Article citation info:
SHI J, LI Y, WANG G, LI X. Health index synthetization and remaining useful life estimation for turbofan engines based on run-to-failure datasets. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2016; 18 (4): 621–631, http://dx.doi.org/10.17531/ein.2016.4.18.

Jianming SHI
Yongxiang LI
Gong WANG
Xuzhi LI

HEALTH INDEX SYNTHETIZATION AND REMAINING USEFUL LIFE ESTIMATION FOR TURBOFAN ENGINES BASED ON RUN-TO-FAILURE DATASETS

SYNTEZA WSKAŹNIKÓW STANU TECHNICZNEGO ORAZ OCENA POZOSTAŁEGO OKRESU UŻYTKOWANIA SILNIKÓW TURBOWENTYLATOROWYCH Z WYKORZYSTANIEM ZBIORÓW DANYCH O PRACY DO CZASU USZKODZENIA

Turbofan engines will gradually degrade until failure occurs or life ends if without maintenance. Reliable degradation assessment and remaining useful life (RUL) estimation make sense on both aviation safety and rational maintenance decisions. This paper proposes a data-driven prognostic method on the premise of run-to-failure (RtF) data which are multivariate sensory data collected from the engines operating from normal to failure. After necessary pre-processing to the data, clustering analysis is executed to generate the clusters which represent the multi-states of the degradation process. The failure state cluster is extracted, and then the distance between the pre-processed data and the cluster is calculated. Therefore, one-dimensional time series are generated and defined as the health indices. Afterwards the degradation models are built based on the health indices. Finally, the RUL of a testing unit can be estimated by similarity analysis with the models. Hierarchical clustering (HC) and relevance vector machine (RVM) are the main algorithms employed in this paper. To validate the proposition, a case study is performed on turbofan engines data from Prognostics Center of Excellence (PCoE) at NASA Ames Research Center, and sufficient comparisons were given.

***Keywords**: Hierarchical clustering; Relevance vector machine; Run-to-failure; Remaining useful life; Health indices; Prognostics.*

Silniki turbowentylatorowe niepoddane konserwacji ulegają stopniowej degradacji aż do czasu wystąpienia uszkodzenia lub zakończenia cyklu życia. Rzetelna ocena degradacji oraz pozostałego okresu użytkowania (RUL) mają wpływ zarówno na bezpieczeństwo maszyn lotniczych jak i racjonalne podejmowanie decyzji dotyczących utrzymania ruchu. W artykule zaproponowano sterowaną danymi metodę prognostyczną opartą na danych o pracy do czasu uszkodzenia (run-to failure, RTF), które są wielowymiarowymi danymi sensorycznymi zbieranymi podczas normalnej pracy silnika aż do jego uszkodzenia. Po niezbędnej wstępnej obróbce danych, przeprowadzono analizę skupień w celu wygenerowania skupień reprezentujących multi-stany procesu degradacji. Wyodrębniono klaster stanów uszkodzenia, a następnie obliczono odległość między wstępnie przetworzonymi danymi a wyodrębnionym klastrem. Następnie wygenerowano jednowymiarowe szeregi czasowe, które zdefiniowano jako wskaźniki stanu technicznego. Na podstawie tych wskaźników zbudowano modele degradacji. Wreszcie, w oparciu o analizę podobieństwa do opracowanych modeli oceniono RUL jednostki testowej. Główne algorytmy zastosowane w niniejszym opracowaniu to algorytmy grupowania hierarchicznego (HC) oraz maszyny wektorów istotnych (RVM). Aby zweryfikować zaproponowaną w pracy metodę, przeprowadzono studium przypadku z wykorzystaniem danych dot. silników turbowentylatorowych pochodzące z Prognostic Center of Excellence (PCoE) przy NASA Ames Research Center oraz przedstawiono odpowiednie porównania.

***Słowa kluczowe**: grupowanie hierarchiczne; maszyna wektorów istotnych; praca do czasu uszkodzenia; pozostały okres użytkowania; wskaźniki stanu technicznego; prognostyka.*

1. Introduction

The goal of this paper is to estimate the health states of turbofan engines and predict the RUL of them. Turbofan engine health related parameters represent engine component efficiencies and flow capacities [4, 24]. The engine health conditions deteriorate over time until end of life which can be subjectively determined as a function of operational thresholds that can be measured. These thresholds depend on user specifications to determine safe operational limits [24]. The RUL estimates are in units of time (cycles for engines).Reliably estimating RUL can promisingly save engine operational costs and improve safety level. Wear or deterioration of the five rotating components (fan, low pressure compressor (LPC), high pressure compressor (HPC), High pressure turbo (HPT) and low pressure turbo (LPT)) can be monitored by various sensors. The sensory data collected during flight is utilized to estimate the health trending of the engine and its components. In practice very few faults are allowed to go all the way to a failure especially for aero engines. The turbofan engines degradation simulations can be carried out using the commercial modular aero-propulsion system simulation (CMAPSS) test-bed developed by NASA for noisy sensor measurements. Fig.1 was a simplified diagram of the simulated engine. It is a good choice to adopt the simulated RtF data to validate the prognostic methods.

In data-driven prognostic context, the first step is to draw the health index from the multivariate sensory data. Health indices can be divided into two types: 1) physics health index (PHI); 2) synthesis health index (SHI) [12]. The sensory data which directly reflect the damage process or health degradation is served as PHI. Nevertheless, in most situations system or subsystem health is related with various parameters measured by a mount of sensors. Therefore, we need an effective approach to generate health indices from multivariate data. Wang T. et al. [30] employed a linear regression model to transform multi-dimensional sensory signals to one-dimensional SHI. Logistic regression was employed to transfer the multivariate data to health indices by reference [32]. The shortcomings of the above models are that they rely on the whole degradation space and sometimes will be over-fitting. Most of the time it is hard to acquire a data set that is representative of the whole degradation space [21]. Alternatively health state can be decided based on the quantization error away from the normal/failure feature space. Huang R.Q et al. [15] developed a new bearing degradation indicator from three time features and three frequency features based on self-organizing mapping (SOM) method and minimum quantization error away from normal feature space. Inspired by existing research, this paper proposes an approach for turbofan engine health indices generation based on clustering of multivariate sensory data and distance-based similarity analysis with a target cluster.

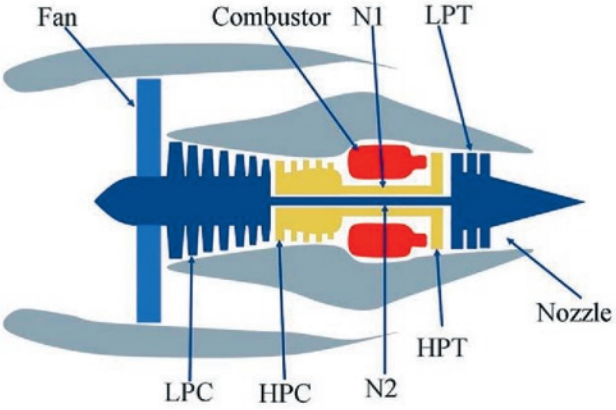

Fig. 1. Simplified diagram of engine simulated in C-MAPSS

From the first time the engines come into use to the final failure, besides the good and failure states, there as well exist intermediate states. As the system operational modes and failure modes are very complicated. It is a practical problem to define the number of the states during the clustering stage. For instance, reference [16] employed fuzzy clustering method to discriminate four health states for the specified application, e.g. good, mild wear, critical and failure. The ground truth or prior knowledge along with the input data should be taken into account to confirm the state number.

Clustering analysis is a class of discovery process that divides data into subsets. Each subset represents a cluster, where the intra-cluster similarity is maximized and the inter-cluster similarity is minimized [12].Clustering is conventionally an unsupervised machine learning algorithm. Xiao W.C [31] et al. proposed a novel clustering ensemble models including semi-supervised method and discussed its application in fault diagnosis of high speed train (HST) running gear. Many literatures applied clustering methods for fault diagnosis, but few for health indices generation.

Cluster stamped by the failure state is served as the baseline. Thus the topological distances between the feature vectors of the sensory data with the baseline are calculated. The distances form a one-dimensional time series defined as the SHI in this paper. In prognostics context, it is generally desirable to have early RUL estimates rather than late RULs, since the main aspect is to avoid failures. For engines degradation scenario, an early RUL estimate is preferred over late RUL. Therefore, we consider it is more appropriate to take the failure cluster, rather than the failure point, as the baseline. Although it may sacrifice the prediction accuracy, it is safe in realistic applications.

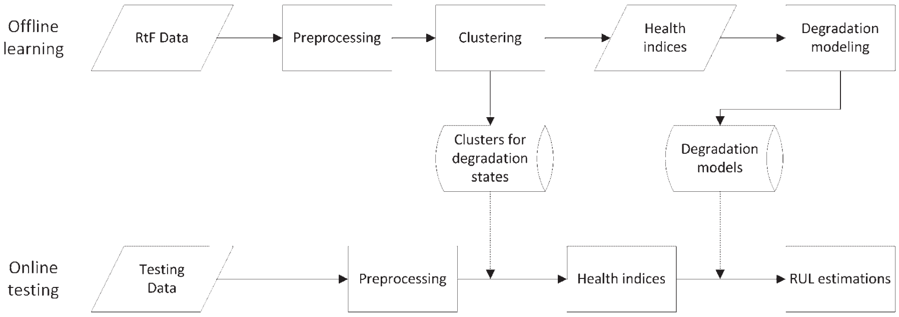

Fig. 2. The framework of the proposed method

With the health indices available, the engine health degradation is then modeled so as to predict the RUL. Machine learning approaches are commonly used such as linear regression [30], neural network [10, 15, 33], stochastic process regression [18], Bayesian learning methods [22] and etc. Goebel K et al. [10] compared three data-driven algorithms for RUL prediction, namely RVM [26], Gaussian process regression (GPR) [14], and neural network. Results of reference [10] show that the RUL estimation errors of RVM are the minimum. In [33], a novel method is developed using unscented Kalman filter (UKF) with relevance vector regression (RVR) and applied to RUL and short-term capacity prediction of batteries. Once the RtF models are constructed, the RUL of a testing unit can be predicted by analysing the similarity between the health indices of the testing unit with the models. After the optimal matching model and the most similar segment of the model are found, the RUL can be confirmed.

Prognostics are meaningless unless the uncertainties in the predictions are account for. Uncertainties arise from various sources, including modeling uncertainties, sensory data uncertainties, future profile uncertainties and etc. [23]. Since the software CMAPSS can ideally simulate the RtF processes for many engines under the same operational settings with different initial states. An ensemble of degradation models can be obtained. Therefore, for a testing engine, we will get the samples of RULs based on the degradation models, then the samples can be used to output the RUL interval estimations.

Fig. 2 illustrates the framework of the proposed method. This paper is organized as follows. Section 2 focuses on synthesizing the health indices from multivariate feature vectors based on the hierarchical clustering method. Section 3 elaborates RUL estimation procedure based on SHI. Section 4 demonstrated the proposition by the NASA CMAPSS datasets and compares the results with different methods and different feature vector data. Finally, section 5 concludes this paper and indicates future work.

2. Health indices synthetization

2.1. Preprocessing of the RtF data

Before clustering, three preprocessing measures are taken.

(1) Parameter selection

Firstly, parameters of interest were selected to construct the feature vectors related with the health degradation. One vector consists of the parameters at a time instant, which can be treated as a point in the multidimensional space. It is crucial to choose the appropriate parameters. The criterion is that the selected parameter must be coupling tightly with the health degradation or the concerned failure modes. Typically, the health parameters are correction factors on the efficiency and flow capacity of the components (fan, compressors, turbines, and nozzle) of the engines, while the measurements are, for instance, inter-component pressures, temperatures, and shaft speeds [3].

(2) Outlier removal

In data mining community, there always exist outliers in the raw that must be removed using some algorithms [2, 19]. An outlier is a data object that deviates significantly from the rest of the objects. This paper used K nearest neighbor (KNN) algorithm to remove the outliers, as it is the most-frequently used algorithm and computational economic. The flow of the KNN algorithm is as follows.

a) Set the value of K, and establish a rule for confirming an outlier.
a) Calculate the first point's distances to the other points to obtain the K nearest points. The average of the K distances will be obtained.
b) Repeat step b until all vectors are traversed.
c) Rank the average distance values in descending order. Select the top points as outliers, or select the points whose average distance values exceed the set line. This depends on what type of rule is applied.

(3) Normalization

Normalization is executed in the last step of data preprocessing. It must be noted that normalization should be taken carefully. It has a precondition for normalization that the parameters contribute equivalently to the health degradation expression. Thereby, normalization has two advantages in health index synthetization. 1) Each parameter is transformed to the value varying in the range of [0, 1], which is helpful for expressing the health index. 2) The negative effect caused by the different scales of parameters can be eliminated.

2.2. Clustering of the preprocessed data

As an unsupervised machine learning method, clustering can be used in health monitoring and state estimation especially when the number of states is unknown. The sensory data of system or subsystem is time series data, for which the clustering methods are summarized by Liao [17]. Reference [12] divides the clustering methods into four categories, i.e. partitioning methods, hierarchical methods, density-based methods, and grid-based methods. The agglomerative hierarchical clustering method based on the Matlab functions is used in this paper. The result of agglomerative hierarchical clustering is a structured tree graph called a dendrogram. The tree is not a single set of clusters, but rather a multilevel hierarchy, where clusters at one level are joined as a cluster at the next level [5, 25, 28]. This allows us to decide the level or scale of clustering that is most appropriate for our application. And HC require no initial settings beforehand. The algorithm flow is as follows:

a) Take each data point as a dependent cluster so that there is only one member in a cluster;
b) Calculate the Euclidean distance between every two points, then each point and its nearest point converge to a cluster. This process is called linkage;
c) Calculate the Euclidean distance between every two clusters, then each cluster and its nearest cluster converge to a new cluster;
d) Repeat step c until all points converge to one cluster;
e) A binary cluster tree is created and is trimmed based on some rules to get the final clusters.

For a given input $m \times n$ matrix X, each row vector of this matrix is a data point at a time slice. In step c, for two different vectors x_i and x_j, their distance can be obtained by the following formula:

$$d_{st}{}^{2} = (x_i - x_j)(x_i - x_j)' \tag{1}$$

Here the distance metrics are discussed shortly. There are several different distance metrics, for instance, the Euclidean distance, Standardized Euclidean distance, Mahalanobis distance, and etc, among which Euclidean distance is a commonly used. Mahalanobis distance takes the scales into account by introducing a covariance matrix in the distance measures. As the data was normalized afore, the scales of dimensions import no effects on the distance measures. Therefore, Euclidean distance is a reasonable similarity measure in this application.

The combination of two generated clusters is called linkage. There are numerous linkage methods, including single linkage, complete linkage, average linkage, centroid linkage, and Ward's linkage. This paper adopts Ward's linkage, which is a type of least mean square (LMS) algorithm that is only adapted to Euclidean distance. The Ward's linkage algorithm is as follows. Suppose that there exist cluster r and cluster s. Their numbers are n_r and n_s, respectively. The centroids are $\bar{x}_r$ and $\bar{x}_s$. The Ward's distance between them is calculated by the equation:

$$d(\mathrm{r,s}) = \sqrt{\frac{2n_r n_s}{n_r + n_s}} \left\| \bar{x}_r - \bar{x}_s \right\|_2 \tag{2}$$

where $\| \ \|_2$ represents the Euclidean distance.

The dendrogram is trimmed according to a certain rule to acquire the final clusters. The common rules are "max cluster number", "inconsistency coefficient", etc. The inconsistency coefficient characterizes each link in a cluster tree by comparing its height with the average height of other links at the same level of the hierarchy. Clusters are formed when a node and all of its subnodes have an inconsistent value less than c. In cluster analysis, inconsistent links can indicate the border of a natural division in a data set. In the RtF process of one failure mode, the states can be divided into four categories, i.e. healthy state, subhealthy state, degraded state, and failure state. For the failure state cluster, it can be further divided into more clusters according to the dendrogram. This can be useful for refining the failure state cluster but should be executed carefully according to different applications.

2.3. Distance-based synthesis health indices

After the preprocessing and clustering of the primitive RtF data, clusters are acquired that represent different states, i.e. normal state, failure state or intermediate degraded states. The cluster centroid and radius are related with the health state and accordingly kept as the so-called knowledge. Health state estimation is actually a similarity analysis process for testing or real-time data with the learned clusters. The Euclidean distance between the testing data (preprocessed as well) point with each cluster centroid is calculated, and the nearest cluster is taken as the target cluster for the point. Generally, the distance should not exceed the radius. However, there might be some data points that fall outside the target cluster. If the learning stage covered the whole feature space, the data points are considered as outliers. Otherwise further analysis should be carried on in case the points form a new cluster or expand an existing cluster.

The failure state cluster is extracted and served as the baseline to generate the health indices. The centroid of the failure state cluster c_F is a $1 \times n$ vector. Qualitatively, a larger distance value represents a better state; otherwise, the health is deteriorative. For the training data, suppose there are S_1 units (engines) that run through the degradation process. The preprocessed training data for the *i*th unit is denoted by Tr that is a $L_i \times n$ matrix. The health indices for the ith unit is denoted by $h_i(l_i)$ with $i = 1,2,\cdots,S_1$, and $l_i = 1,2,\cdots,L_i$. L_i is the length of #i engine life. The Euclidean distances between Tr and c_F are calculated by the following formula:

$$h_i(l_i) = \sqrt{(Tr_{l_i} - c_F)(Tr_{l_i} - c_F)'} \qquad (3)$$

So we can obtain a health index time series for each training unit. And it is the same for the testing units. The health indices are the basis for RUL estimation. The health indices of a training unit always contain an uptrend tail, as seen in Fig.3. In order to output a safer estimated RUL based-on model-matching method, the tail should be cut.

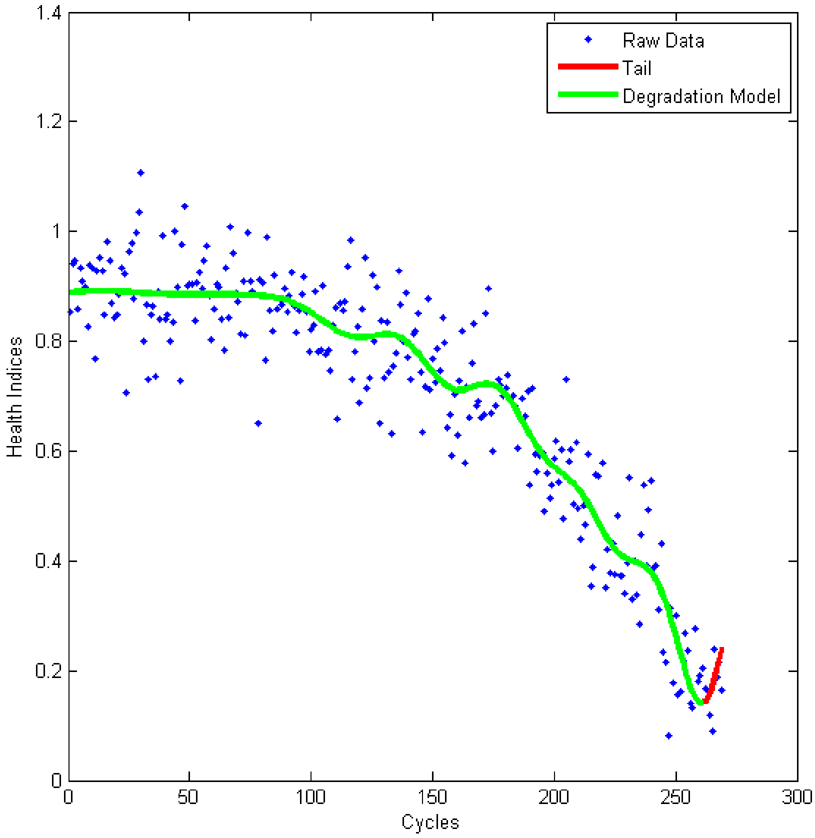

Fig. 3. The tail of a training unit health idices

3. RUL estimation based on health indices

3.1. Algorithms for degradation modeling

The health indices can be directly used to estimate the RUL. Furthermore, degradation models can be firstly constructed based on the health indices. In prognostic application, many degradation modeling algorithms are researched, for example, auto-regression and moving average [9, 13], exponential regression [30], and RVR. This section introduces the theoretical background of RVM.

The relevance vector machine (RVM) algorithm is a sparse Bayesian learning algorithm proposed by Tipping in 2000 [26, 27]. On the basis of SVM, Tipping applied the kernel theory to the Bayesian inference of the Gaussian process. The irrelevant points are removed to acquire the sparse model by the theory of automatic relevance determination (ARD) under the hierarchical prior parameters [7, 20]. Compared with SVM, RVM offers some advantages including non-“Mercer” kernels, sparsity, fewer hyperparameters and probabilistic predictions.

Given a dataset of input–target pairs $\{\mathbf{x}_n, t_n\}_{n=1}^{N}$, the aim is to learn a model for the dependency of the targets on the inputs to make accurate predictions of t for unseen values of $\mathbf{x}$. The targets are samples from the model with additive noise:

$$t_n = y(\mathbf{x}_n, \mathbf{w}) + \varepsilon_n \qquad (4)$$

where y is used to base the prediction, and ε_n are independent samples from some noise process that is assumed to be mean-zero Gaussian with variance σ^2.

Assuming the independence of t_n, the likelihood of the complete dataset can be written as:

$$p(\mathbf{t} \mid \mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\{-\| \mathbf{t} - \mathbf{\Phi w} \|^2 / (2\sigma^2)\} \qquad (5)$$

where $\mathbf{t} = (t_1 \ldots t_N)^T$, $\mathbf{w} = (w_0 \cdots w_N)^T$, and $\mathbf{\Phi}$ is the $N \times (N+1)$ ‘design’ matrix with $\mathbf{\Phi} = [\phi(x_1), \phi(x_2), \ldots, \phi(x_N)]^T$ wherein $\phi(x_n) = [1, K(x_n,x_1), K(x_n,x_2), \ldots, K(x_n,x_N)]^T$, with $K(x_n,x_i)$ being a kernel function.

If directly estimated by the maximum likelihood estimation (MLE), this will lead to over-fitting of the parameters. To solve this problem, Tipping defines a zero-mean Gaussian prior distribution over $\mathbf{w}$:

$$p(\mathbf{w}|\alpha) = \prod_{i=0}^{N} N(w_i | 0,\ \alpha_i^{-1}) \qquad (6)$$

where $\alpha = \{\alpha_0, \alpha_1, \cdots, \alpha_N\}$ is a vector of $N+1$ hyperparameters.

Hyperpriors for α and σ^2 are then defined as:

$$p(\alpha) = \prod_{i=0}^{N} \mathrm{Gamma}(\alpha_i | a, b)$$
$$p(\sigma^2) = \prod_{i=0}^{N} \mathrm{Gamma}(\beta | c, d) \qquad (7)$$

Where $\mathrm{Gamma}(\alpha_i | a,b) = \Gamma(\alpha_i)^{-1} b^a \alpha_i^{a-1} e^{-b\alpha_i}$, and $a = b = c = d = 10^{-4}$. When the hyperparameters approach infinity, the probabilistic distribution of the corresponding weights tends to 0. The related inputs with nonzero weights are deemed to be “relevant” and are the core points characterizing the time series. Next, the hyperparameters should be optimized based on the observed data. The posteriori distribution of weights $\mathbf{w}$ satisfies Gaussian distribution:

$$p(\mathbf{w} | t, \alpha, \sigma^2) = (2\pi)^{-(N+1)/2} |\Sigma|^{-1/2} \exp\left\{ -\frac{(\mathbf{w}-\mu)^T \Sigma^{-1} (\mathbf{w}-\mu)}{2} \right\} \qquad (8)$$

The posteriori variances and means of weights $\mathbf{w}$ are:

$$\Sigma = \left(\sigma^{-2}\mathbf{\Phi}^T\mathbf{\Phi} + \mathbf{A}\right)^{-1} \qquad (9)$$
$$\mu = \sigma^{-2}\Sigma\mathbf{\Phi}^T t$$

where $\mathbf{A} = diag(\alpha_0, \alpha_1, ..., \alpha_N)$; the detailed derivation process can be found in [5]. The optimal values of $\boldsymbol{\alpha}$ and $\boldsymbol{\sigma}^2$ should maximize the following:

$$p(\mathbf{t}|\alpha,\sigma^2) = \int p(\mathbf{t}|\mathbf{w},\sigma^2)p(\mathbf{w}|\alpha)dw$$
$$= (2\pi)^{-N/2}\left|\sigma^2\mathbf{I} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^T\right|^{-1/2} \exp\left\{-\frac{1}{2}\mathbf{t}^T\left(\sigma^2\mathbf{I} + \mathbf{\Phi}\mathbf{A}^{-1}\mathbf{\Phi}^T\right)^{-1}\mathbf{t}\right\} \qquad (10)$$

$\boldsymbol{\alpha}$ is estimated by the iterative method:

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2} \qquad (11)$$

wherein $\gamma_i = 1 - \alpha_i \Sigma_{ii}$, μ_i is the mean value of the ith posteriori weight in eq.(15), and Σ_{ii} is the ith diagonal element of the posteriori variance matrix Σ . The variance $\boldsymbol{\sigma}^2$ is estimated by the same method:

$$(\sigma^2)^{new} = \frac{\|\mathbf{t} - \mathbf{\Phi}\boldsymbol{\mu}\|^2}{N - \Sigma_i \gamma_i} \qquad (12)$$

3.2. RUL estimation through similarity analysis

The commonly used RUL estimation approach is by extrapolating the degradation model of testing units across the specified failure threshold (FT). However, how to define an appropriate FT for a specified testing unit is still a challenge. The degradation behavior of individual engine can differ according to operational environment. It is realistic that the FTs can vary and should be assigned dynamically for different testing unit, rather than static ones based on fixed number of states as presented in [8]. In reference [16] the authors proposed a dynamic FT assignment technique by looking at distance similarity among learned classifiers and indexes of test data. Center of the last cluster of the most matched classifier that learned from a training unit is defined as the FT. In reference [29, 30] and this paper, the most matched training unit is selected by distance based similarity analysis with all the points of the degradation trajectory. Therefore, although computation time is shorter as declared in reference [16], the most matched training unit selection process is more elaborate by the trajectory similarity measures. There is another problem not considered in reference [16]. The data of each training unit is clustered independently, so there are hundreds of clusters for the training data. From the whole view of the training data set, there must be serious overlaps between clusters. So the distance between the data point of the testing unit with A cluster (for example) and B cluster might be very close. Besides, the most matched clusters for a testing unit might locate in different training units. Then how to select the matched classifier in these situations is not illustrated by the authors.

3.2.1. Basic model-matching (BMM) RUL estimation

With enough RtF training data available, RUL prediction is conducted more favorably by similarity analysis, which finds the best matched model and locates the point where the preprocessed testing data fit into the degradation model to acquire the value of the RUL [29]. The similarity between the testing data and the model is measured by the root mean square error (RMSE). The RUL estimation via model-matching method was executed as follows.

The health index data series of the units of the training data sets are modeled (e.g., smoothing, regression) to obtain $TrainData = \{traindata1, \cdots, traindataS_1\}$ (new data series).

a) Match the first testing unit data “testdata1” with “traindata1” based on the least mean square error (LMSE) principle to obtain the optimal position pos_{test1_train1} with minimal MSE mse_{test1_train1} .

b) Repeat step b; i.e., match “$testdata1$” with other training data. The optimal position data $POS_1 = \{pos_{test1\_train1}, \cdots, pos_{test1\_traink}, \cdots, pos_{test1\_trainS_1}\}$ and the corresponding MSE $MSE_1 = \{mse_{test1\_train1}, \cdots, mse_{test1\_traink}, \cdots, mse_{test1\_trainS_1}\}$ are then obtained. Select the minimum MSE from “MSE_1” as the optimal matching unit denoted by “$train_{test1}$” for “$testdata1$”. The corresponding optimal position is denoted by “$pos_{test1} = pos_{test1\_traink}$”.

c) Repeat step b and step c to match each test unit with “TrainData” to obtain the optimal matching units $Train\_match = \{train_{test1}, train_{test2}, ..., train_{testS1}\}$ and positions $POS = \{pos_{test1}, pos_{test2}, ..., pos_{testS1}\}$.

d) Calculate the remaining cycles for the testing units. With regard to the ith unit, its RUL is $\hat{L}_{RUL_i} = length(train_{testi}) - pos_{testi}$.

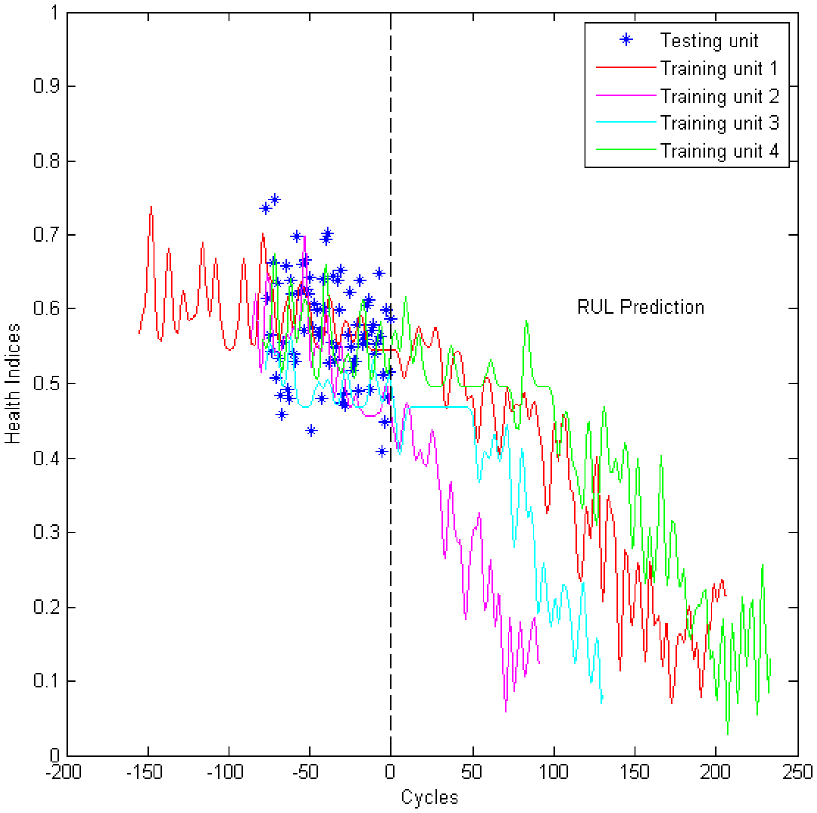

Fig. 4. RUL Prediction error based on model-matching method

3.2.2. Improved model-matching (IMM) RUL estimation

As depicted in Fig.4, the RUL prediction error based on the LMSE principle is too large. There are several training units that match well with a testing unit; however, the training units differ largely from each other. The best matching training unit we picked suits the testing unit only locally; when applied to predict the RUL, the error became too large. The health states of some these testing units usually concentrate on "healthy" and "sub-healthy" states, and the health indices change gently. This implies that RUL prediction at the early stage of degradation might not be inaccurate.

To solve this problem, we proposed an improved model-matching RUL estimation. The test units can be categorized into three types by the health states: the first are the units containing "failure" state; the second are the units whose health states concentrate on "healthy" and "sub-healthy" and whose health indices are placidly evolving; the remaining units are the third type, which contain the "degradation" state. We adapted different model-matching strategies for the three cases.

Case 1: The first type of unit was treated the same way as in sec.3.2.1.

Case 2: The second type of unit was processed as follows. Suppose that the ith testing unit " $testdata_i$ " was of the second type; calculate the MSE values by all the training units according to the method proposed in sec.3.2.1, then sort the values and select p training units with minimum MSE values $\{mse_{i\_1},\cdots,mse_{i\_j},\cdots,mse_{i\_p}\}$.

Therefore, the related RUL $\left\{RUL_{i1},RUL_{i2},\cdots,RUL_{ip}\right\}$ vector was generated. To estimate the ultimate RUL, the p RULs were given the weights $\{w_1,w_2,\cdots,w_p\}$. The weight w_j was obtained by the following formula:

$$w_j = mse_{i\_p-j} \Big/ \sum_{j=1}^{p} mse_{i\_j} \tag{13}$$

The RUL of the ith testing unit was then $\hat{L}_{RUL_i} = \sum_{j=1}^{p} \omega_j \cdot RUL_{ij}$.

Case 3: The third type of unit was treated as follows. As in case 2, select p matching training units for one testing unit of this type. Pick out the last portion of data points from the testing unit. Calculate the MSEs with the p training units, and obtain the minimum MSE within the p values. The RUL of the testing unit is the one calculated by the minimum MSE related training unit.

A good prognostic system not only provides accurate and precise RUL predictions but also specifies the level of confidence associated with such predictions. In addition to the point estimation, rational interval estimation can support the risk decision. The uncertainty of RUL prediction arises from several aspects—e.g., the sensory data noise, the modeling error, the operational condition variance, the initial state difference of the system, and the working load variation. Therefore, RUL prediction is a complex dynamic nonlinear problem [1, 34].

When the RUL is estimated by applying the model matching method and there are multiple degradation models, each model can generate an RUL value intuitively. In this way, an optimal RUL can be given in the sense of mathematics and can also output the probabilistic results.

With the samples of the RUL, there are two types of approaches to obtain the probabilistic results: one is the parametric method, and the other is the nonparametric method. For the first approach, the distribution of RUL should be estimated first, followed by the parameters of the distribution. The parametric method must define the distribution type. When the distribution type is unknown, the nonparametric estimation methods are more applicable. The common methods include rank method and statistical histogram.

4. Case study on simulated turbofan engines

4.1. Data sets and prognostic assessment criterions

The data for demonstration are provided by the prognostic-data-repository of the PCoE of NASA. The datasets were generated by CMAPSS and kept in several text files. A dataset is constructed of 26 dimensions/columns, wherein the first column is the unit number for different engines, the second column is the time index (cycle), the third through fifth columns are the settings of operational conditions, and the other 21 columns are simulated sensory data [24].

The dataset "train_FD001", which contains the simulated RtF data, was selected for training. The dataset "test_FD001", which contains the partial degradation process data, is used for RUL estimation. The simulation experiment contains one failure mode, which is the deterioration of the high-pressure compressor (HPC). "Train_FD001" covers the RtF process of 100 testing engines and consists of 20,631 data points (row vectors). "Test_FD001" contains the data collected in the running process with no failures, which is used to predict when the failure will occur—in other words, to estimate the RUL. The operational conditions of "FD001" are the same but have different initial states, which are caused by the variant initial wear and manufacture bias.

For results evaluation, estimated RULs are compared with actual RULs provided in the file "Rul_FD001.txt." Most importantly, for a given testing unit, an interval I = [−10,13] is considered to assess RUL estimates as on-time, early or late. In PHM context, it is generally desirable to have early RUL estimates rather than late RULs. Another criterion is the accuracy of prognostics model evaluated by coefficient of determination that should be close to 1.

$$R2 = 1 - \frac{\sum_{i=1}^{n}\left(L_{RUL}-\hat{L}_{RUL}\right)^2}{\sum_{i=1}^{n}\left(L_{RUL}-\bar{L}_{RUL}\right)^2} \tag{14}$$

4.2. Offline learning

As the three parameters out of the 21 sensory signals, i.e. Total temperature at HPC outlet (P1), Total pressure at HPC outlet (P2), and Ratio of fuel flow to Ps30 (P3), are related with HPC health degradation. They are used to construct the 3-dimeansional feature space. Fig. 5 shows the curves of P1, P2 and P3 for one engine (#2). It is explicit that the sensory data are contaminated by noise. The charts of normalized P1, P2 and P3 are also depicted in Fig. 5. According to the framework in Fig. 2, there exist outliers within the raw data, which might impact the results. Therefore, outliers should be removed to enhance the accuracy of the outputs. The outliers are detected through the KNN method and removed from the 20,631 data points. The value K is a trial value, and we set different values to test the effects. It is discovered that the distances to the K nearest neighbors of the top 10 points are much larger than those of the others. Then the 10 points are removed with K=4000.

The hierarchical clustering for the 20,621 data vectors was conducted to generate the dendrogram in Fig. 6. The cut line in Fig. 6

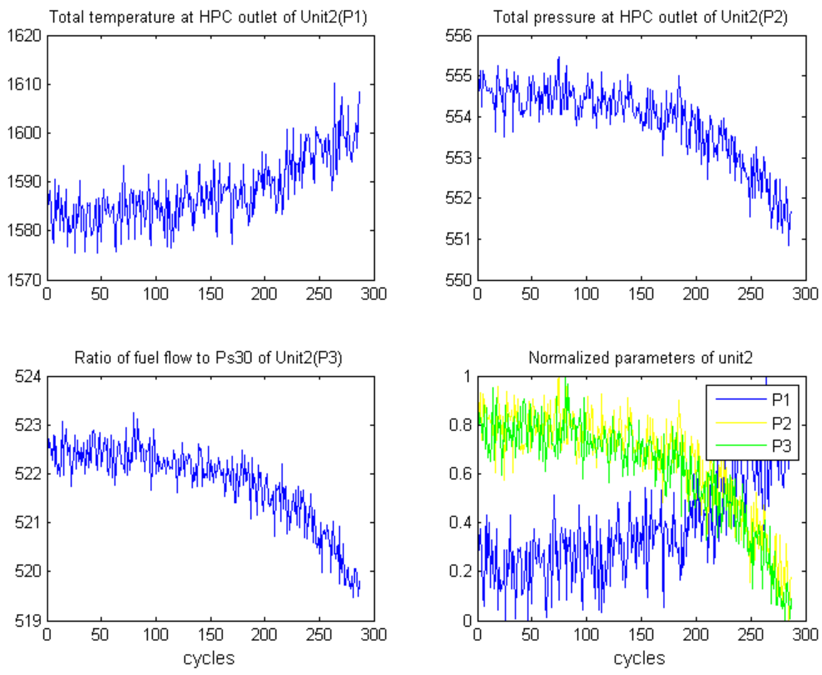

Fig. 5. Curves of 3 parameters for #2 engine

shows that the generated cluster number is 4.The engine health states are divided into four classes: healthy state, subhealthy state, degraded state, and failure state. The color coding of the health states is given in Table 1.

Because "train_FD001" is the RtF dataset of the 100 engines organized in sequence, according to the unit number, the clustering can detect the endpoint of each engine. Taking engines #1 and #2 for example, the health indices and related health states are shown in Fig. 7. The health states are denoted in order by "1", "2", "3" and "4", which represents the RtF process.

According to the dendrogram, the failure cluster can be further divided to refine the failure state information. As shown in Fig.8, there are three patterns of divisions for the failure cluster, i.e. 2, 3, and 4 sub-clusters. The refined failure state information is shown in Table 2. The refined clusters are denoted by RC1 and RC2. We will compare RUL estimations based on different health indices based on cluster "4" and the refined clusters.

4.3. RUL estimation

This section gives the results of RUL estimation of the 100 testing units by different methods and input data.

(1) Different degradation modeling methods

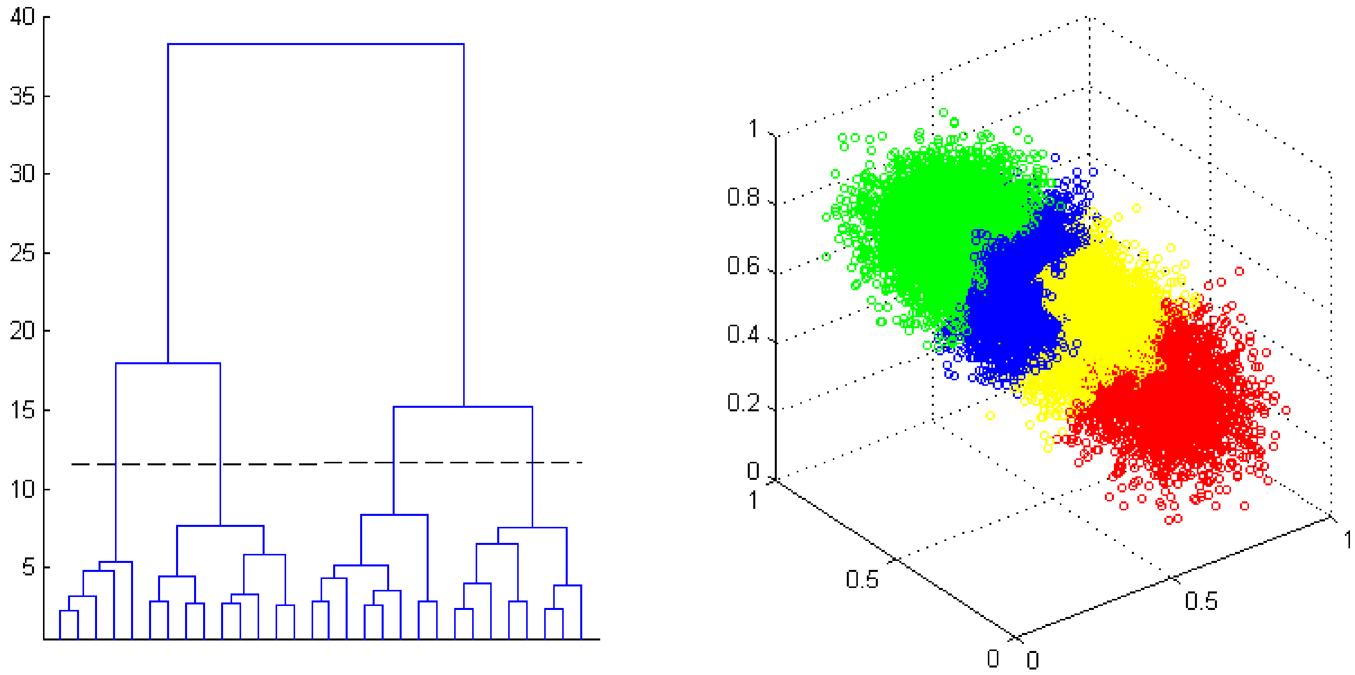

Fig. 6. clustering of the training data

Table 1. Results of the clustering of RtF data

Color		Green	Blue	Yellow	Red
Health state		Healthy	Sub-healthy	Degraded	Failure
Centroid	P1 P2 P3	0.337718 0.704982 0.734335	0.453279 0.596571 0.61862	0.58049 0.500512 0.503711	0.716569 0.319629 0.30542
Radius		0.37004	0.377641	0.369111	0.415286
Point quantity		6087	5964	6055	2515

Table 2. Results of failure state cluster divisions

No.		1	2	3
Sub-cluster Case		2 sub-clusters	3 sub-clusters	4 sub-clusters
Centroid	P1 P2 P3	0.765613 0.322051 0.304777	0.777587 0.259857 0.238734	0.777587 0.259857 0.238734
Point quantity		1774	770	770

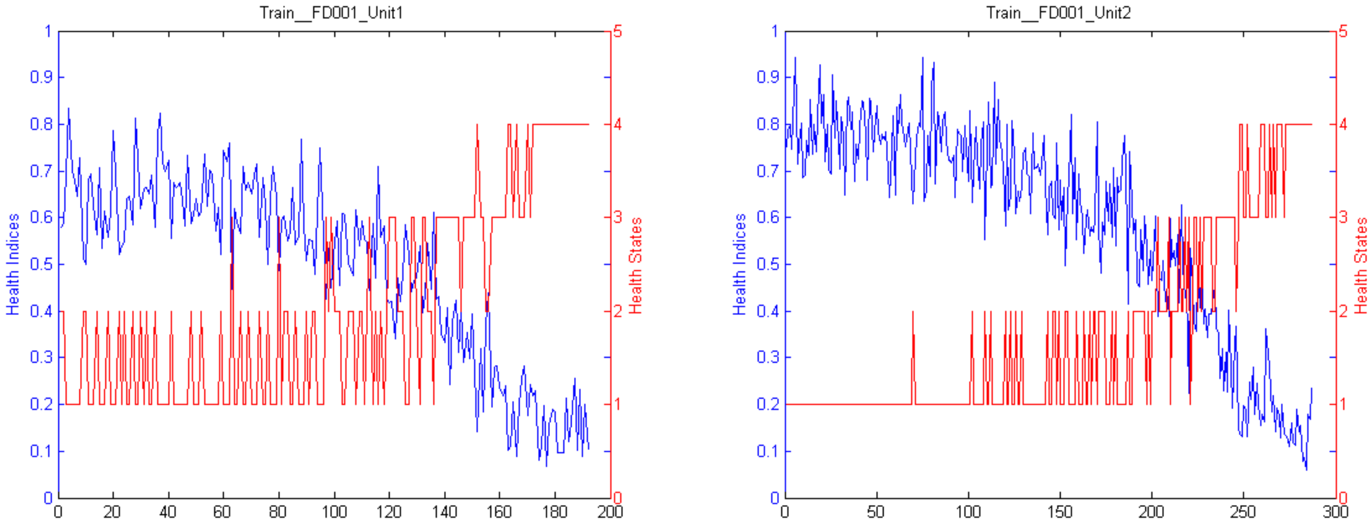

Fig. 7. Health indices and states of #1 and #2 engines

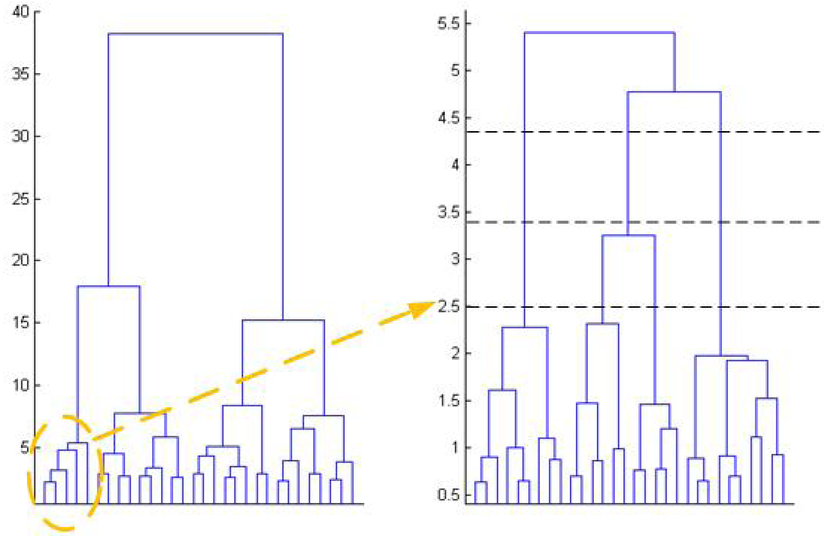

Fig. 8. Failure state cluster further divisions

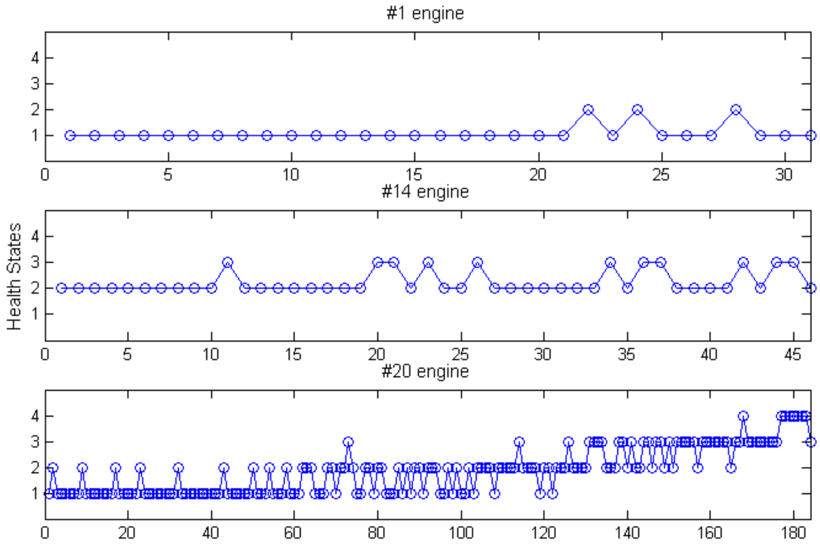

Fig. 9. Health state evolutions of the testing engines

This part is to compare different degradation modeling methods based on health indices derived from cluster “4”. The tails of the health indices are not cut. And BMM method is used. As shown in Table 3 are the results.

(2) Individual parameter versus SHI

This part is to validate the effectiveness of health indices compared with one-dimensional data. The health indices are derived from cluster “4” with tails retained. RVM and BMM method are used here. The results are shown in Table 4.

(3) BMM versus IMM

Before carrying out the IMM for the testing units, we classified the testing units based on health state estimation. Three typical units (#1, #14 and #20 engines) were selected to account for the classification. As shown in Fig. 9, the health states of unit #1 are almost “1”, meaning that it is in the healthy condition. For unit #14, the health states are mostly “2” in the earlier stage but switch to “3” in the later stage, which indicates that the engine health state is changing from the sub-healthy to the degraded state. With regard to the #20 unit, the engine has passed through the 4 states and might be near failure. This part compares BMM and IMM with RVM modeling algorithm based on health indices derived from cluster “4”. The tails of the health indices are not cut. Results are displayed in Table 5.

(4) Tail retained versus tail cut

This part is to compare the results based on health degradation models with tail retained and cut. RVM and IMM are used here. The results are shown in Table 6.

The last column of Table 6 indicates the proposed method in this paper is effective and has better performance than others. When compared with reference [9], the results of on-time RULs and R2 are better.

Each testing unit had 100 probable RUL values, which could be used for uncertainty analysis. Here, the nonparametric method was applied to output the interval estimation results based on the IMM method.

The median of the sorted 100 RULs was obtained first. The confidence interval (CI) was set as 70%, which means that 70 RUL values were selected. The RULs located between the lower confidence limit

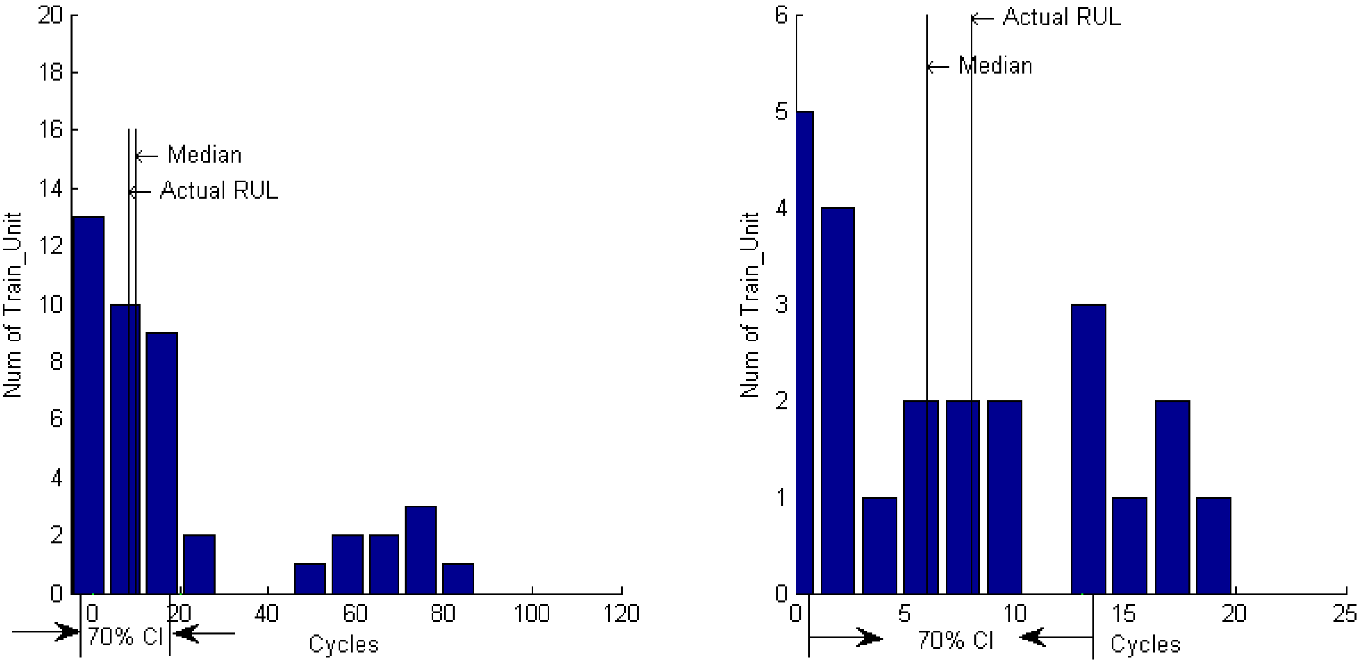

Fig. 10. The uncertainty of #31 testing engine RUL estimation

Table 3. Comparisons with different degradation models

Criteria	Original	Smoothing	ER	RVM
RUL error interval	[-146,87]	[-103,48]	[-183,79]	[-75,114]
On-time RULs	37	24	38	37
Early RULs	42	11	19	40
Late RULs	21	65	43	23
R2	0.2	-0.2	-0.44	0.42

Table 4. Comparisons with different input data

Criteria	P1	P2	P3	SHI
RUL error interval	[-154,70]	[-200,89]	[-147,82]	[-75,114]
On-time RULs	31	34	34	37
Early RULs	36	27	40	40
Late RULs	33	39	26	23
R2	0.08	-0.38	0.25	0.42

Table 5. Comparisons between BMM and IMM

Criteria	BMM	IMM
RUL error interval	[-75,114]	[-56,58]
On-time RULs	37	39
Early RULs	40	22
Late RULs	23	39
R2	0.42	0.70

Table 6. Comparisons between tail cut and retained

Criteria	Tail retained	Tail cut		
		Cluster "4"	RC1	RC2
RUL error interval	[-56,58]	[-74,64]	[-57,63]	[-58,64]
On-time RULs	39	45	49	60
Early RULs	22	33	27	21
Late RULs	39	22	24	19
R2	0.70	0.58	0.67	0.69

(LCL) and the median occupied approximately 65% of the total 70 RULs, whereas the RULs located between the median and the upper confidence limit (UCL) occupied 35%. Furthermore, we cut off the RULs whose corresponding MSE values with the training units were larger than 1.5 times the minimum MSE value. The 70% CI then decreased to facilitate a more accurate prediction.

Let us take test #31, of which actual RUL is 8, as an example. To use the model-matching method, the length of the train unit must be longer than the test, and there are 43 train units that meet this requirement. The median of these 43 RULs is 10, and its 70% CI is [0, 20] (see left part of Fig. 10). If 1.5 times the minimum MSE value is used to cut off the RULs, then the median is 6, and the 70% CI is [0, 13] (see right part of Fig. 10).

4. Conclusion

In this work, we investigated data-driven prognostic methods for turbofan engines. The machine condition monitored data are collected and used for health state estimation and RUL prediction. We proposed a health index synthetization approach by hierarchical clustering and distance-based similarity analysis. With RtF data available, the whole degradation model was constructed by RVR based on health indices of the RtF training unit. Then the RUL of a testing unit was estimated by similarity analysis with the degradation models of training units. In real world failure prognostics are difficult as the degradation of system/subsystem under monitored is complex and nonlinear. The simulated engine degradation datasets by CMAPSS are noisy and nonlinear, which are employed by many researchers. Some significant issues are raised and the adaptive adjustments for the methods are highlighted. In the specified turbofan engine application our work has the following advantages.

a) The health index synthetization is robust even there is no RtF or failure state data. Since the good state cluster can substitute as the baseline.
b) The hierarchical clustering algorithm is flexible and needs no initial settings and little prior knowledge.

c) The proposed methods are applicable in safety-critical systems/ subsystems, as the number of on-time and early RULs are relatively larger.

But still there are a few testing units whose RUL errors are relative too large. These units are in an early degradation state as a matter of fact, which means it is difficult to estimate the RUL of a unit in early stage. Besides, in real world, the RtF datasets can hardly be collected so that we cannot construct an ensemble of whole degradation models for RUL estimation. In the perspective of this paper, further research work should focus on the dynamic RUL estimation methods for real in-service engines without RtF datasets.

Acknowledgement

This research work is supported by the safety, reliability and maintainability program of space utilization. The authors would like to thank the PCoE for publishing the datasets. The authors are very grateful to the editors and reviewers for their valuable comments and suggestions.

References

1. Adriaan Van Horenbeek, LilianePintelon. A dynamic predictive maintenance policy for complex multi-component systems. Reliability Engineering & System Safety 2013; 120: 39-50, http://dx.doi.org/10.1016/j.ress.2013.02.029.
2. Angiulliand F, Pizzuti C. Fast outlier detection in high dimensional spaces. In proceedings of the sixth European conference on the principles of data mining and knowledge discovery 2002; 15-26.
3. Borguet S, Leonard O. Coupling principal component analysis and Kalman filtering algorithms for on-line aircraft engine diagnostics. Control Engineering Practice 2009; 17: 494-502, http://dx.doi.org/10.1016/j.conengprac.2008.09.008.
4. Dan S, Donald L.S. Kalman filter constraint switching for turbofan engine health estimation. European Journal of Control 2006; 12:331-343, http://dx.doi.org/10.3166/ejc.12.341-343.
5. Devillez A, Billaudel P, Lecolier GV. A fuzzy hybrid hierarchical clustering method with a new criterion able to find the optimal partition. Fuzzy Sets Syst. 2002; 128(3):323–338, http://dx.doi.org/10.1016/S0165-0114(01)00187-7.
6. D. Frederick, J. DeCastro, and J. Litt, "User's Guide for the Commercial Modular Aero-Propulsion System Simulation (CMAPSS)," NASA/ARL, Technical Manual TM2007-215026, 2007.
7. D.J.C.MacKay. Bayesian methods for backpropagation networks. Spring New York 1996; 211-254
8. E. Ramasso, M. Rombaut, and N. Zerhouni, Joint prediction of continuous and discrete states in time-series based on belief functions. IEEE Trans. Cybern. 2013; 43 (1): 37–50, http://dx.doi.org/10.1109/TSMCB.2012.2198882.
9. Godolphin EJ. A direct representation for the large sample maximum likelihood estimator of a Gaussian autoregressive moving average process. Biometrika 1984; 71:281–289, http://dx.doi.org/10.2307/2336244.
10. Goebel K, Saha B, Saxena A, 2006, A comparison of three data-driven techniques for prognostics, IEEE aerospace conference, Big Sky, MT, Mar 4-11.
11. Hu C, D.Youn B, Wang P.F, T.Yoon J. Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life. Reliability Engineering and Systems Safety 2012; 103: 120-135, http://dx.doi.org/10.1016/j.ress.2012.03.008.
12. Han J.W, Kamber M, Pei J. Data mining: concepts and techniques. Elsevier. 2012, http://dx.doi.org/10.1007/978-1-4419-1428-6_3752.
13. Hannan EJ, Rissanen J. Recursive estimation of mixed autoregressive-moving average order. Biometrika 1982; 69: 81–94, http://dx.doi.org/10.1093/biomet/69.1.81.
14. He Z.K, Liu G.B, Zhao XJ, Wang MH. Overview of Gaussian process regression. Control and Decision 2013; 28(8):1121-826.
15. Huang R.Q, Xi L.F, Li X.L, Liu C.R, Qiu H, Lee J. Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. Mechanical Systems and Signal Processing 2007; 21: 193-207, http://dx.doi.org/10.1016/j.ymssp.2005.11.008.
16. Kamran J, Rafael G, Noureddine Z. A New Multivariate Approach for Prognostics Based on Extreme Learning Machine and Fuzzy Clustering. IEEE Transactions on Cybernetics 2015; 45(12): 2626-2639, http://dx.doi.org/10.1109/TCYB.2014.2378056.
17. Liao T.W. Clustering of time series data-a survey. Pattern Recognition 2005; 38:1857–1874, http://dx.doi.org/10.1016/j.patcog.2005.01.025.
18. Liu D.T, Pang J.Y, Zhou J.B, Peng Y, Pecht M. Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression. Microelectronics Reliability 2013; 53: 832-839, http://dx.doi.org/10.1016/j.microrel.2013.03.010.
19. Lozano E, Acuna E, 2005, Parallel algorithms for distance-based and density-based outliers, Fifth IEEE International conference on data mining, Miami, FL, Dec 6-9.
20. Neal RM. Bayesian Learning for Neural Networks. Lecture Notes in Statistics 1996; 118, http://dx.doi.org/10.1007/978-1-4612-0745-0.
21. Qiu H, Lee J, Lin J, Yu G, Robust performance degradation assessment methods for enhanced rolling element bearing prognostics, Advanced Engineering Informatics 2003; 17: 127–140, http://dx.doi.org/10.1016/j.aei.2004.08.001.
22. Saha B, Goebel K, Poll S, Christophersen J. Prognostics methods for battery health monitoring using a Bayesian framework. IEEE Transactions on Instrumentation and Measurement 2009; 58(2): 291-6, http://dx.doi.org/10.1109/TIM.2008.2005965.
23. Saxena A, Celaya J, Saha B, Saha S, Goebel K. Metrics for Offline Evaluation of Prognostic Performance. International Journal of Prognostics and health management 2010; 1(1): 2153-2648,.
24. Saxena A, Goebel K, Simon D, Eklund N., 2008, Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation, IEEE International Conference on Prognositcs and Health Management, Denver, CO, Oct 6-9, http://dx.doi.org/10.1109/phm.2008.4711414.
25. Tang X.Q, Zhu P. Hierarchical Clustering Problems and Analysis of Fuzzy Proximity Relation on Granular Space. IEEE Transactions on Fuzzy Systems 2013; 21(5): 814-824, http://dx.doi.org/10.1109/TFUZZ.2012.2230176.
26. Tipping M. Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research 2001; 1: 211-244.
27. Tipping M. The Relevance Vector Machine. Neural Networks & Machine Learning 1999; 1(3): 652-658.
28. Tsekouras G, Sarimreis H, Kavakli E, Bafas G. A hierarchical fuzzy clustering approach to fuzzy modeling. Fuzzy Sets Syst. 2005; 150(2): 246–266, http://dx.doi.org/10.1016/j.fss.2004.04.013.

29 Wang T. Trajectory Similarity Based Prediction for RUL Estimation. PhD thesis, University of Cincinnati 2010.

30. Wang T, Yu J, Siegel D, and Lee J, 2008, A similarity-based prognostics approach for remaining useful life estimation of engineered systems, International Conference on Prognostics and Health Management, Denver, CO, Oct 6-9, http://dx.doi.org/10.1109/phm.2008.4711421.
31. Xiao W.C, Yang Y, Wang H.J, Li T.R, Xing H.L. Semi-supervised hierarchical clustering ensemble and its application. Neurocomputing 2016; 173: 1362-1376, http://dx.doi.org/10.1016/j.neucom.2015.09.009.
32. Yan J, Koc M, Lee J. A prognostic algorithm for machine performance assessment and its application. Production Planning & Control 2004; 15(8): 796-801, http://dx.doi.org/10.1080/09537280412331309208.
33. Zheng X.J, Fang H.J. An integrated unscented kalman filter and relevance vector regression approach for lithium-ion battery remaining useful life and short-term capacity prediction. Reliability Engineering and System Safety 2015; 144: 74-82, http://dx.doi.org/10.1016/j.ress.2015.07.013.
34. Zio E, Di Maio F. A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear power plant. Reliability Engineering and System Safety 2010; 95(1): 49-57, http://dx.doi.org/10.1016/j.ress.2009.08.001.

SHI Jianming
LI Yongxiang
WANG Gong
LI Xuzhi
Technology and Engineering Center for Space Utilization
Chinese Academy of Sciences
No.9 South Dengzhuang Rd, Haidian Dist
Beijing, 10094, China
E-mail: sjm@csu.ac.cn, liyongxiang@csu.ac.cn,
wanggong@csu.ac.cn, xzhli@csu.ac.cn