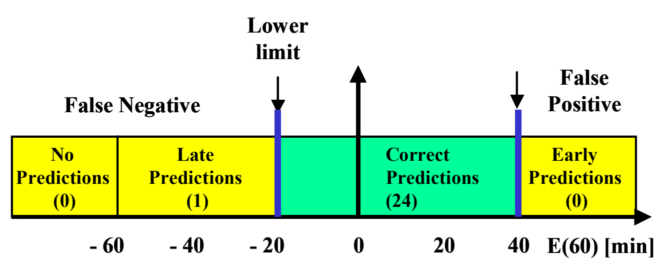

Figure 8 – FP and FN evaluation at 60 minutes prior to break

The variance at the time of the break (t=0) is the same as at the time of the alarm (t=60 minutes). Out of a total of 25 break trajectories, we made 25 predictions, of which 24 were correct (according to the lower and upper limits established for the prediction error at time = 60, e.g. E(60). This corresponds to coverage of 100% of all trajectories.

The relative accuracy, defined as the ratio or correct predictions over the total amount of prediction made was 96%. The global accuracy, defined as the ratio or correct predictions over the total amount of trajectories, was also 96%.

Summary and Conclusions

We described a process for fusing diagnostic and prognostic information to arrive at a remaining life estimate for constant load systems. The process is based on sensor readings coupled with data analysis, principal component analysis (PCA), adaptive network based fuzzy inference system (ANFIS), and trending analysis. Specifically, the process is summarized by the following components: principal components analysis, filtering, smoothing, normalization, transformation, ANFIS modeling, trending analysis and performance evaluation. We then applied the process to estimate time-to-breakage for web breaks in the wet-end part of paper machines used in paper mills. This process generates a very accurate model that minimizes false alarms (FP) while still providing an adequate coverage of the different type of breaks caused by unknown causes.

Pertaining specifically to the paper machine example (but extending to all systems in general) we feel that increasing the information content of the 2nd and 3rd principal component could be helpful in improving the results further. This means that less correlated data need to be acquired which can be accomplished through other process variables including chemical data and external variables, such as "time since felt change", etc. In addition, this particular study used a rather small amount of break trajectories for training which to develop a model that can provide better generalization. Current feedback of break cause for the trajectories is sparse and not always correct. We do not want to pollute the model by trying to learn scheduled downtimes, mechanical failures, etc. Therefore, a best practice would be proper annotation that would allow the elimination of improper trajectories from the training set to properly segment trajectories for model development and to construct a more informative validation set.

We feel this prognostic process is not only applicable for wet-end time-to-breakage prediction but also applicable to other prognostic problems for constant load systems. Indeed, we conducted successful pilot studies in other domains such as predicting failure for certain failure modes of magnets in magnetic resonance imaging (MRI) equipment.

Where the constant load assumption is not valid, prognostics becomes considerably more difficult. This is true because future load or environmental conditions will in most cases change the prediction horizon. In those circumstances, the model will become considerably more complex. Where possible, it is in then often times more successful to employ a physics-based model that mimics the system under consideration and predicts remaining life using future anticipated load/environmental conditions as input variables.

A FP/FN pair constitutes a point on a receiver operating characteristic (ROC) curve. Designing a complete prognostic ROC curve will be the subject of a future work.

Predictive models must be maintained over time to guarantee that they are tracking the dynamic behavior of the underlying process. Therefore, we suggest to repeat the steps of the model generation process every time that the statistics for coverage and/or accuracy deviate considerably from the ones experienced in this report. It is also suggested to reapply the model generation process every time that a significant number of new system failure trajectories (say, twenty) with similar causes has been observed.

References

R. Babuska, R. Jager, and H.B. Verbruggen, Interpolation Issues in Sugeno-Takagi Reasoning, *Proc. Third IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE'94)*, pp. 859-863, Orlando, FL, 1994.

W. Bauer, S. Kim, M.A. Kouritzin, Continuous and discrete space particle filters for predictions in acoustic positioning, *Proc. SPIE*, vol. 4792, pp. 193-206, 2002.

H. Bersini, G. Bontempi, and C. Decaestecker, Comparing RBF and fuzzy inference systems on theoretical and practical basis, *Proc. of Int. Conf. on Artificial Neural Networks. ICANN '95,* Paris, France, vol.1, pp. 169-74, 1995.

P. Bonissone and K. Goebel, When will it break? A Hybrid Soft Computing Model to Predict Time-to-break Margins in Paper Machines, *Proc. SPIE 47th Annual Meeting, Int. Symp. Optical Science and Technology*, Vol. #4787, pp. 53-64, 2002.

B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, Least Angle Regression, *Ann. Statist.* 32, no. 2, pp. 407–499, 2004.

C. Frelicot, A fuzzy-based prognostic adaptive system, *RAIRO-APII-JESA Journal Europeen des Systemes Automatises*, v 30, n 2-3, p 281-299, 1996.J.S. R. Jang, ANFIS: Adaptive-Network-Based Fuzzy Inference System, *IEEE Trans. Systems, Man, Cybernetics*, 23(5/6), pp. 665-685, 1993.

K.-H. Fuh, C.-F. Wu, Proposed statistical model for surface quality prediction in end-milling of Al alloy, *Int. J. Machine Tools & Manufacture*, vol 35, no 8, p 1187-1200, 1995.

R. Haskell, Market Sector Selection Using Fuzzy Regression Trees, *Proc. International ICSC Congress on Computational Intelligence - Methods and Applications*, RIT, Rochester, NY, June 22-25, 1999

E.H. Mamdani and S. Assilian, An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, *Int. J. Man Machine Studies*, vol. 7, no. 1, pp. 1-13, 1975.

H. Ojelund, P.J. Brown, H. Madsen, P. Thyregod, Prediction based on mean subset, *Technometrics*, vol 44, no. 4, pp. 369-378, 2002.

A. Singh and A. Harrison, Standardized principal components, *Int. J. Remote Sensing*, no. 6, pp.989-1003. 1985.

T. Takagi and M. Sugeno, Fuzzy Identification of Systems and Its Applications to Modeling and Control, *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 15, pp. 116-132, 1985.

S. Petit-Renaud and T. Denoeux. Nonparametric regression analysis of uncertain and imprecise data using belief Functions. *International Journal of Approximate Reasoning*, Vol. 35, No. 1, 1-28, 2004.

A. Sfetsos and M. Magro, Short term load forecasting with a hybrid clustering algorithm and pattern recognition, *Engineering Intelligent Systems*, v 12, n 1, pp. 13-19, 2004.

R. Tibshirani, Regression shrinkage and selection via the lasso, *J. Roy. Statist. Soc.*, vol. B 58, no. 1, pp. 267–288, 1996.

G. Vachtsevanos, Performance Metrics for Fault Prognosis of Complex Systems, *Proc. AUTOTESTCON*, pp. 341-345, 2003.

P. Wang and G. Vachtsevanos, Fault prognostics using dynamic wavelet neural networks, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, v 15, n 4, pp. 349-365, 2001.