# Week 6: Prognostics of composite structures utilizing clustering methods and strain data.

Artificial Intelligence for Aerospace Engineering

AE-2224II - Q3 2022/3

Nick Eleftheroglou, Panos Komninos
March 15, 2023

The term 'prognosis' is derived from the Greek word *Πρόγνωσις,* which means to know in advance, to foresee. Between the 7th and 4th century BC, people from all over the known world would visit the temple of Apollo in Delphi Greece, to consult the Oracle for personal matters. The Oracle, in a state of awareness and inspiration, would provide an enigmatic prophecy, still allowing people to take the final decision by themselves. This decision-making process was performed using the information provided by Oracle. Over centuries, the mystic process of prophecy became the science of prediction. Nowadays it is an emerging research field known as prognostics. Prognostics refers to the prediction of the remaining useful life (RUL) of an engineering system based on its current condition and historical data. By predicting the RUL of an engineering system, maintenance and repairs can be scheduled at the most optimal time, rather than waiting for the system to fail. This leads to increased system reliability, as well as cost savings by avoiding unplanned downtime and emergency repairs.

The procedure of damage accumulation in composites, especially during fatigue loading, is a complex phenomenon of stochastic nature that depends on a number of parameters such as type and frequency of loading, stacking sequence, material properties, and so on. To that end, the need for prognostic tools rises and draws increasing attention in the last few years. In this direction, the present case study explores how clustering methods, such as agglomerative hierarchical and k-means clustering, can provide prognostic solutions. The required input data are axial strain data that are obtained from the Digital Image Correlation (DIC) technique.

Figure 1 summarizes the content of the current exercise, which consists of three parts: the training and validation (‘testing’) process. The training process contains the training data, the selected clustering method, and the sojourn cluster time distributions, while the validation process uses the testing data, the estimated clustering parameters and the estimated sojourn time distributions for diagnostics and prognostics purposes.

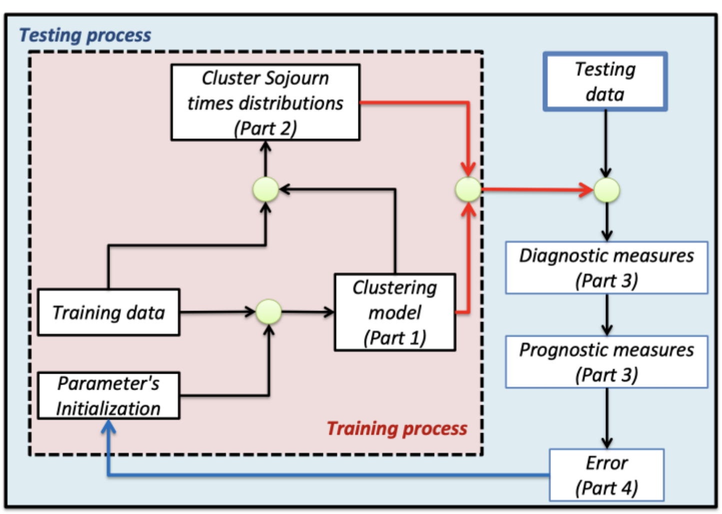



Figure 1. Flowchart of exercise.

You will find the files necessary to complete this exercise in directory week6/clustering of your docker container. From now on, all the paths will be relative to that directory.

## Dataset

Eight composite specimens with $[0/45/90/-45]_{2s}$ lay-up and the following geometrical details; dimensions [300mm x 30mm] and a central hole of 6mm diameter, were tested at 90% of the static tensile strength (S=36 kN) with R=0.1 and f=10 Hz. The tests were executed in an MTS 100 kN universal test machine and they run up to final failure.

A stereo vision system was used to perform 3D full-field DIC measurements. DIC technique enabled strain measurements on the entire surface of the specimen. To obtain the strain distribution in a periodic fashion, the following steps were executed: every 500 cycles (50 sec) the tests were interrupted, and the specimens were loaded quasi-statically up to the maximum load within 1 sec where the load was kept constant for 2 sec so as to take a picture, then the specimens were unloaded within 1 sec and the fatigue tests continued. Figure 2 presents the experimental setup, the fatigue loading scenario, and an illustration of the axial strain distribution of specimen a1. Based on the analytical model of Lekhnitskii, which calculates the effect of a notch on the stress/strain distribution, the green rhomboid
point (half a diameter distance for the hole center in the transverse direction), highlighted in the picture of 0 cycles, was chosen as the critical point to extract the axial strains. Figure 3 presents the eight axial strain degradation histories extracted for the aforementioned critical point.

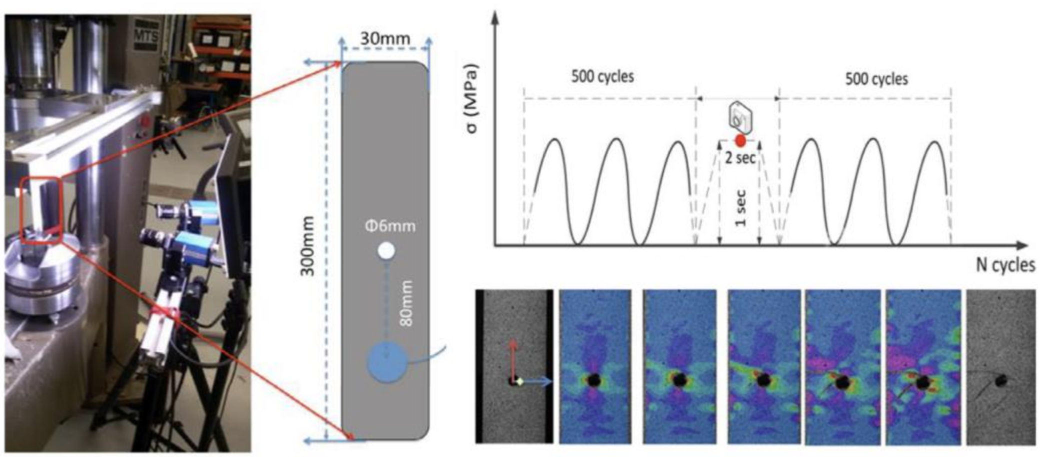
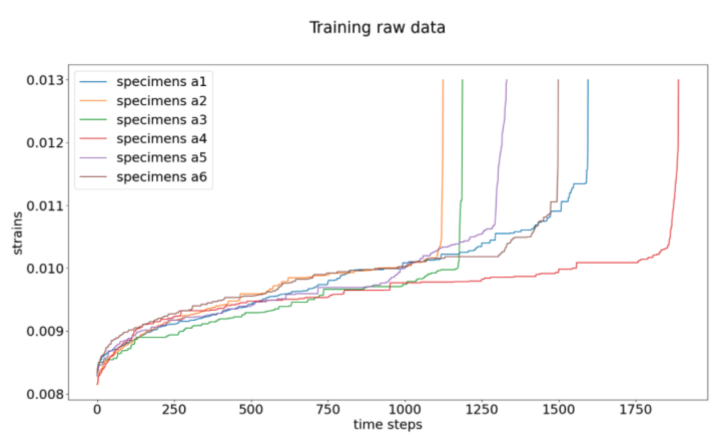



Figure 2. A schematic of the experimental setup, loading procedure, and the acquisition of pictures using DIC.



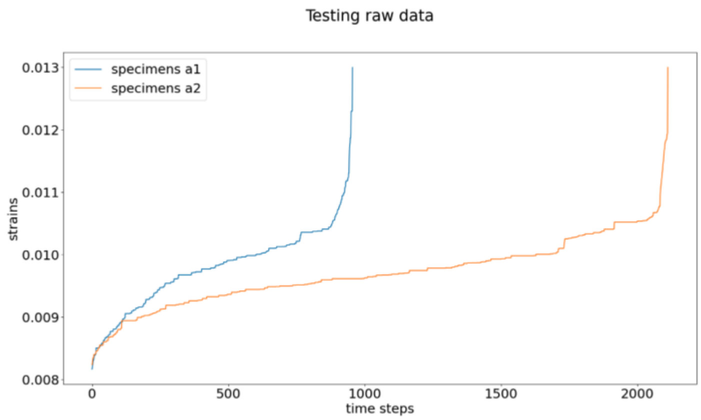



Figure 3. Axial strain degradation histories of eight open-hole specimens.

In conclusion, eight axial strain degradation histories are available. The training dataset employs six degradation histories in order to estimate the clustering model's parameters and keeps two degradation histories as the validation ('testing') prognostic dataset.

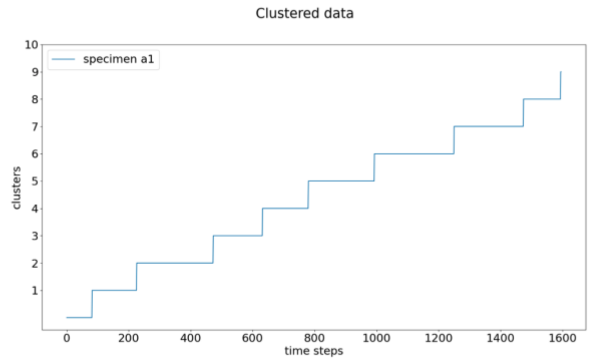
**Part 1: Clustering**

1. Training data: training specimen a1- a6 (train.json), Testing data: testing specimen te_a1 - te_a2 (test.json)
2. Select clustering method: '0' for k-means and '1' for Agglomerative.
    a. For the agglomerative method you need to call the AgglomerativeClustering class from sklearn library and use the arguments: number of clusters, linkage methods. The linkage methods that should be tried are 'ward', 'complete' and 'average'. Although the number of clusters can be directly calculated by the agglomerative technique (can be excluded from hyperparameter), for the case of comparison with the k-means it is still required as input. Also, for simplicity reasons the training data set of the agglomerative method will include not only the training data set (specimens a1-a6) but also the two testing specimens a1 and a2. The optimum number of clusters and the optimum linkage method will be defined based on the mean RUL MSE (Part 3) (code needed in train function)
    b. For the k-means method the code is provided to you. However, you need to identify the optimum number of clusters 'n_clusters' based on two methods; the mean RUL MSE (Part 3) and the Inertia Elbow method (code needed in train function).
3. Output: 'cluster_model' object containing the following:
    a. Agglomerative: methods: fit, predict_labels
    b. k-means: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

   For the following parts, we need the label of the strain data (in which cluster each strain value belongs) after clustering and this can be found in the object's attribute 'cluster_model.labels_' for k-means and in the object's method 'cluster_model.predict_labels' for Agglomerative.

**Part 2: Sojourn time distributions**

1. For each training specimen, plot its degradation history in terms of label (cluster) sequence (code needed in visualize_cluster_results function). An example of the training specimen a1 with 10 selected clusters follows:




2. Write a script for identifying the sojourn time at each cluster for each training specimen. The labels have been already constructed in predict function and used as input for identifying the corresponding sojourn times.
3. The output of the step 2 will be used for fitting a probability density function (pdf) able to describe the sojourn time at each cluster. The input for the fitting process should be a sojourn time matrix with dimensions: [specimen, n_clusters]. The pdf function that will characterize the sojourn time of all clusters is the Weibull pdf. (This function has already been created for you!)

**Part 3: Prognostics**

Time to put everything together!

The clustering results should be used for estimating the RUL (function 'rul_estimation') and visualize its mean versus the actual one (function 'visualize_rul'). For estimating and visualizing the RUL, NO code is needed. However, the labels should be given as input according to your clustering process through the dictionary 'test_labels_dict'. The output of the 'rul_estimation' function is a dictionary of RUL estimations for each testing specimen with the variable name 'test_pred_rul_dict' versus the actual ones that were loaded at the beginning of the code (dictionary 'test_dict'). Those two dictionaries will be used as inputs for the error calculation (Part 4).

***Optional task****: Write a script able to predict not only the mean RUL but also the related 95% confidence intervals (CIs) of a testing specimen and plot the estimated mean RUL values versus the actual RUL (y-axis: mean RUL, 95% CIs, actual RUL/ x-axis: timestep).*

**Part 4: Error Calculation**

Write a script able to calculate the Mean Square Error (MSE) between the estimated mean and the actual RUL values (code needed in mse_loss function).

$$MSE = \sqrt{\frac{\sum_{i=1}^{D}\left(RUL_{actual}(t) - R\hat{U}L_{mean}(t)\right)^2}{D}}$$, where t ϵ [1, D], t is the timestep and D the end-of-life timestep.

# <u>Questions</u>

1. What is the optimal number of k-means clusters utilizing the Elbow method?
    a. K=3
    b. K=4
    c. K=6
    d. K=10

2. Identify the timestep in seconds (time step x 50 sec) that specimen a2 transits from cluster 2 to 3 (first cluster is cluster 1) given that the selected clustering method is k-means and the number of clusters is 5?
    a. 31230 sec
    b. 7550 sec
    c. 12420 sec
    d. 23150 sec

3. Identify the optimal number of clusters and the best linkage method in terms of average testing MSE (average the mean RUL MSE of the two testing specimens).
    a. K=11, Ward linkage method
    b. K=6, Average linkage method
    c. K=3, Complete linkage method
    d. K=8, Average linkage method

4. What is the optimal number of k-means clusters in terms of average testing MSE (average the mean RUL MSE of the two testing specimens)?
    a. K=4
    b. K=6
    c. K=14
    d. K=11