An Operating Condition Classified Prognostics Approach for Remaining Useful Life Estimation

Qi Li
School of Automation Science and Electrical Engineering
Beihang University
Beijing, China
liqi961104@gmail.com

Zhan Bao Gao
School of Automation Science and Electrical Engineering
Beihang University
Beijing, China
gaozhanbao@126.com

Li Qun Shao
School of Automation Science and Electrical Engineering
Beihang University
Beijing, China

Abstract—**This paper presents a prognostics approach based on operating condition for estimating the Remaining Useful Life (RUL). Operating condition is used to describe the state or environment of a system. This approach is suit for the dataset that contains sensor measurements and operational settings. Predicting RUL contains two stages: modeling stage using the training dataset and predicting stage using the result of modeling and testing dataset. This approach can increase available information in modeling stage and simulate the actual work situation of the test unit in the predicting stage. The performance of this approach was tested by the dataset from 2008 PHM Data Challenge Competition where sensor measurements and operational settings were provided. The task of the competition was to estimate the RUL of an unspecified system. The results showed that this prognostic method could get accurate predictions in most situations and had a good rank in all competition results.**

Keywords- Remaining useful life, Operating condition, Prognostics, perfomance degradation, Data driven.

I. INTRODUCTION

Prognostic for Remaining Useful Life (RUL) is a crucial technique in the research field of Prognostics and Health Management (PHM). The task of PHM is to maintain equipments effectiveness that is works continuously and efficiently. In order to complete this mission better and reduce maintenance costs, the maintenance strategy has gradually changed from Corrective Maintenance (CM) to Condition-Based Maintenance (CBM). Large amounts of monitoring data make it possible to predict the state of equipment in real-time. RUL prognostic plays an important role in CBM. Although many researchers have conducted a lot of studies in this area and propose many methods. Most methods focus on the optimization of algorithms instead of data features and objects analysis.

As most systems are too complex to develop mechanism model, data driven method becomes a core approach in prediction of RUL. However, it is not enough to get a mathematic model using fundamental algorithm. Working environment, system's feathers and other factors have great influence of system's RUL. So, the data should be explored in more detail to gain more comprehensive information which can improve predictive performance. Admittedly, if all the relevant factors involved in the calculation of RUL are known, RUL computation would be deterministic, accurate and precise[1].

In this paper, we present a new data driven method for RUL prognostics, which is based on the operating condition of the complex system. We used a dataset that the PHM'08 Data Challenge provided. The dataset is consisted of 3 operational settings and 21 sensor measurements for 218 units. All of them are time series. This dataset is further divided into training and testing subset. There are some effective methods that have been proposed by researchers to predict RUL using the dataset. TianyiWang et al. [3] proposed a similarity-base matching method which most effectively completes this challenge. Felix O. Heimes et al. [4] put forward a recurrent neural networks approach. Leto Peel et al. [5] combined MLP and RBF neural network with a kalman filter to predict RUL. However, all methods made good use of 21 sensor measurements but didn't fully exploit the operational settings. In Tianyi Wang's method, it have emphasized the significance of operational settings and used them to analyze sensor trends. However, they didn't use them in the matching progress. Leto Peel's approach didn't distinguish the operational settings but just used them as all of the sensor measurements. They only suggested that selecting a reduce set of sensors would improve the performance of the approach in discussion part. Unfortunately, there was rare research on this dataset after 2008.

Operating Condition (OC) is used to describe the working state or working environment. When components work on different OCs, their lifetimes show a significant difference. When a component works on varying OCs, the degeneration curve actually is multi-track instead of monotonic. So, it is necessary to develop models according to OCs, which would be more likely real working situation. In this way, we can get multiple models for different OCs, and develop a degeneration pattern library depending on OCs for predicting RUL. Although a lot of researchers point out that working environments and working loads have great influence on the RUL in their papers. As far as we know, there are few algorithms consistent with the assuming.

In this paper, we proposed a new approach to take OCs into consideration in our modeling strategy for prognostic RUL and developed the whole algorithm. Firstly, we analyzed the experiment dataset and extracted OCs using 3 operational

*Supported by Beijing Natural Science Foundation (Study on real time fault prognostics technique in equipment integrated health management system)
*Supported by the Fundamental Research Funds for the Central Universities(PHM signal simulation key technology research)

9781479949434/14/$31.00 ©2014 IEEE

settings. Then, degradation model was developed by training dataset based on OCs. The third step was matching with testing dataset according to OCs. Finally, The RULs of all testing units were calculated by matching results.

The analysis of experimental dataset and OCs extraction are described in Section II. The detailed methodology employed to develop the OCs based degradation model is introduced in Section III. The implement algorithm procedures are elaborated in Section IV. The testing results of our algorithm are analyzed in Section V. Finally conclusion as well as future work on the approach is discussed in Section VI.

II. EXPERIMENTAL DATASET

The dataset of PHM'08 data challenge is a multivariate time series for each unit. The dataset is further split into training and testing subset. Training dataset consists of 218 units for the same component or system. Each unit begins with a different degree of initial wear and manufacturing variation which are considered normal. Indeed, each training unit starts at a normal state and evolves a fault at a random point during the time series. Finally, the unit runs to failure and the final time point is the unit's lifetime. The testing dataset is also composed of 218 units for the same component or system. Nevertheless, each unit of testing dataset runs to an unspecified position before failure. The challenge is to predict the RUL of all testing units. There is a score method to estimate the performance of prediction method as shown in (1):

$$d_k = \hat{R}_k - R_k$$

$$S_k = \begin{cases} e^{-\frac{d_k}{13}} - 1, d_k \le 0 \\ e^{-\frac{d_k}{10}} - 1, d_k > 0 \end{cases}, k = 1,2,\dots,\mathrm{K} \tag{1}$$

$$S = \sum_{k=1}^{\mathrm{K}} S_k$$

As we can see, $\mathrm{d_k}$ is the difference between RUL prediction and actual RUL. $\hat{\mathrm{R}}_\mathrm{k}$ is the RUL prediction for unit k. $\mathrm{R_k}$ is the actual RUL of unit k. $\mathrm{S_k}$ is the score of unit k. S is total score which is expected to be as low as possible.

All datasets have 26 rows which include the unit ID, cycle index, 3 values for the operational settings and 21 values for sensor measurements. The sensor measurements are contaminated with noise.

A. Operation Settings

After the PHM'08 data challenge, there were some papers described how the dataset was generated. The dataset was a simulation data by the modules of aircraft gas turbine engines using C-MAPSS (Commercial Modular Aero-Propulsion System Simulation)[6]. According to the paper, The 3 value of operational settings can be interpreted as the following parameters:

1. Altitude (ALT)
2. Mach (MA)
3. Throttle Resolver Angle (TRA)

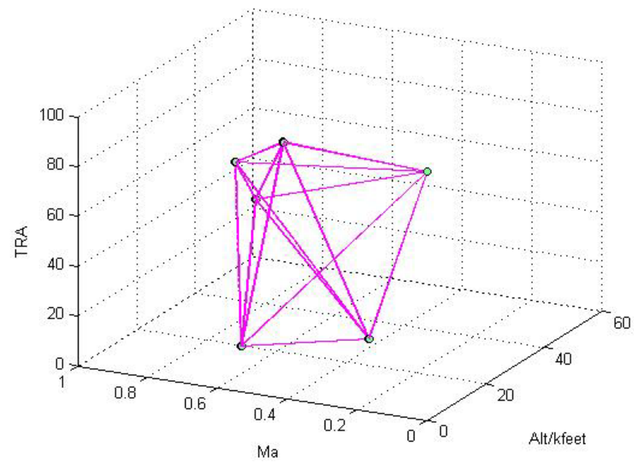

Figure 1. 3D graph for six operating conditions. There are six OCs to express different operating condition, and the line between those dots can reflect their convert relationship.

B. Operation Condition Extraction

The 3-D plot of the three variables shows that the data concentrated in six clusters which present six independent OCs, as shown in Figure 1. We give a definition and an ID for each OCs according to our research in TABLE I.

TABLE I. DIVIDED OPERATING CONDITIONS

operating condition number	ALT	MA	TRA
1(Let down)	20	0.7	0
2(Descent)	9.99	0.25	20
3(Cruise)	42	0.84	40
4(Climb)	34.99	0.84	60
5(Accelerate climb)	25	0.62	80
6（Take off）	0.00	0.00	100

C. Sensor Measurements

The 21 column values for sensor measurements were used to present the system's degeneration information, which could obtain the running status and RUL information. When we drew the figure using one of sensor measurements with all training units, as shown in Figure2, We found the data automatically split into different groups. Some of them had six groups and others had less than six. This phenomenon indicated that sensor values and the six OCs have some relationship. Therefore, taking OCs into consideration was essential at the modeling and prognostic stage which we would describe in next section.

III. OCS MATCHING METHOD

This approach consisted of three essential stages: interpolation, modeling and prognostic. We finished OCs extraction in Operation Condition Extraction, which was an important preprocessed step. Data capacity was reconstructed used to develop models of OCs by interpolation. Then, sensor selection should be done in the modeling stage to improve performance. In prognostic stage, each unit of testing dataset matched with models according to OCs. Finally, we estimated the RULs of all testing units.

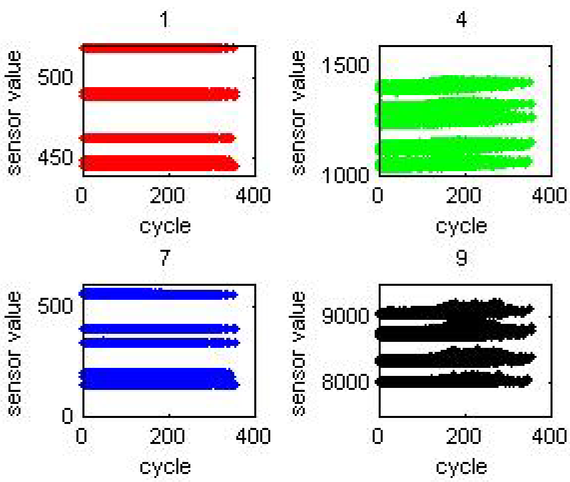

Figure 2. Selected sensors data for all unit in all operating conditions which can be divided to different groups. Some of them have six groups, others have less than six. There are some relationship with six operating condition.

A. Interpolation

Each unit worked in hopping OC over time, which changed from the six OCs. In order to develop prior regression models with different OCs using training dataset, we adopted the interpolation approach to expand the scale of data. This approach was used in each unit of training dataset. In the first step, the dataset was grouped into six based on OCs. Then, the length of each group was expanded to the unit's lifetime by the interpolation. In this way, each group had same length with the unit's lifetime. Finally, we got six groups for one unit, which each group showed a unit working in one OC over time. The same handling method was adopted in all training units. The scale of dataset was changed from 218*21 to 218*21*6 (218 means the unit number; 21 means the rows of sensor measurements; 6 means six OCs). We used all expanded training dataset in the following steps. There was a confusion that we need to classify. Because the values of interpolation nodes were influenced by those points that worked in other OCs, The interpolation results couldn't present the curve's trend acting in sole OC accurately. However, the testing units worked in the same situation, which continuously changed OCs over time. Furthermore, the interpolation result could reflect the changed rule of OCs reasonable.

TABLE II. INTERPOLATION METHOD COMPARISON

approach	advantage according to training data	disadvantage according to training data
nearest	Zero order remain; keep the original data	Not smooth
linear	With small extrapolation deviation	Not very smooth
spline	Smooth	With great extrapolation deviation
pchip	Smooth	With great extrapolation deviation

There were some methods for interpolation we tested to find a suitable one, as shown in TABLE II. In our study, 'spline' and 'pchip' had extrapolation deviation, which were more likely to lead to errors. It was not a good choice. In order to model, smoothness should be taken into consideration. Therefore, we chose 'linear' interpolation.

B. Modeling

Regression models were used to present the regularity of degeneration. When we finished the previous steps, we built 218*6 degeneration model, which each model represented a unit in a specified OC. The multi-dimensional sensor measurements couldn't show the trend of degeneration intuitively. A single Health Indicator (HI) was produced to show the system's health state over time. HI ranged from 0 to 1, which 0 means the system was failure and 1 means the system was health[3]. Assuming that degradation grew exponentially, the target HI at each time could be expressed as:

$$\begin{gathered} HI_y = A + B \times \log(len + 1 - t) \\ A = yf, B = (y0 - yf)/\log(len + 1) \end{gathered} \tag{2}$$

HI_y is the goal output. len is the lifetime of one unit. A is the connecting point parameters, which can express the state of system in the OCs; B is the rate of degradation, which can reflect the degree of each OC's impact. y0 is the degradation start point. yf is the degradation terminal point. User can set the two parameters (y0, yf) depending on different situation and requirement. In order to preserve the original patterns of sensor measurements, a linear regression model is adopted, as shown in (3).

$$\begin{gathered} y = \beta^T X + \varepsilon \qquad \varepsilon \sim N(0, \sigma^2) \\ \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} \quad X = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_{11} & x_{21} & \cdots & x_{n1} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1m} & x_{2m} & \cdots & x_{nm} \end{bmatrix} \end{gathered} \tag{3}$$

Where β is the coefficient vector; X is sensor matrix and ε is the noise term.

There were 21 sensor serials which presented different trends, as shown in Figure 3. Some sensors changed monotonically, which could be used in regression models. However, some sensors contained high noise, or were not sensitive to degradation like unit 1 in Figure 3. If the models included them, the accuracy of prediction may decrease. Therefore, Sensor selection before modeling was necessary. In Tianyi Wang's paper[3], the approach of selection relied on observation and used the same combination in all OCs. This approach couldn't select sensors automatically and didn't take OCs into consideration. In our study, a stepwise approach was used to select sensors. Indeed, each factor was selected according to the sum of squares of partial regression. The maximum one was chose. Then, significant test was done to ensure the accuracy of the model in every step. There was a threshold, which need to be set, to determine how many factors could enter the model. The selection was based on the expanded training dataset, which contained six groups for each unit. For one unit, we used stepwise in each group to get corresponding model in different OCs.

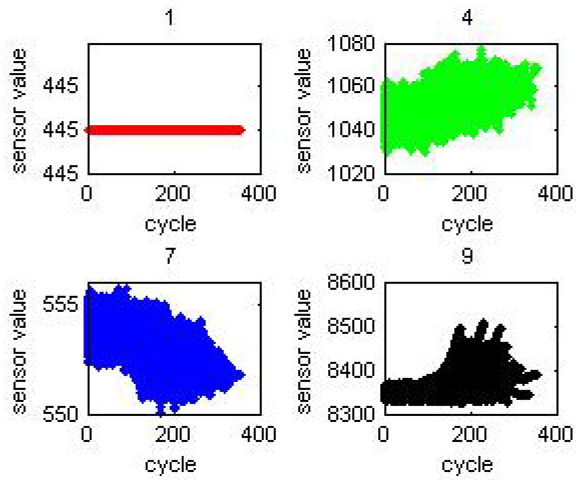

Figure 3. Different sensor values for all units in one OC. Form the figure, we can find that unit1 remain unchanged over time; unit4 monotone increasing over time; unit7 monotone decreasing over time; unit9 scatter over time.

Different units had different lifetimes in training dataset. It was influenced by the combination of OCs and the acting time of each OC, which was reasonable in physical and mathematical results. In physical sense, the engine worn out most in climb phase and least in land phase. The degree of wear was difference in each OC. HI and the degree of physical damage were consistent in mathematical sense. Therefore, developing model with six OCs was useful in prognostic stage.

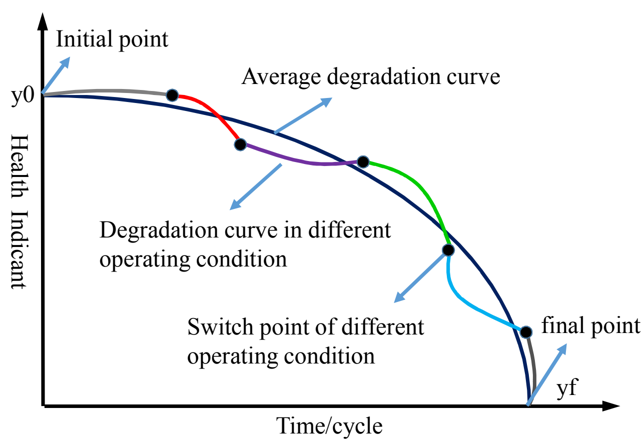

Figure 4. The remaining useful life prognostic strategy. Different with creating the average degradation curve, piecewise curves are developed based on operating conditions. The curve is a multi-track curve.

C. *Prognostic*

Extrapolating curve was a traditional approach to predict RUL. However, if the run-to-failure data couldn't be fully utilized, extrapolation may produce large errors. Neural network was a good approach for prediction. But the training progress was a black box, which we couldn't have good understanding on the inner coding progress. Once step further, there weren't systematic approaches to design the structure and parameters. All the things made it hard to improve the performance of prognostic.

In our study, we proposed a segmentation matching method to predict RUL based on OCs. As we mentioned previously, traditional method could only develop the average degradation curve. But each system (or component) differed on manufacture, maintenance condition and working load. The average curve couldn't fit all situations. Indeed, the real degradation curve was multi-track, as shown in Figure 4. Our prognostic approach was to establish the degradation curve in different OCs. In this way, the model could fit different environments or requirements. The detail method had four steps. We took one testing unit as an example to explain. The specific steps were:

- Step 1. Updating prior dataset. For a testing unit (Tk, k=1,2,…,218), we gained the number of cycle (Ck, k=1,2,…,218). Then we searched all training units, which lifetime was more than the Ck. If the lifetime of the training unit was less than the Ck, matching progress had no significance that the prediction must be wrong. So we got all useful training units of the given Tk (k=1,2,…,218). These training units were used to match with the Tk. The number of total useful training units was Nk (k=1,2,…,218). In this way, we updated the specific prior training dataset for corresponding testing units.

- Step2. Segmenting dataset. Before matching testing units with prior dataset, there were some preparations that we needed to do. We segmented the testing unit data depending on different OCs and recorded the total number of data segments as Mk(k=1,2,…,218). At the same time, we extracted the model parameters and sensor combinations of Nk training models in all OCs. Then, we used them to get all HI segment for the testing unit data in its OC and marked as $\tilde{y}_{ij}$(i=1,2,….,Mk, j=1,2,….,Nk).

- Step 3. Matching segments. The HI segments for corresponding Nk training units were marked as y_{ij}(i=1,2,…, Mk,j=1,2,….,Nk). Then, we calculated the markov distance between $\tilde{y}_{ij}$ and y_{ij}(used same i,j to do all match) according to formula (4) and found out the segment with the minimum distance. The corresponding $\tilde{y}_{ij}$ was the final HI segment we chose for the testing unit.

$$d_{ij} = \left(y_{ij} - \tilde{y}_{ij}\right)^T S^{-1}\left(y_{ij} - \tilde{y}_{ij}\right) \quad (i = 1,2, \dots, Mk, j = 1,2, \dots, Nk \tag{4}$$

In formula (4), the S^{-1} was the covariance matrix between $\tilde{y}_{ij}$ and y_{ij}.

- Step 4. This was the final step. When we got all the best HI segments, A fusion calculation was needed to get RUL. The total cycle number of each training unit was the corresponding lifetime. In matching progress, we got the total training units' matching times for each testing unit (R_j j=1,2,…218, j means testing unit number). The matching times of each training unit were described as $a_{ij}(i=1,2,\ldots,218, j=1,2,\ldots 218)$ (i means training unit number. j means testing unit number. For example, the value of a_{58} was Num.5 training unit' s matching times for Num.8 testing unit). The result was a fusion and calculated by (5).

$$\mathrm{life_j} = \sum_{i=1}^{218} \frac{a_{ij}}{R_j} \times \mathrm{life_i} \tag{5}$$

The result was the lifetime of a testing unit. We could get the run cycles of the unit. The difference between them was RUL of the testing unit. All testing units could use this approach to get their RUL. There were two issues in step3 that we needed to explain:

- Firstly, we used Markov distance to evaluate the degree of matching. Markov distance is the data covariance distance. It is different from Euclidean distance, because all factors can be taken into consideration. It could rule out the influence of correlation interference variables of the sensor measurements, which suited for the quantify similarity of the two HI segments. Meanwhile, it had no limit of the dimensional, which was useful for this dataset with different dimension.

- Secondly, we chose the segment with minimum distance as the matching one. The reason was that the distance between training and testing segments reflected their similar degree. Furthermore, each training segment was influenced by its previous OCs, which could indicate different degree of wear. So, the testing segment with the minimum distance could describe the status of the testing unit best.

This matching progress described the OCs influence for the units instead of establishing average degradation curve directly. In our research, we simulated the real working process of the system like Figure 4.

IV. PROCEDURES

To achieve the method using experimental dataset, nine procedures were developed. All procedures were divided into three stages: per-process, modeling and estimate, as shown in Figure5. This was a schematic diagram which could detail the process and the structure of each matrix. Furthermore, the figure could show the location and the effect of each procedure and link them together. We used matrix thinking in all procedures to improve program efficiency. Firstly, we built needed matrices. Then the procedures were designed based on them. In this way, we designed the program by data stream. In order to improve the operating speed of the program, we made some changes when we finished the first generation. We changed the methods to manipulate the data matrix. Indeed, We tried to operate a chunk of data every time and redesign the matching processes. Firstly, We collected each unit data segments with different OCs and then operated them together for the sake of achieve our design. After those changes, we finished an efficient matching instead of point by point which was the original approach with slow speed. The final version of program finished all units prognostic in less than half an hour (average run time was 984.634 seconds in our experiment) which was one-tenth running time comparing with original program. The detail description of program is following.

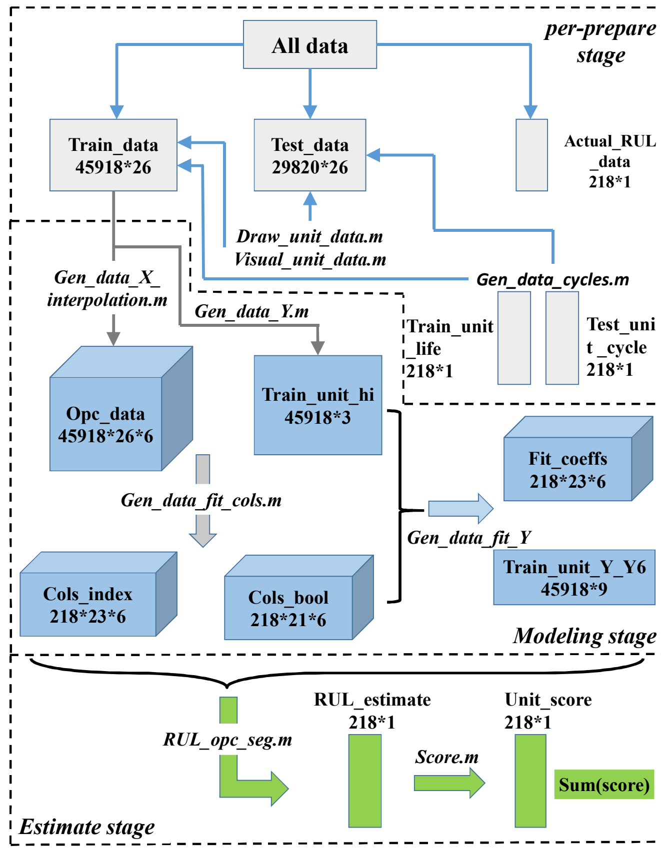

Figure 5. RUL prognostic procedures flow diagram to present each procedure's effect.

A. Per-prepare stage

The PHM'08 dataset was needed to pre-process before developed the regression models. Firstly, the training matrix (45918*26) included 218 units. In the matrix, the first array represented the unit number; the second array represented operating cycle; the 3th to the 5th arrays were operational settings; the 6th to 26th arrays were sensor measurements. This was the standard form of PHM'08 challenge dataset. The testing matrix (29820*26) had the same structure. Finally, the actual RUL matrix of all testing units was used to score. After develop all data matrices, there were some programs to extract specific number unit and data visualization.

1) Draw_unit_data.m

We used the procedures to draw specific number unit. The input matrix was the whole matrix like training matrix or testing matrix. and the unit number which was needed to extract. The output matrix was the matrix with specific number unit which contained all information as the standard structure and the operating cycle of the unit.

2) Visual_unit_data.m

It was used to analyze OCs and the trend of different sensors, As shown in Figure 1,2,3.

3) *Gen_data_cycles.m*

This procedure was used to get the length of units and develop the cycle matrix (218*1). There were two cycle matrices. One cycle matrix was lifetime matrix for all training units. Another cycle matrix was operational cycle matrix for all test units. Using them to add the corresponding unit's RUL, the result that we got equaled to the units' lifetime.

All the pre-process works could offer convince to operate chunks of data. In that way, we could improve the coding efficiency.

B. *Modeling stage*

The training matrix was used to build multiple regression models which established corresponding relationship between sensor measurements and HIs. In order to get them in all OCs, interpolation was employed to expand the size of the training matrix. Then, we applied the stepwise to select sensors before develop regression models. Finally, the model matrix (45918*3) could be found. The first array was unit number, the second array was operating cycle and the third was HI. There were some extra matrices generated that could be used in the estimate stage.

1) *Gen_data_X_interpolation.m*

This procedure finished the interpolation task. According to Method part, we adopted a linear interpolation for the training matrix. Interpolation made the original two-dimensional matrix (45918*26) changed into a three-dimensional matrix (45918*26*6) named Opc_data. Meanwhile, there was something we needed to note:

- Firstly, column length of each unit in each OC was the same as the original length of the unit. Using unit 1 as an example, the size of original matrix was 213*26 and the result of interpolation was 213*26*6, which the unit worked in six OCs had same operating cycle with original one.
- Secondly, interpolation didn't suit for the testing matrix. Because the amount of testing dataset was not enough, it couldn't provide enough interpolation nodes, which was likely to cause big errors. Like unit 24, it only had one point in OC 1.

2) *Gen_data_Y.m*

The target Y used in regression progress was calculated by formula (2). It reflected the average degradation like the dark blue curve in Figure 4. We assumed the failure mode of each unit was worn. So, the initial HI was set to 0.99 (y0) and the final HI was 0 (yf) to show failure. Then we got the value of A,B and target HI matrix (45916*3*6) in six OCs named Tar_hi.

3) *Gen_data_fit_cols.m*

We employed this program to finish sensor selection which was a vital step. Stepwise could select and test automatically. The maximum p value for a term to be added was set to 0.05. The minimum p value for a term to be removed was set to 0.10. The result of selection was different combination for varying OC in each unit. Unit 1 is an example, as shown in TABLE III.

TABLE III. THE SIX REGRESSION MODEL OF UNIT 1

operating condition	**sensor number**
1	2 7 10 11 12 13 14 15 16 17 20 21
2	6 7 8 9 12 13 15
3	3 4 6 7 8 11 15 20 21
4	2 3 4 6 7 9 11 12 13 14 15 17 21
5	2 3 6 9 11 12 15 17 20 21
6	2 3 7 8 9 11 12 14 20

There was no intersection for six OCs, which we made deeply researched. No matter how to broaden the stepwise parameters, intersection was still inexistence. We thought the result was reasonable after testing and discussing, because the trend of degeneration was different in diverse OCs.

There were two means to get training unit model. One was getting an average model of each unit. in this way, we had to use all results and do frequency statistics, because of no intersection between models with different OCs. Another was reserving the six models and using all of them in the next step. We adopted the second method for the following reason. Developing the average model must reselect sensors, which may lose information. The first approach conflicted with the demand of prognostic which required as much information as possible. The second approach provided sufficient prior information for 218 testing units which RUL of testing units could predict based on them.

There were some useful matrices generated in this stage. The cols_bool (218*21*6) was a logic matrix whose output was 0 or 1. It showed the selection result of sensors, 1 means selected, and 0 means unselected. The column number was the relative position of sensors which ranged from 1st to 21st Another three-dimensional matrix named col_index(218*23*6) which first array was unit number and second was total number of selected sensors. The other arrays were same as cols_bool, but the column number was the absolute position of sensors, ranging from 6th to 26th.

4) *Gen_data_fit_Y.m*

The final step of modeling was to get the real HI matrix. Comparing glmfit with robustfit, they had similar maximum deviation. But the second one had smaller residual variance, which the output curve had less fluctuation. So, we adopted robustfit to generate the coefficient matrix and named fit_coeffs (218*23*6). The first array was unit number, second was total number of selected sensors, and the 3rd to 23rd were the coefficients ranged by sensor position, which put 0 in vacancy position. The real HI matrix was Training_unit_Y_Y6 (45918*9). The first array was unit number; the second was operating cycle; the third was the target HI; the real HI segment with six OCs was ranged in 4th to 9th.

C. *Estimate stage*

The final stage was matching and estimate. RUL_estimate matrix (218*1) was generated to show the prognostic result.

We used unit_score matrix to evaluate the effectiveness of our algorithm.

1) Rul opc_seg.m

The detail of matching was introduced in Section III. The first step was updating the prior model set according to given testing unit. We chose the training units which the lifetime was longer than the testing unit's run time. This step was achieved by comparing life matrix with cycle matrix. However, there was only one exception (unit 31) that we should treat special. Its run time (364) was longer than any training units. The way to calculate this unit was using all training dataset to do statistical analysis. We obtained the mean lifetime and the standard deviation by all training units to build normal distribution. Then, we got the RUL of the special unit. The result showed that the predicting effect was neither accurate nor significant deviation. This indicated that the way was a reasonable way to handle the special unit. Indeed, the basic statistical method maybe a good aiding for predicting RUL.

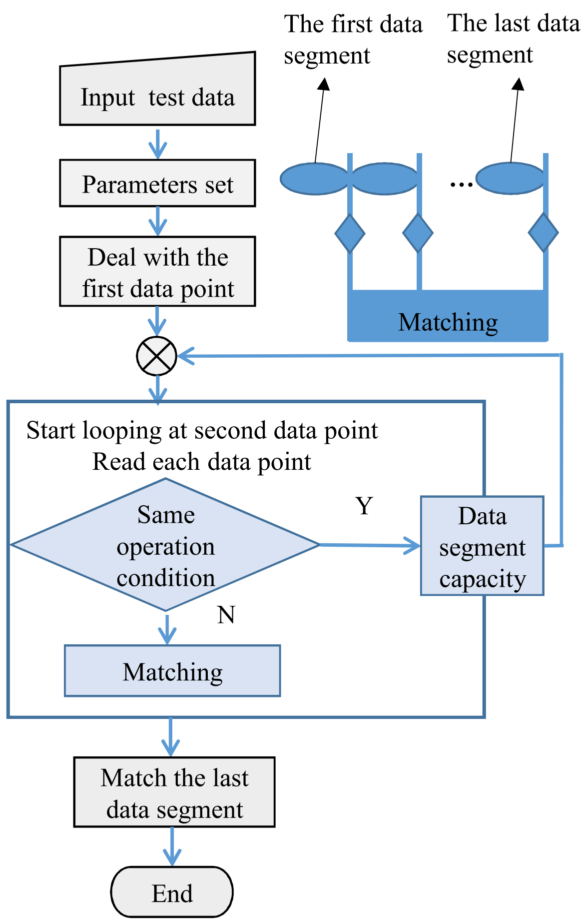

Figure 6. algorithm flow chart of the matching process

Matching process was showed in Figure 6. There were some notices in matching progress:

- Before entering the loop, the first point should be processed as the initialization parameters.
- Because the judgment of loop base on OCs, the final data segment couldn't be matched owing to no OC hopping. So After the end of the loop, the final segments had to be processed.

After matching, the output was the HI sequence of each unit which could be transformed to lifetime sequence using formula (5). We generated the RUL_estimate matrix for testing units by lifetime sequence and Test_unit_cycle matrix.

2) Score.m

According to formula (1) provided by PHM'08 data challenge, this procedure was designed to get score. First, we made the difference between RUL_estimate matrix and Autual_RUL matrix. Then, we got a positive matrix and a negative matrix. The calculation was found by the piecewise functions. Finally, The Unit_score matrix and sum(score) could be gained.

V. RESULT AND ANALYSIS

Figure 7 showed the distribution of scores for each unit. We found several large error among all predictions contributed to a big part of the final score. Meanwhile, most results concentrated in a small range. For example, the worst case (unit 35) was a late prediction of 77 cycles, which received a penalty score of 2259.482. As comparison, the best case was an early prediction of 0.072 and got a penalty score of 0.006. This was a more accurate prognostic comparing with Tianyi Wang's method[2] which the best case had a score of 1.72.

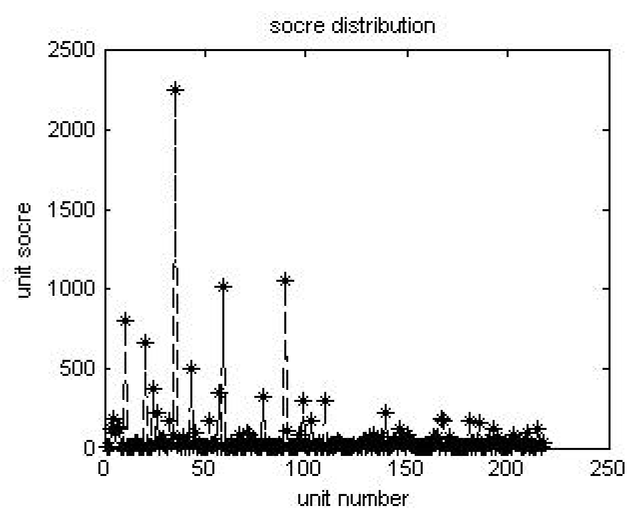

Figure 7. The distribution of scores.

The statistical results indicated that units' score less than 1 can reach 39 (17.8%), which the difference of prediction ranged from -8 to 6. The score less than 10 was 108 (49.5%), which the difference of prediction ranged from -30 to 22. Some best cases and worst cases were showed in TABLE VI. The mean squared prediction error was 364.632 which was better than Leto Peel et al. [4] approach with 984.

TABLE IV. COMPARING THE BEST AND WORST CASES

type	unit number	run cycles	score
max	35	60	2259.482
	59	58	1012.018
	90	23	1048.975
min	83	108	0.036
	94	125	0.006
	119	148	0.028

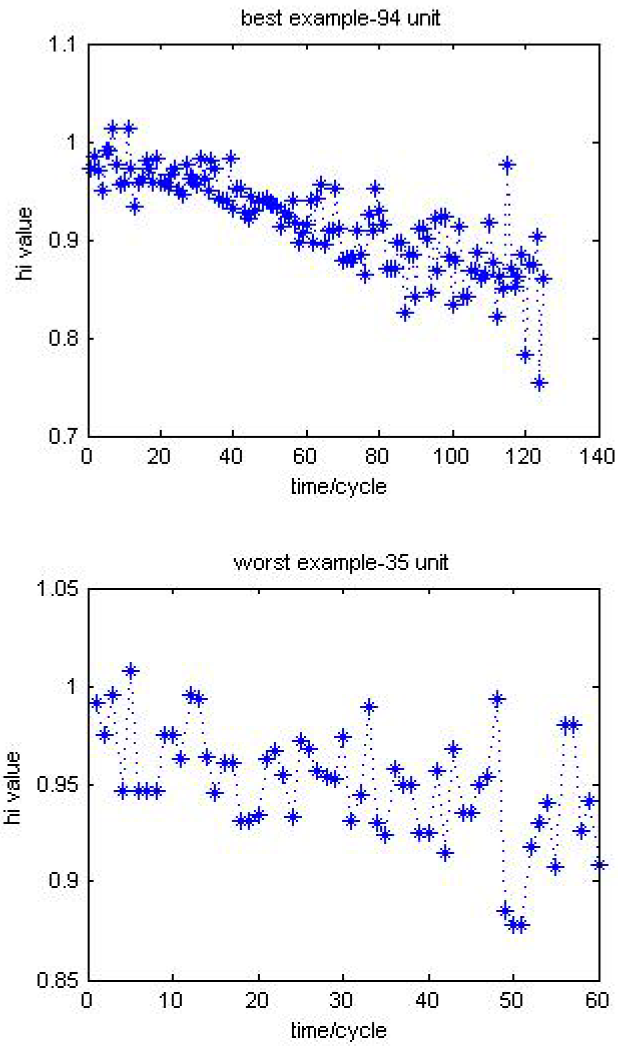

Figure 8. Comparing the HI curve of best case (unit 94) with worst case(unit 35).

Comparing the HI curve of best case (unit 94) with worst case (unit 35), as shown in Figure 8. The few bad predictions could have a reasonable explanation. The run cycles of them were short, which meant there was no fully prior knowledge to predict the RUL. However, the best case which had longer run cycles contained enough prior knowledge. There were other factors that may affect the accuracy of prognostic. The basic hypothesis in the study was that the failure pattern was worn, which suited for most situations. The HI curve trend of best case met the basic laws of degradation and the fluctuation within the permissible range of error. However, there may be some exceptions which do not fit the failure mode which we adopted. Besides, as matching based on different OCs, Particularly short segments that used in matching could cause large errors. The effect of those factors should be taken into consideration when we evaluated the results. Excluding the six units with large errors, the mean score of each unit could reach 36 (the original mean score is 64). There were 156 units (71.6%) below 36. A total score of 14074.66 was achieved, which was a good rank in all results according to challenge official website[16].

VI. CONCLUSION AND FUTURE WORK

This paper described a data-driven approach to predict remaining useful life. This approach paid more attention on operating conditions. It could expand the prior dataset which was used to predict. At the same time, it could simulate the real situation of systems. The results had shown that the prediction error can be reduced if we could examine enough run cycles.

Suggestion for further work would be to investigate different degradation model to improve the performance of matching based on operating conditions and study the score sequence. In this work, all degradation patterns were based on wear. But it is possible that other forms of failure exist in the testing and training dataset. In the future, we can develop the prior model set. In this way, more models can be built based on different model to expand the prior model set. Meanwhile, more work could be done on the matching process to improve the accuracy of units with short run cycles. Those suggestions for further improvement could be made by studying the score sequence for all units.

ACKNOWLEDGMENT

The authors want to thank the anonymous reviewers for their help and constructive comments. The authors also want to thank Beijing Natural Science Fund Committee and the Fundamental Research Funds for the Central Universities for their funding.

REFERENCES

[1] S J Engel, B J Gilmartin, K Bongort, A Hess, "Prognostics, The Real Issues Involved With Predicting Life Remaining," 2000 IEEE Aerospace Conference Proceedings,2000, pp. 457-469.

[2] Chen Xiongzi, Yu Jinsong et al, "Remaining Useful Life Prognostic Estimation for Aircraft Subsystems or Components: A Review", The Tenth International Conference on Electronic Measurement & Instruments,2011.

[3] Wang T, Yu J, Siegel D, Jay Lee,"A similarity-based prognostics approach for remaining useful life estimation of engineered systems, " 2008 International Conference on Prognostics and Health Management, 2008.

[4] Felix O. Heimes, BAE Systems, “Recurrent Neural Networks for Remaining Useful Life Estimation”, 2008 International Conference on Prognostics and Health Management,2008.

[5] Leto Peel et al, “Data Driven Prognostics using a Kalman Filter Ensemble of Neural Network Models”, 2008 International Conference on Prognostics and Health Management,2008.

[6] M.Abbas, and G J. Vachtsevanos, “A System-Level Approach to Fault Progression Analysis in Complex Engineering Systems”, Annual Conference of the Prognostics and Health Management Society, 2009,PP.1-7.

[7] Sony Mathew, Diganta Das et al, “Failure Mechanisms Based Prognostics”, 2008 International Conference on Prognostics and Health Management, 2008.

[8] Roger K. Youree, Jeffrey S. Yalowitz et al, “A Multivariate Statistical Analysis Technique for On-Line Fault Prediction”, 2008 International Conference on Prognostics and Health Management, 2008.

[9] K. Medjaher, N. Zerhouni, J. Baklouti, “Data-Driven Prognostics Based on Health Indicator Construction: Application to PRONOSTIA’s Data”, 2013 European Control Conference, Zürich, Switzerland,2013

[10] Jamie B. Coble and J. Wesley Hines, “Prognostic Algorithm Categorization with PHM Challenge Application”, 2008 International Conference on Prognostics and Health Management, 2008.

[11] Alexandr Kirilloy ,Sergey Kirilloy et al, “The Calculating PHM Cluster: CH&P Mathematical Models and Algorithms of Early Prognosis of Failure”, 2012 Prognostics & System Health Management Conference,Beijing,2011.

[12] Serdar Uckun, Kai Goebel et al, “Standardizing Research Methods for Prognostics”, 2008 International Conference on Prognostics and Health Management, 2008.

[13] Muheng Wei, Maoyin Chen et al, “Multi-Sensor Information Based Remaining Useful Life Prediction With Anticipated Performance”, IEEE TRANSACTIONS ON RELIABILITY, 2013.3,62(1) .

[14] Xiao-Sheng Si, Wenbin Wang et al, “Remaining Useful Life Estimation Based on a Nonlinear Diffusion Degradation Process”, IEEE TRANSACTIONS ON RELIABILITY, 2012,3,61(1).

[15] Pingfeng Wanga, ByengD.Youn et al, “A generic probabilistic framework for structural health prognostics and uncertainty management”, Mechanical Systems and Signal Processing,2012,28,622-637.

[16] "Challenge Problem";http://www.phmconf.org/OCS/index.php/phm/2008/challenge.

[17] Liang Tang,Jonathan Det.astro et al, “Filtering and Prediction Techniques for Model-Based Prognosis and Uncertainty Management”, 2010 Prognostics & System Health Management Conference, 2012.

[18] Zhang Lei, “A Multivariate Relevance Vector Machine Based Algorithm for On-Line Fault Prognostic Application with Multiple Fault Features”, 2012 Fifth International Conference on Intelligent Computation Technology and Automation,2012.