A Prognostic Driven Predictive Maintenance Framework Based on Bayesian Deep Learning

Liangliang Zhuang, Ancha Xu, Xiao-lin Wang*

Zhejiang Gongshang University

目录

Introduction

- The developments of **low-cost sensing** and **monitoring technologies** make more industrial systems equipped with on-board sensors to monitor their health conditions.
- The collected data can be used to predict system failures and thus guide maintenance scheduling $\Rightarrow$ **predictive maintenance (PdM)**.
- However, the existing PdM literature separates two inter-related stages—**prognostics** and **maintenance decision making**
  - Either studies remaining useful life (RUL) prognostics without considering maintenance issues.
  - Or optimizes maintenance plans based on given prognostic information.

Prognostic driven PdM

- Nguyen and Medjaher [8] were the first to study prognostic driven PdM problems. They utilized an LSTM network to predict system failure probabilities at future time intervals, which drive **maintenance** and **spares ordering decisions** upon **periodic inspections**.
- Other papers: [1–3].

Shortcomings

1. Cannot capture prognostic uncertainties.
2. Maintenance decisions can be freely implemented. In practice, however, maintenance execution is largely constrained by operational schedules (e.g., aircraft maintenance).
3. Cannot recommend long-term maintenance plans with evolving prognostics.

Contributions

- **Prognostic stage**: we adopt the BDL-based framework in [5] to characterize two type of **prognostic uncertainties** and produce a predictive RUL distribution.
  1. *Epistemic uncertainty*: the lack of knowledge on the true model and can be reduced by acquiring more information.
  2. *Aleatoric uncertainty*: concerned with random, uncontrollable disturbances in sensory data such as measurement errors.
- **Maintenance decision-making stage**: update *maintenance* and *spares ordering decisions* with the latest prognostic information, while satisfying **operational constraints** on maintenance execution.

目录

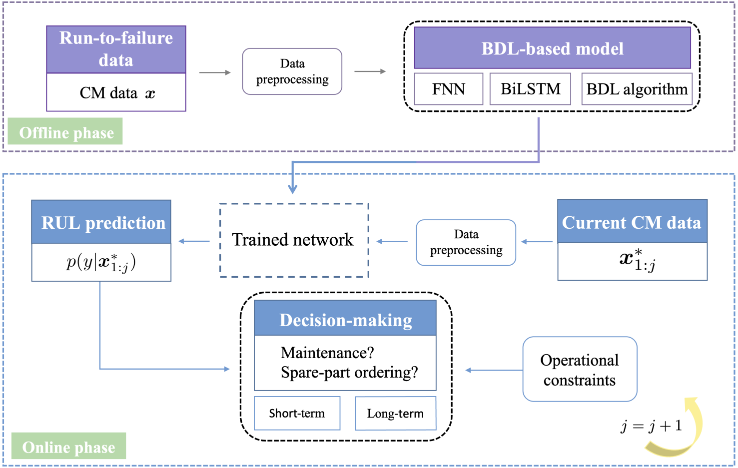


Figure 1: Proposed prognostic driven dynamic PdM framework.

Two type of uncertainties

$$y = f(\boldsymbol{x}; \boldsymbol{\Theta}) + \epsilon, \tag{2.1}$$

where $f(\cdot; \boldsymbol{\Theta})$ represents a functional mapping with parameters $\boldsymbol{\Theta}$ and $\epsilon \sim N\left(0, \eta^2\right)$ is a Gaussian noise term with mean zero and variance η^2.

1. **Epistemic uncertainty** arising from $\boldsymbol{\Theta}$ and **aleatoric uncertainty** captured by η^2.
2. Both would contribute to divergence between prognosis and actual results.

BDL-based network for RUL prognostics

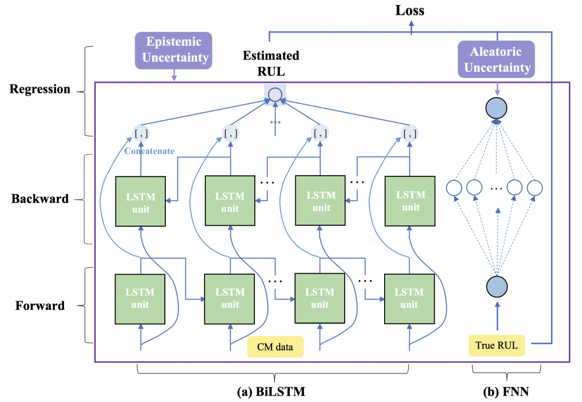


Figure 2: BDL-based RUL prognostics at the training time.

Prognostic driven maintenance decision making

- Update **maintenance** and **spares ordering** decisions dynamically, while satisfying **operational constraints** on maintenance execution.

Assumption

i) Maintenance is perfect through replacing systems with new identical spares.

ii) Spare parts are ordered only when needed so as to minimize inventory holding costs, and the lead time is a constant, denoted by $\mathcal{L}$.

iii) Maintenance can be executed when spare parts are unavailable, but it incurs an extra out-of-stock cost.

iv) Maintenance activities can only be executed in a series of time windows $\mathcal{S} = \{[t_{d_1}, t_{e_1}], \ldots, [t_{d_s}, t_{e_s}]\}$ and can be completed within a single period, whereas spare parts can be ordered at any time.

Tentative PdM scheduling with operational constraints

- Given the current time t_j, the probability density function of RUL, $p(y|x^*_{1:j})$, for any y can be obtained via the prognostic framework.
- c_p: preventive maintenance cost; c_c: corrective maintenance cost; c_{os}: out-of-stock cost when spare parts are not available; c_f: cost of wasting a unit of system RUL; c_q: spare-part holding cost per unit time.

Two scenarios

$$\begin{aligned} \mathcal{C}^{\mathrm{R}}_{j,j+k} &\approx \sum_{h=0}^{k-1} p_{h|j} \frac{c_c + c_{os}}{t_{j+h}} + \sum_{h=k}^{+\infty} p_{h|j} \frac{c_p + c_f(h-k)\Delta t}{t_{j+k}}, \\ \mathcal{C}^{\mathrm{DN}}_{j,j+k} &\approx \sum_{h=0}^{k+1} p_{h|j} \frac{c_c + c_{os}}{t_{j+h}}. \end{aligned} \tag{2.2}$$

Tentative PdM scheduling with operational constraints

The optimal action at time t_{j+k} is thus the one with a lower cost rate:

$$\mathcal{A}_{j,j+k} = \begin{cases} R, & \text{if } \mathcal{C}^{\mathrm{R}}_{j,j+k} \leq \mathcal{C}^{\mathrm{DN}}_{j,j+k}, \\ DN, & \text{if } \mathcal{C}^{\mathrm{R}}_{j,j+k} > \mathcal{C}^{\mathrm{DN}}_{j,j+k}. \end{cases} \tag{2.3}$$

Remark

The framework can not only make an instantaneous maintenance decision at the current time t_j, but also recommend a long-term decision for any future moment t_{j+k}.

Tentative maintenance time

$$\mathcal{T}_j^{\mathrm{m}\prime} = \mathcal{T}_j^{\mathrm{m}} \cdot \mathbb{1}_{\{\mathcal{T}_j^{\mathrm{m}} \in \mathcal{S}\}} + \tilde{\mathcal{T}}_j^{\mathrm{m}} \cdot \left(1 - \mathbb{1}_{\{\mathcal{T}_j^{\mathrm{m}} \in \mathcal{S}\}}\right), \tag{2.4}$$

where $\mathbb{1}_{\{\cdot\}}$ is the indicator function that equals to 1 if the argument is true, and 0 otherwise. In this expression, $\mathcal{T}_j^{\mathrm{m}} = \inf_{k \in \{0,1,\dots\}} \{t_{j+k} : \mathcal{C}_{j,j+k}^{\mathrm{R}} \leq \mathcal{C}_{j,j+k}^{\mathrm{DN}}\}$ is the tentative maintenance time when it is in the window $\mathcal{S}$, while $\tilde{\mathcal{T}}_j^{\mathrm{m}} = \inf_{\zeta \in \{\alpha,\beta\}} \{t_\zeta : \mathcal{C}_{j,\zeta}^{\mathrm{R}} \leq \mathcal{C}_{j,\zeta}^{\mathrm{DN}}\}$ is the tentative maintenance time when $\mathcal{T}_j^{\mathrm{m}} \notin \mathcal{S}$.

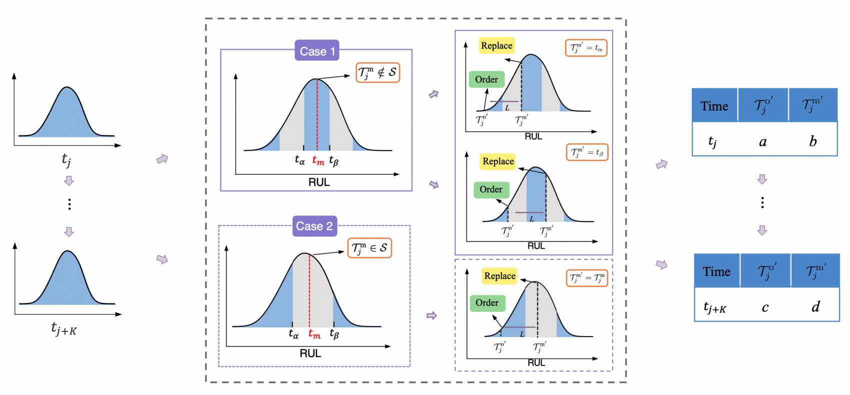


Figure 3: Tentative PdM scheduling with operational constraints.

Tentative ordering time

To reduce the fluctuation due to dynamic updating, the predictions $\mathcal{T}_j^{\mathsf{m}'}$'s for the recent Q cycles are averaged to determine $\mathcal{T}_j^{\mathsf{o}'}$:

$$\mathcal{T}_j^{\mathsf{o}'} = \left\lfloor \frac{\sum_{q=0}^{\min\{j-1,Q-1\}} \mathcal{T}_{j-q}^{\mathsf{m}'}}{\min\{j, Q\}} \right\rfloor - \mathcal{L}. \tag{2.5}$$

where $\lfloor x \rfloor = \max\{n \in \mathbb{Z} \mid n \leq x\}$.

Dynamic PdM updating and adjusting

With successively updated $(\mathcal{T}_j^{\mathsf{m}'}, \mathcal{T}_j^{\mathsf{o}'})$, we update a dynamic PdM.

Optimal ordering time

$$\mathcal{T}^{\mathsf{o}^*} = \inf_{j \in \mathbb{Z}^+} \{t_j : \mathcal{T}_j^{\mathsf{o}'} \leq t_j\}. \tag{2.6}$$

Predicted maintenance time

$$\mathcal{T}^{\mathsf{m}'} = \inf_{t_j \in \mathcal{S}} \{t_j : \mathcal{T}_j^{\mathsf{m}'} \leq t_j\}. \tag{2.7}$$

Mismatch $(\mathcal{T}^{\mathrm{o}^*} + \mathcal{L} \neq \mathcal{T}^{\mathrm{m}'})$, resulting in **spare-part holding** or **shortage costs**.

Let $t_a = \mathcal{T}^{\mathsf{o}^*} + \mathcal{L}$, and $t_b = \mathcal{T}^{\mathsf{m}'}$. To assess a series of average cost rates over the interval $[\min\{t_a, t_b\}, \max\{t_a, t_b\}]$, there are three scenarios:

$t_a < t_b$

$$\mathcal{T}^{\mathsf{m}^*} = \underset{t_{a+j} \in \mathcal{S}}{\arg\min} \left\{\mathcal{C}_{a,a}, \cdots, \mathcal{C}_{a,a+j}, \cdots, \mathcal{C}_{a,b}\right\}, \tag{2.8}$$

where

$$\mathcal{C}_{a,a+j} = \sum_{h=0}^{j-1} p_{h|a} \frac{c_c + c_q h}{t_{a+h}} + \sum_{h=j}^{+\infty} p_{h|a} \frac{c_p + c_q j + c_f (h-j)}{t_{a+j}}$$

for $j = 0, 1, \ldots, b-a$.

$t_a > t_b$

$$\mathcal{T}^{\mathrm{m}^*} = \underset{t_{b+j} \in \mathcal{S}}{\arg\min} \left\{ \mathcal{C}_{b,b}, \cdots, \mathcal{C}_{b,b+j}, \cdots, \mathcal{C}_{b,a} \right\}, \tag{2.9}$$

where

$$\mathcal{C}_{b,b+j} = \sum_{h=0}^{j-1} p_{h|b} \frac{c_c + c_{os}}{t_{b+h}} + \sum_{h=j}^{+\infty} p_{h|b} \frac{c_p + c_{os} \cdot \mathbb{1}_{\{j<a-b\}} + c_f (h-j)}{t_{b+j}}$$

for $j = 0, 1, \ldots, a-b$.

$t_a = t_b$

$$\mathcal{T}^{\mathrm{m}^*} = \mathcal{T}^{\mathrm{m}'}. \tag{2.10}$$

Performance evaluation of the dynamic PdM policy

Actual cost rate of the r-th life cycle (runs R life cycles)

$$CR_r = \begin{cases} \dfrac{c_p + c_{os}\delta_r + c_q\kappa_r(1-\delta_r) + c_f \sum_{h=1}^{+\infty} h p_{h|\psi_r}}{\mathcal{T}_r^{\mathrm{m}^*}}, \mathcal{X}_r = \mathcal{T}_r^{\mathrm{m}^*}, \\ \dfrac{c_c + c_{os}\delta_r + c_q\kappa_r(1-\delta_r)}{\mathcal{T}_r^{\mathrm{f}}}, \mathcal{X}_r = \mathcal{T}_r^{\mathrm{f}}, \end{cases} \tag{2.11}$$

Average cost rate for all life cycles

$$\overline{CR} = \frac{\sum_{r=1}^{R} \mathcal{X}_r \cdot CR_r}{\sum_{r=1}^{R} \mathcal{X}_r}. \tag{2.12}$$

目录

Case description

C-MAPSS dataset

- Made available by the NASA Ames Prognostics Center of Excellence.
- "FD001" sub-dataset are used, which are obtained under the same operational condition (training data 100, test data 100).
- All sensor signals, operational variables, and cycle times are used as inputs (an end-to-end solution).
- Sliding time window approach increase the amount of training data (25);
- Each sample is preprocessed by min-max normalization.
- A linear RUL function with a maximum value of 125 is utilized for each training sample.

Evaluation criteria

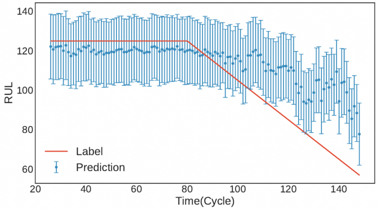
- Score: $\mathrm{SC} = \sum_{m=1}^{M} s_m$, where $s_m = \begin{cases} e^{-\frac{d_m}{13}} - 1, & \text{if } d_m < 0, \\ e^{\frac{d_m}{10}} - 1, & \text{if } d_m \geq 0. \end{cases}$
- Root mean square: $\mathrm{RMSE} = \sqrt{\frac{\sum_{m=1}^{M} (d_m)^2}{M}}$.
- Accuracy: $\mathrm{AC} = \frac{100}{M} \sum_{m=1}^{M} a_m$, where $a_m = \begin{cases} 1, & \text{if } d_m \in [-13, 10], \\ 0, & \text{if } d_m \notin [-13, 10]. \end{cases}$

Note that a smaller SC, a smaller RMSE, or a larger AC indicates a better RUL prediction performance.

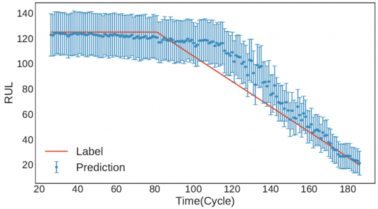
Comparison with other prognostic methods

Table 1: Comparison of point prediction with other methods.

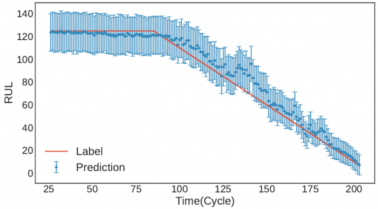
	SC	RMSE	AC
DCNN [6]	273.7	12.6	–
LSTMBS [7]	481.1	14.5	–
BDL-LSTM [5]	267.2	**12.2**	–
GA-RBM-LSTM [4]	231.0	12.6	–
DBNBP-IPF [9]	543.0	–	51%
DBN-IPF [9]	314.0	–	63%
BiLSTM-ED [11]	273.0	14.7	57%
SBI-EN [10]	**228.0**	13.6	67%
Proposed method	234.9	12.7	**70%**



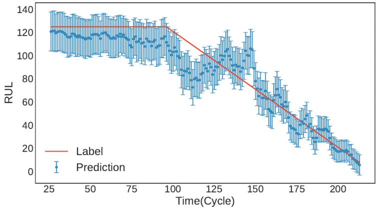
(a) Test engine unit #21



(b) Test engine unit #24



(c) Test engine unit #34



(d) Test engine unit #81

Figure 4: RUL interval estimates for four test engine units.

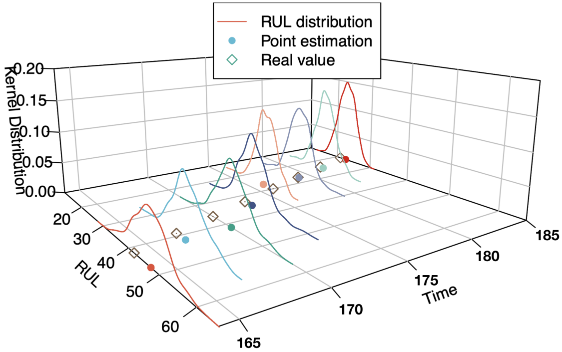
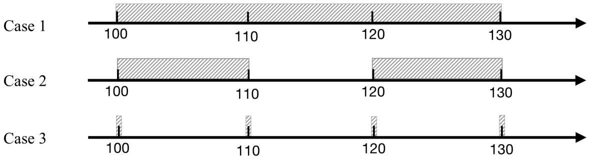


Figure 5: Uncertainty quantification for test engine unit #24.

Dynamic predictive maintenance framework

- the FD001 training set will be divided into two parts (80 engines for network training, 20 units for validation).
- $c_p = 100$, $c_c = 500$, $c_{os} = 10$, $c_f = 1$, $c_q = 0.1$, $\mathcal{L} = 20$ and $Q = 6$ [8].
- Three maintenance-window cases: (i) $\mathcal{S} = \{1, 2, \dots\}$. (ii) $\mathcal{S} = \{[10, 20], [30, 40], \dots\}$. (iii) $\mathcal{S} = \{10, 20, \dots\}$, which is a periodic inspection.



Tentative PdM scheduling with operational constraints

Table 2: Cost rates for engine unit #81.

Cycle $(200+k)$	CMF	DN-cost	R-cost	$\mathcal{A}_{200,200+k}$
200	0	0	0.736	DN
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
234	0.301	1.095	1.031	DN
235	0.442	1.316	1.328	DN
236	**0.529**	**1.609**	**1.490**	$\boldsymbol{R}$
237	0.653	1.803	1.740	R
238	0.730	2.166	1.862	R
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
247	1.000	2.452	2.452	R

Tentative time: $(\mathcal{T}_{200}^{\mathrm{m}'}, \mathcal{T}_{200}^{\mathrm{o}'}) = (236, 216), (236, 215), (230, 212)$.

Dynamic PdM updating and adjusting

Table 3: Dynamic PdM policy of unit #81 in the three cases.

Cycle	Case (i)		Case (ii)		Case (iii)	
	$\mathcal{T}_j^{\mathrm{m}'}$	$\mathcal{T}_j^{\mathrm{o}'}$	$\mathcal{T}_j^{\mathrm{m}'}$	$\mathcal{T}_j^{\mathrm{o}'}$	$\mathcal{T}_j^{\mathrm{m}'}$	$\mathcal{T}_j^{\mathrm{o}'}$
216	241	219	240	218	240	**216**
217	238	219	238	218	230	216
218	237	219	237	**218**	230	215
219	236	**219**	236	218	230	215
220	238	219	238	218	230	215
⋮	⋮	⋮	⋮	⋮	⋮	⋮
236	240	221	240	219	**240**	217
237	240	221	240	219	-	-
238	243	221	**240**	219	-	-
239	**242**	221	-	-	-	-

Optimal time :$\mathcal{T}^{\mathrm{o}^*} = 219, 218, 216$; $\mathcal{T}^{\mathrm{m}^*} = 239, 238, 236$, Actual $= 240$.

Benchmark maintenance policies

- **Classical PdM policy** (CPM): based on historical reliability data.

$$\mathcal{T}_r^{\dagger} = \bar{\mathcal{T}}^F \cdot \mathbb{1}_{\{\bar{\mathcal{T}}^F \in \mathcal{S}\}} + t_{\alpha^\dagger} \cdot \left(1 - \mathbb{1}_{\{\bar{\mathcal{T}}^F \in \mathcal{S}\}}\right),$$

  where $\bar{\mathcal{T}}^F$ is the system's mean time to failure and $\alpha^\dagger$ is the time slot at the end of the last window before $\bar{\mathcal{T}}^F$.

- **Ideal PdM policy** (IPM): based on the assumption of perfect predicted failure time.

$$\mathcal{T}_r^{\ddagger} = \mathcal{T}_r^P \cdot \mathbb{1}_{\{\mathcal{T}_r^P \in \mathcal{S}\}} + t_{\alpha^\ddagger} \cdot \left(1 - \mathbb{1}_{\{\mathcal{T}_r^P \in \mathcal{S}\}}\right),$$

  where $\alpha^\ddagger$ is the time slot at the end of the last window before $\mathcal{T}_r^P$.

- **Three state-of-the-art PdM policies** (see [8], [12], and [2]) under the periodic inspection policy.

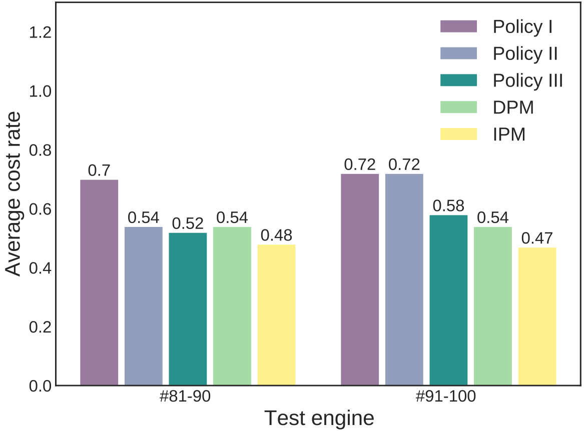
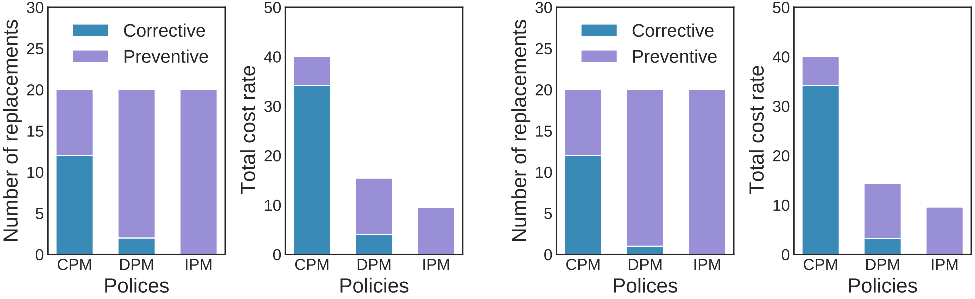
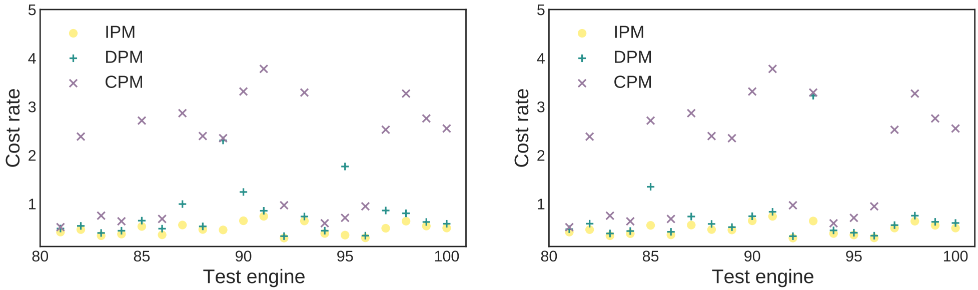


Figure 6: Average cost rates for test engines under periodic inspection.



(a) Case (i) (b) Case (ii)

Figure 7: Performance of the three policies under cases (i) and (ii).



(a) Case (i) (b) Case (ii)

Figure 8: Cost rate for each test engine under cases (i) and (ii).

目录

Prognostic driven PdM framework

- **Prognostic stage**, we propose a BDL-based framework to qualify aleatoric and epistemic uncertainties, and output a predictive distribution of RULs.
- **Maintenance decision-making stage**:
  - a practical policy in general inspection scenarios is presented. This model enables rapid evaluation of the cost rates of R- and DN-option at any moment, and produces tentative PdM scheduling with operational constraints.
  - As more CM data are progressively collected, our framework dynamically **updates** and **adjusts maintenance** and **spare-part ordering** decisions to generate a more reliable PdM scheduling.

[1] C. Chen, N. Lu, B. Jiang, and C. Wang, *A risk-averse remaining useful life estimation for predictive maintenance*, IEEE/CAA Journal of Automatica Sinica **8** (2021January), no. 2, 412–422.

[2] C. Chen, Z. H. Zhu, J. Shi, N. Lu, and B. Jiang, *Dynamic predictive maintenance scheduling using deep learning ensemble for system health prognostics*, IEEE Sensors Journal **21** (2021October), no. 23, 26878–26891.

[3] I. de Pater, A. Reijns, and M. Mitici, *Alarm-based predictive maintenance scheduling for aircraft engines with imperfect remaining useful life prognostics*, Reliability Engineering & System Safety **221** (2022May), 108341.

[4] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, *Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture*, Reliability Engineering & System Safety **183** (2019March), 240–251.

[5] M. Kim and K. Liu, *A Bayesian deep learning framework for interval estimation of remaining useful life in complex systems by incorporating general degradation characteristics*, IISE Transactions **53** (2020June), no. 3, 326–340.

[6] X. Li, Q. Ding, and J.-Q. Sun, *Remaining useful life estimation in prognostics using deep convolution neural networks*, Reliability Engineering & System Safety **172** (2018April), 1–11.

[7] Y. Liao, L. Zhang, and C. Liu, *Uncertainty prediction of remaining useful life using long short-term memory network based on bootstrap method*, 2018 ieee international conference on prognostics and health management, 2018June, pp. 1–8.

[8] K. T. Nguyen and K. Medjaher, *A new dynamic predictive maintenance framework using deep learning for failure prognostics*, Reliability Engineering & System Safety **188** (2019August), 251–262.

[9] K. Peng, R. Jiao, J. Dong, and Y. Pi, *A deep belief network based health indicator construction and remaining useful life prediction using improved particle filter*, Neurocomputing **361** (2019October), 19–28.

[10] W. Yu and Kim, *An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme*, Reliability Engineering & System Safety **199** (2020July), 106926.

[11] W. Yu, I. Y. Kim, and C. Mechefske, *Remaining useful life estimation using a bidirectional recurrent neural network based autoencoder scheme*, Mechanical Systems and Signal Processing **129** (2019August), 764–780.

[12] L. Zhang and J. Zhang, *A data-driven maintenance framework under imperfect inspections for deteriorating systems using multitask learning-based status prognostics*, IEEE Access **9** (2020December), 3616–3629.

Thanks!