Classification and Quantification of Matrix Micro-cracks in Composite Structures

C. Larrosa, K.Lonkar, S.K.Shankar
Dept of Aeronautics and Astronautics, Stanford University, CA

December 10, 2010

Fiber reinforced composites have been widely used in aerospace applications due to their functional properties such as high strength-to-weight ratios. But many challenges are yet to be overcome in order to have more confidence on the performance of composites structures. Particularly, composites have different failures types and modes that make the damage prediction and propagation a very challenging subject. The current work will focus on developing structural health monitoring (SHM) techniques for inspection to reduce maintenance costs and increase aircraft safety by understanding and identifying damage types. The steps in implementing a successful Structural Health Monitoring System are to: 1) Locate damage, 2) Identify damage type, 3) Quantify damage and 4) Estimate remaining life. Stanford's Structures and Composites Lab has developed Structural Health Monitoring techniques based on ultrasonic Lamb wave [1] propagation via the use of piezoelectric sensor networks. Using the sensor network in its active sensing mode, damage in a composite structure can be monitored. But the current state-of-the-art detection methods can only detect changes in sensor signal but do not distinguish among damage types or quantify the damage. Knowing the type of damage and quantifying it would provide the inputs needed for physics based prognostic models (lifetime estimation) which would predict the remaining life of the structure for its safe use.

Experimental studies of damage caused in composite plates under tensile-tensile fatigue loading are being conducted by Stanford SACL in collaboration with NASA Ames. The generation of guided Lamb waves is achieved by a surface mounted piezoelectric sensor network (Acellent's SMART Layers®). The sets of sensors serve as a signal generator and receiver pair. The tensile-tensile cyclic loading is generated using SACLs MTS machine. At the end of every N cycle, sensor data along with X-radiography of the sample are recorded. The X-ray images provide the real damage type and quantity. The sensor signals for all paths (the different paths correspond to different combinations of the 6-signal generating sensor and the 6-signal receiving sensors attached to the composite structure) can be processed using different signal processing techniques and different features/parameters can be extracted in the time and frequency domain. The objectives of the current work are: 1) to find which features are more sensitive to different damage types: delamination or matrix micro-cracks; and 2) to learn the relationship between these features and matrix micro-crack density. We will compare different learning algorithms as well as different combinations of features, to find the features and algorithm that yield the best results.

Experiment

Each sample has two SMART Layer® strips with six piezoelectric sensors attached to a surface as shown in Fig. 1. One SMART Layer® strip acts as actuators while the other as the sensors. There are 36 actuator-sensor path combinations. The sensed data is acquired using the commercial package ScanScentry from Acellent. Before loading the sample baseline data is taken which will be used as the base case for data analysis and feature extraction. The sample is then tested under tension-tension fatigue using a standard MTS machine. Every 50,000 cycles the test is paused, sensor data is acquired and an X-ray image is taken. The test is continued until the sample fails.

Data Set

For the pilot run of the algorithms 6 different features were chosen and computed from the raw experimental data. The feature and the corresponding quantity it represents is shown in Table. 1. PSD here is the power spectra density, a quantity which is the measure of the energy content of the time-signal in frequency domain. The data were

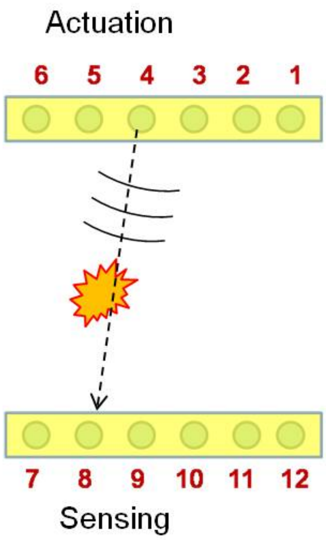

Figure 1: Illustration of the wave propagation across the composite structure showing the 36 different paths $(1, 2, 3, 4, 5, 6) \rightarrow (7, 8, 9, 10, 11, 12)$ across which the signal can be measured.

non-dimensionalized with the corresponding values of the baseline signal. Different learning algorithms were employed to learn on the training data, and their performance was measured on the test data.

Classification algorithms

Five different algorithm in supervised and unsupervised category were used and their performance measured on the data set. They are described as follows

Supervised learning algorithms

Logistic Regression

First model we implemented was logistic regression, as described in the lecture notes [2] with learning rate $\alpha = 1$. Hold-out cross validation was used to test the performance of the algorithm, in which the data set S was randomly split into S_{train} (75% of the data) and S_{CV} (the remaining 25%).

Naive Bayes

The Naive Bayes algorithm was implemented using the discretized set of features (the feature is discretized into 190 bins, the choice based on Fig. 2). The error obtained on the test set was measured by means of a (75 % -25% test) hold-out cross validation method.

GDA

In order to test generative learning algorithms, we implemented GDA, first assuming same covariance matrix (Σ) for $4x|y = 0$ and $x|y = 1$; and then assuming different covariance matrices ($x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma_0)$ and $x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma_1)$). Expression for the parameters ϕ_y, μ_0 and μ_1 are as given in the lecture notes [3] while expressions for Σ_0 and Σ_1 are:

$$\Sigma_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=0\}\left(x^{(i)}-\mu_0\right)^T\left(x^{(i)}-\mu_0\right)}{\sum_{i=1}^{m} 1\{y^{(i)}=0\}}$$
$$\Sigma_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)}=1\}\left(x^{(i)}-\mu_1\right)^T\left(x^{(i)}-\mu_1\right)}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}}$$

SVM

The SVM classification was implemented using the SVM^{light} source package provided by Thorsten Joachims [4]. It implements a fast optimization algorithm to solve both the classification and regression problem. The software supports various other features and the details can be found on (http://svmlight.joachims.org). We patched the source code writtein in C language with MATLAB using the MATLAB-MEX interface to SVM^{light} provided by Anton Schwaighofer [5]. This provided us the seamless capability to perfom the SVM-classification on our data set from MATLAB user interface. The classification was performed using a linear kernel, polynomial and sigmoid tanh kernels. The test error measured by $75\% - 25\%$ hold-out validation using the linear kernel was found to be the least.

Unsupervised learning algorithms

Mixture of Gaussians

To test the performance of unsupervised learning algorithm on our data set, we implemented the mixture of Gaussian method to classify our examples.

To test the performance of unsupervised learning algorithm on our data set, we implemented the mixture of Gaussians method to classify our examples. To fit mixture of two Gaussian distributions, a MATLAB function 'gmdistribution' in statistics toolbox was used. 'gmdistribution' gives us the model parameters, ϕ_j, μ_j and Σ_j for j = 1,2. Once we have model parameters, probabilities of latent parameter z for a given feature vector can be computed using the following relation.

$$w_j = \frac{p(x^{(i)}|z^{(i)}=j;\mu,\Sigma)p(z^{(i)}=j;\phi)}{\sum_{l=1}^{2} p(x^{(i)}|z^{(i)}=l;\mu,\Sigma)p(z^{(i)}=l;\phi)}$$

$$w_j = \frac{\mathcal{N}(\mu_j,\Sigma_j)\phi_j}{\sum_{l=1}^{2} \mathcal{N}(\mu_l,\Sigma_l)\phi_l}$$

The classification error on test data measured while using MG is shown in Fig. 4 for different combinations of the input feature vector.

Feature Selection

The features set consists of continuous real number valued variables. To discretize the feature set the interval of variation of each feature was divided into equal bins of different sizes.

We started out with a train data set containing 6 features. Hold-out cross validation (75-25) was used to test the performance of the models under consideration. The accuracy of the models were tested for all possible combinations of the input feature set. With an input feature size of 6, there are $2^6-1 =$ different combinations of feature sets. This study will help us in 1) identifying the set of features that yield the least test error and 2) it would help in identifying other features which would be more informational for the classification problem at hand. Table 2 gives the test error (i.e. estimate of generalization error) for ten best combinations of feature set for the four supervised learning algorithms.

Using the results from the above feature selection study we conducted a physics based analysis of the problem and introduced three additional features into the input variable. The added set of features is listed in Table. 1 hence increasing the size of the feature vector to nine.

The correlation of the individual features to delamination is measured. Mutual information was scored based on Kulleback-Liebler divergence. The mutual information scores for the nine features for different bin numbers is shown in Fig. 2. Preliminary physics based studies on this data concluded that Time of Flight (x_1) should be highly correlated to delamination. With this background information in hand, a bin size of 190 bins was selected because it produces the highest mutual information score for x_1.

To assess the algorithms, the error that they yield on the test data set is measured while using the whole set of 9-features. The resulting training and test errors are shown in Fig. 3. With the added features, there are $2^9 - 1 = 511$ different combinations of the features. Fig. 4 shows the test error obtained for the 511 different combinations for each of the algorithms being considered. Table 3 gives the test error for the ten best combinations of features.

Precision and recall were computed for all different combinations of new feature set. The scatter plot of precision versus recall is shown in Fig. 6. Choice of best algorithm and feature vector depends on the objective function that we want optimize. Objective 1 in Fig. 6 is to minimize total error (i.e. false alarms (FP) + missed alarms (FN)), which is equivalent to maximizing precision + recall. Objective 2 in Fig. 6 represents minimization of missed alarms, which is equivalent to maximization of recall. In SHM, missed alarms are more important than false alarms.

Table 3 gives the test error for ten best combinations of feature set for all the algorithms considered.

Conclusions and Future Work

The present study focuses on applying machine learning algorithms for classification of delamination in composite samples loaded under in-plane tension-tension cyclic loading. The samples were instrumented with a piezoelectric sensor network that was able to monitor damage. Features were extracted using time domain and frequency domain analysis. The study had two stages. We first implemented the algorithms with a 6 feature set. The test and training errors were very close, indicating that there could be high bias in the models. This suggested using a higher dimensional feature set. Three new features were added and the algorithms were tested for the 511 different possible combinations of features. The addition of new features improved the performance of the algorithms from approximately 26 % test error to 20 % test error. Gaussian Discriminant Analysis with different covariance matrices yielded the least error of 19.5 % and maximum recall. GDA assumes that the data is normally distributed. Since it had the best performance it can be concluded that the data is normally distributed with respect to damage types. The best feature combinations consisted of both time domain and frequency domain features which indicates that for a successful classification both time domain and frequency domain analysis must be considered. In this study the experimental data for three different layups was merged into one set of data, therefore ignoring lamb wave propagation sensitivity to laminate configuration. In the future when more data is acquired for each separate layup, the classification algorithms could be implemented to each layup separately. This will reduce variability in the data compared to this study, and lower classification errors (better performance) would be expected. This work has proved that classification algorithms can distinguish between delamination and matrix cracks. With this information in hand, we can quantify matrix cracks per path. Fig. 5 presents the results of implementing Weighted Linear Regression on one feature for the matrix crack data (label = 0). Future work will explore different regression analysis to learn a relationship between matrix cracks and the raw signal data.

Acknowledgements

We are grateful to Structures And Composites Laboratory (SACL),Stanford and Prognostics Center Of Excellence (PCOE), NASA Ames for providing data for the present studies.

References

[1] Su, Z., Ye, L., Lu, Y. "Guided Lamb waves for identification of damage in composite structures: A review," *Journal of Sound and Vibration*, 295 (2006) 753780.

[2] Andrew Ng, "Lecture Notes 1-Supervised Learning, Discriminative Algorithms," *CS229-Course Material*, pp 16-19.

[3] Andrew Ng, "Lecture Notes 2-Generative Algorithms," *CS229-Course Material*, pp 2-7.

[4] Joachims, T., "Making large-scale SVM learning Practical," *Advances in Kernel Methods-Support Vector Learning, MIT-Press*, 1999

[5] Anton, S., "MATLAB-MEX interface to SVM-light,"*Software version provided by antonsc at microsoft dot com*

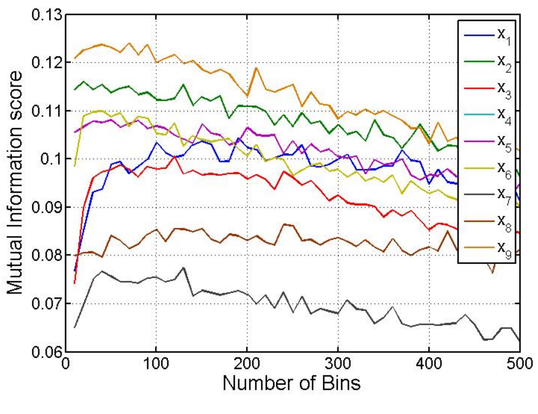

Figure 2: Measure of mutual information.

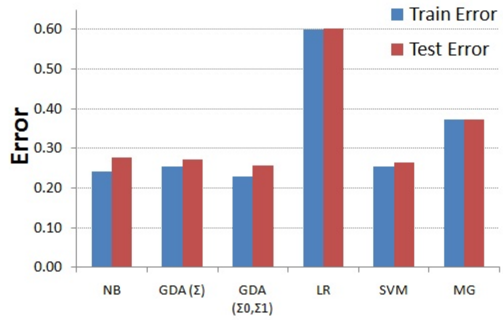

Figure 3: Generalization error measured while using 9-dimensional feature vector for the three different algorithms.

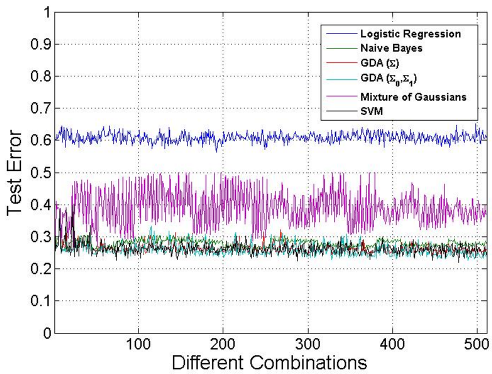

Figure 4: Error obtained while using different combination of input feature set for different algorithms.

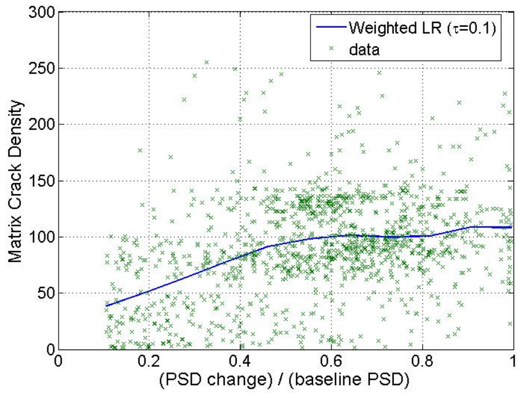

Figure 5: Weighted linear regression model to quantify the matrix crack density.

Feature	LR	GDA (Σ)	GDA (Σ_0, Σ_1)	NB	SVM
(x_1, x_2, x_4)	0.6044	0.2551	0.2408	0.2592	0.2494
(x_1, x_2, x_3, x_4)	0.6253	0.2467	0.2408	0.2628	0.2481
$(x_1, x_2, x_3, x_5, x_6)$	0.6167	0.2466	0.2414	0.2664	0.2584
(x_1, x_2, x_3)	0.6155	0.2535	0.2418	0.2572	0.2674
(x_1, x_2, x_5, x_6)	0.5934	0.2545	0.2422	0.2628	0.2377
(x_1, x_2, x_3, x_6)	0.6032	0.2472	0.2422	0.2597	0.2623
(x_1, x_2, x_6)	0.602	0.2566	0.2426	0.2564	0.2481
$(x_1, x_2, x_3, x_4, x_5)$	0.6229	0.2449	0.243	0.2653	0.2636
$(x_1, x_2, x_4 x_5)$	0.602	0.2554	0.2444	0.2653	0.2481
(x_1, x_2, x_5)	0.6106	0.2563	0.2455	0.2558	0.27

Table 2: Test error for different feature sets

Feature	LR	GDA (Σ)	GDA (Σ_0, Σ_1)	NB	SVM	MG
$(x_1, x_2, x_4, x_7, x_8)$	0.6202	0.2155	0.1953	0.2589	0.2429	0.4436
$(x_1, x_3, x_4, x_6, x_9)$	0.6021	0.2109	0.1984	0.2961	0.2481	0.3917
$(x_2, x_3, x_5, x_6, x_7, x_9)$	0.6253	0.2326	0.2047	0.2915	0.261	0.3692
$(x_2, x_3, x_4, x_7, x_8, x_9)$	0.6163	0.2295	0.2078	0.2961	0.2597	0.384
$(x_1, x_3, x_4, x_6, x_7, x_9)$	0.6176	0.231	0.2078	0.2651	0.2416	0.4676
$(x_1, x_2, x_3, x_8, x_9)$	0.615	0.2388	0.2093	0.2543	0.27	0.3088
$(x_1, x_2, x_4, x_5, x_6, x_8, x_9)$	0.6098	0.2403	0.2093	0.2868	0.2791	0.3402
$(x_2, x_3, x_4, x_5, x_7, x_9)$	0.6021	0.2233	0.2109	0.3008	0.2571	0.3692
$(x_1, x_2, x_3, x_6, x_9)$	0.6279	0.245	0.2124	0.262	0.3062	0.3208
$(x_1, x_2, x_3, x_4, x_8)$	0.6098	0.2295	0.2124	0.2713	0.2752	0.3654

Table 3: Test error for the new feature set

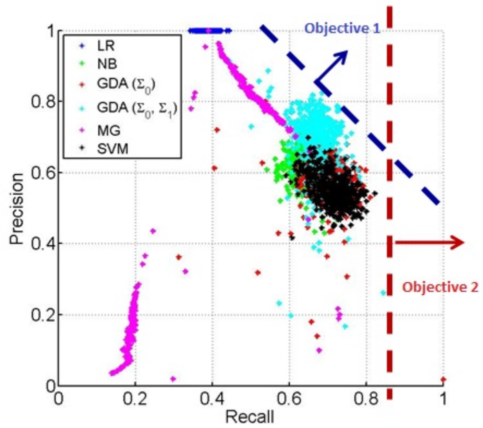

Figure 6: Precision-Recall curve for the different algorithms.

Table 1: Feature set used. The new set of features added are denoted in red.

Feature	Quantity
x_1	Time of flight(TOF) wrt base
x_2	Change in amplitude
x_3	Energy content
x_4	Peak value of PSD
x_5	Change in PSD wrt base
x_6	Rate of change in PSD
x_7	TOF/PSD
x_8	PSD/TOF
x_9	Amplitude/TOF