Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio

doi: 10.3850/978-981-18-8071-1_P174-cd

Predictive maintenance of multi-component aircraft system using convolutional neural networks and deep reinforcement learning

Juseong Lee

Department of Industrial Engineering & Innovation Sciences, Eindhoven University of Technology, The Netherlands. E-mail: j.lee@tue.nl

Mihaela Mitici

Faculty of Science, Utrecht University, The Netherlands. E-mail: M.A.Mitici@uu.nl

Predictive maintenance is a new approach to replacing components based on the data-driven Remaining-Useful-Life (RUL) prognostics. However, implementing predictive maintenance remains challenging for aircraft. First, as aircraft maintenance requires high reliability, it is necessary to quantify the uncertainty of the predicted RUL. Moreover, the maintenance of multi-component systems should be planned considering the updated RUL distributions of individual components and complex cost models. This paper proposes an integrated method for the predictive maintenance of multi-component aircraft systems. We estimate the probability distribution of RUL using convolutional neural networks and Monte Carlo dropouts. Then, deep reinforcement learning (DRL) is applied to plan the replacement of multiple components based on individual RUL distributions. This method considers the uncertainty of RUL predictions, risk of component failure, time-varying maintenance costs, and maintenance slot costs. A case study on the predictive replacement of two turbofan engines illustrates the proposed method. By considering the probability distribution of RUL and grouping some replacements, the proposed DRL-based predictive maintenance provides lowered long-term maintenance cost.

Keywords: aircraft maintenance, predictive maintenance, data-driven maintenance, Remaining-Useful-Life prognostics, deep reinforcement learning.

1. Introduction

Predictive maintenance (PdM) has evolved based on rapidly developing artificial intelligence and increasing use of health condition monitoring data. PdM can improve the efficiency of traditional time-based maintenance (TBM), which relies on maintenance tasks repeated at fixed time intervals. These intervals are often shorter than the average life of components to prevent unexpected failures. Thus, some components are replaced far before their end-of-life, wasting their useful life and incurring a high cost. Under PdM, however, individual components are replaced based on their Remaining-Useful-Life (RUL) estimated from the up-to-date condition monitoring data. Using the RUL predictions, PdM aims to plan maintenance tasks more efficiently.

During the last decade, many studies proposed RUL prognostics methods for aircraft components and systems (Sprong et al., 2020). For example, a stochastic regression model is used to predict RUL of landing gear brakes (Lee and Mitici, 2020), particle filtering is used to estimate RUL of aircraft cooling units (Mitici and Pater, 2021), etc. Recent studies consider more complex aircraft systems, such as turbofan engines. A popular approach for turbofan engine RUL prognostics is convolutional neural networks (CNN) (Li et al., 2018; Pater et al., 2022), and their variations, such as multi-scale deep CNN (Li et al., 2020), and CNN with temporal pooling (Babu et al., 2016). However, most existing methods are limited to single-value predictions of RUL without explicit quantification of uncertainty, which is necessary to plan safety-critical aircraft maintenance.

Moreover, there are not enough studies on maintenance planning methods considering the data-driven RUL prognostics of multiple components. In (Pater et al., 2022), turbofan engines are replaced based on the alarm triggered by RUL predictions. Similarly, in (Wang et al., 2018), airframe panels are replaced based on the crack size estimated by the extended Kalman filter. These methods, however, rely only on the mean prediction of RUL or crack size without considering the associated uncertainty. In addition, the economic correlation in multi-component system maintenance needs to be exploited more. In (Lee et al., 2022), a more complex cost model considering hangar usage cost, is suggested for the maintenance grouping of aircraft landing gear brakes. However, this method is based on the single-value predictions of RUL.

Overall, implementing predictive maintenance for a multi-component aircraft system has a few remaining challenges: (1) to quantify the uncertainty of RUL predictions and (2) to plan multi-component maintenance considering the predicted RUL, its uncertainty information, and complex cost models.

In this paper, we propose a PdM method for a multi-component aircraft system, which integrates probabilistic RUL prognostics and sequential maintenance planning. Our probabilistic RUL prognostics use CNN and Monte Carlo dropout to estimate the probability distribution of RUL prediction. Then, we apply deep reinforcement learning (DRL) for the sequential planning of predictive maintenance tasks. This method considers time-varying maintenance costs, the risk of component failures, and, most importantly, the uncertainty of RUL predictions.

2. Probabilistic RUL Prognostics

2.1. *Multi-channel 1D CNN model*

We consider health condition data consisting of time-series data of multiple sensors monitoring a component. Let the health condition data x have n_F features corresponding to a time-series data of sensor reading, and let the length of the time-series be n_W, i.e.,

$$x = \begin{bmatrix} x_{1,1} & \dots & x_{n_F,1} \\ \vdots & & \vdots \\ x_{1,n_W} & \dots & x_{n_F,n_W} \end{bmatrix}. \tag{1}$$

We use run-to-failure data for training. Thus, the true Remaining-Useful-Life (RUL) ρ is calculated from the remaining time to failure.

Given this setup, the RUL prognostics is posed as a supervised learning problem to predict RUL ρ (output) from a given health condition data x (input). We propose a convolutional neural network (CNN) model, where multiple 1D kernels are used for multiple channels (Lee and Mitici, 2023). The proposed architecture of CNN is given in Fig. 1.

The input data x is a 2D matrix of size $(n_F \times n_W)$, of which columns are time-series data of n_F sensors. We consider the time-series data of each sensor as an individual channel of CNN, and apply a 1D kernel along each. This approach effectively extracts patterns from multivariate time-series data (Zheng et al., 2014).

We first apply 5 multi-channel 1D convolutional layers to the input data. Each convolutional layer is defined by the kernel length n_K, and the number of output channels n_C. The l^{th} convolutional layer gets input $x^{(l-1)}$ from $(l-1)^{\text{th}}$ layer, where $x^{(l-1)}$ has $n_C^{(l-1)}$ channels. The output of channel $c \in \{1, \dots, n_C^l\}$ of the l^{th} convolutional layer is obtained as follows:

$$x_c^l = g^l \left(b_c^l + \sum_{c'=1}^{n_C^{(l-1)}} \kappa_{c,c'}^l * x_{c'}^{(l-1)} \right) \tag{2}$$

where $*$ is the convolutional operator, $\kappa_{c,c'}^l$ is the kernel for input channel c' and output channel c, n_C^l is the number of output channels in l^{th} layer, b_c^l is the bias of output channel c, and $g^l(\cdot)$ is rectified linear unit (ReLU) activation function.

After the convolutional layers, two linear layers are applied. The number of neurons in l^{th} layer is denoted as n_N^l. Finally, the output neuron represents the RUL of the given input data.

2.2. *Monte Carlo dropout for uncertainty quantification*

We apply Monte Carlo dropout for the convolutional layers and linear layers in Fig. 1. Applying Monte Carlo dropout, random nodes are ignored during the forward propagation of the neural network. In effect, slightly different neural networks are trained simultaneously, preventing the overfitting of the model during training (Srivastava et al., 2014).

During testing, Monte Carlo dropout can be

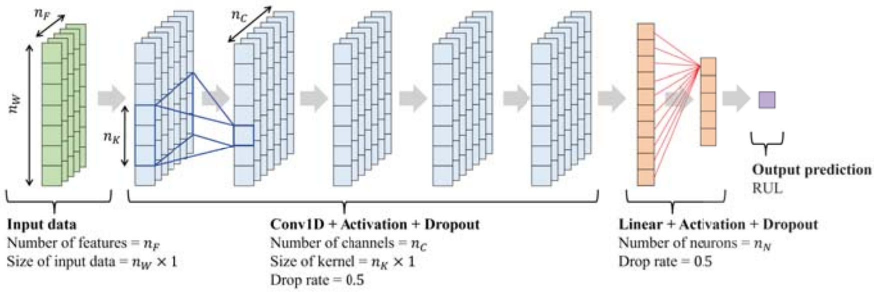

Fig. 1. Multi-channel 1D CNN architecture.

used to quantify the uncertainty of output prediction during testing (Gal and Ghahramani, 2015). For a given test input data, we generate an empirical distribution of output using multiple forward propagations with different neurons dropped out. Mathematically, this distribution approximates the posterior distribution of deep Gaussian process (Gal and Ghahramani, 2015). Thus, we propose to approximate the distribution of the predicted RUL using the distribution of Monte Carlo dropout.

3. Predictive Maintenance of Multi-component Aircraft System

3.1. *Formulating PdM as Reinforcement Learning Problem*

Predictive maintenance (PdM) is a sequential decision-making process (see Fig. 2). A predictive maintenance schedule is updated every D flight cycles, which is referred as a decision step. At the start of t^{th} decision step, we collect sensor data available from the previous decision step. Using the updated sensor data of each component $i \in \{1, \ldots, N\}$, we estimate the distribution of Remaining-Useful-Life (RUL). Given the updated RUL prognostics, we decide when to replace the components during the t^{th} decision step. We can either schedule a component replacement at cycle $k \in \{1, ..., D\}$, or do not replace it in the t^{th} decision step. For the next decision step $(t + 1)$, we repeat the same process.

We formulate this sequential decision-making of predictive maintenance into a reinforcement learning problem (Kaelbling et al., 1996). An *agent* interacts with an *environment*. The agent *observes* the *state* of the environment, and takes

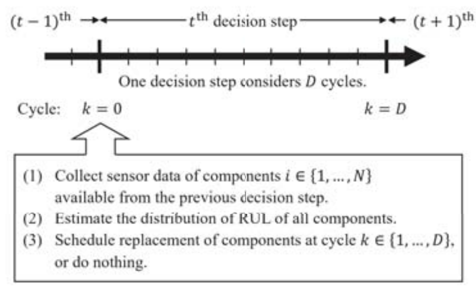

Fig. 2. A sequential decision-making for predictive aircraft maintenance of a multi-component system.

an *action* to change the state of the environment. During this state transition, the agent gets a *reward* that reinforces its specific behavior, namely *policy*.

In Fig. 3, the predictive maintenance is illustrated as a reinforcement learning problem. At decision step t, the (hidden) state is $\rho_t = [\rho_{1,t}, \cdots, \rho_{N,t}]$, where $\rho_{i,t}$ is the true RUL of component i. The maintenance agent cannot observe the true RUL ρ_t. Instead, it observes the estimated distributions of RUL of the components, which are obtained using the RUL prognostics model and sensor data (see Section 2). The observed state s_t is structured as follows:

$$s_t = \begin{bmatrix} p_{1,1,t} & \cdots & p_{1,D,t} \\ \vdots & & \vdots \\ p_{N,1,t} & \cdots & p_{N,D,t} \end{bmatrix}, \qquad (3)$$

where D is the number of flight cycles in a decision step, N is the number of components, and $p_{i,k,t}$ is the probability that the RUL of component i is less than or equal to k, i.e., cumulative distri-

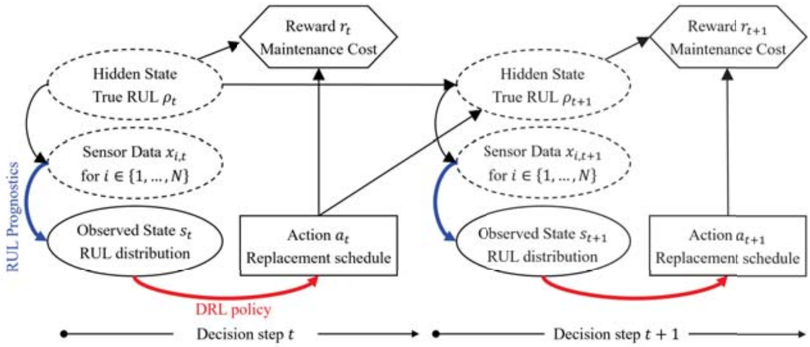

Fig. 3. Formulation of predictive maintenance into a reinforcement learning problem.

bution function (CDF). Formally, $p_{i,k,t}$ is defined as follows:

$$p_{i,k,t} = P(R_{i,t} \leq k \mid x_{i,t}), \tag{4}$$

where $R_{i,t}$ is the predicted RUL of component i at decision step t, and $x_{i,t}$ is the health condition monitoring data of component i at decision step t.

Based on the observed state s_t, the maintenance agent takes action a_t defined as follows:

$$a_t = \begin{bmatrix} a_{1,t} & \cdots & a_{N,t} \end{bmatrix}, \tag{5}$$

where $a_{i,t} > 0$ implies the maintenance decision for component i. If $\lceil a_{i,t} \rceil \in \{1, 2, ..., D\}$, then we schedule a maintenance at flight cycle $\lceil a_{i,t} \rceil$. If $\lceil a_{i,t} \rceil > D$, then we do nothing in the t^{th} decision step, i.e., no maintenance.

A reward of a decision step r_t is the maintenance cost for the upcoming D flight cycles, which consists of maintenance task cost ($r_{i,t}^{\text{task}}$) of component i, and maintenance slot cost ($r_{k,t}^{\text{slot}}$) at flight cycle k.

$$r_t = \sum_{i \in \{1,...,N\}} r_{i,t}^{\text{task}} + \sum_{k \in \{1,...,D\}} r_{k,t}^{\text{slot}}. \tag{6}$$

The maintenance task cost $r_{i,t}^{\text{task}}$ depends on maintenance action $a_{i,t}$ and true RUL $\rho_{i,t}$.

$$r_{i,t}^{\text{task}} = \begin{cases} -c_{\text{sch}}(\lceil a_{i,t} \rceil) & \text{if } \lceil a_{i,t} \rceil \leq D \wedge \lceil a_{i,t} \rceil < \rho_{i,t} \\ -c_{\text{uns}} & \text{if } \lceil a_{i,t} \rceil \leq D \wedge \lceil a_{i,t} \rceil \geq \rho_{i,t} \\ -c_{\text{uns}} & \text{if } \lceil a_{i,t} \rceil > D \wedge \rho_{i,t} \leq D \\ 0 & \text{if } \lceil a_{i,t} \rceil > D \wedge \rho_{i,t} > D \end{cases} \tag{7}$$

If the maintenance is scheduled before the true RUL $\rho_{i,t}$, we pay a scheduled replacement cost $c_{\text{sch}}(k)$ defined as $c_{\text{sch}}(k) = c_0 - c_1 k$. The cost of replacement scheduled in early flight cycles is assumed to be expensive due to the shorter time to prepare spare parts ($c_1 > 0$). If the maintenance is not scheduled before the true RUL $\rho_{i,t}$, the component fails, and an expensive penalty cost is unavoidable ($c_{\text{uns}} > c_0$).

The maintenance slot cost $r_{k,t}^{\text{slot}}$ is paid for each flight cycle when the aircraft undergoes maintenance.

$$r_{k,t}^{\text{slot}} = \begin{cases} -c_{\text{slot}} & \text{if } \exists i \in \{1, \ldots, N\} \\ & \quad \rho_{i,t} = k \vee \lceil a_{i,t} \rceil = k \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

This includes a fixed cost for aircraft maintenance, including hangar usage, lease of alternative aircraft for scheduled flights, etc. Unlike the task cost, it is paid once per maintenance slot regardless of the number of tasks. Thus, we may group the maintenance of several components in a slot to save the maintenance slot cost. Here, the key trade-off lies between the saved maintenance slot cost and the wasted useful life of the component replaced early due to the grouping. The optimal maintenance action a_t should be made to minimize the overall cost r_t based on the observed state s_t, which is the CDF of RUL of multiple components.

After the agent gets reward r_t by taking action a_t, the hidden state true RUL ρ_{t+1} is updated. If component i is not replaced, $\rho_{i,(t+1)} = \rho_{i,t} - D$. If component i is replaced by either scheduled or unscheduled maintenance, $\rho_{i,(t+1)}$ is updated for the true RUL of a new component.

The goal of the maintenance agent is to find an optimal policy $\pi(a_t|s_t) : \mathcal{S} \rightarrow \mathcal{A}$, where $s_t \in \mathcal{S}$ is the state space and $a_t \in \mathcal{A}$ is the action space. The optimal policy π^* is defined as a policy that maximizes the expected reward, i.e.,

$$J(\pi) = \sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[\gamma^t r_t(s_t, a_t)], \quad (9)$$

where γ is a discount factor, and $\rho_\pi(s_t, a_t)$ is the state-action trajectory distribution induced by a policy π.

3.2. *Soft-Actor-Critic algorithm*

We train the maintenance agent using the Soft-Actor-Critic (SAC) algorithm (Haarnoja et al., 2018), which has proven to be effective for single-component predictive maintenance (Lee and Mitici, 2023).

The SAC algorithm trains a policy (actor) and a state-action value (critic). Unlike typical actor-critic algorithms, the SAC algorithm uses a stochastic policy and a *soft* objective to explicitly optimize exploration towards new policies.

The stochastic policy $\pi_\phi(a_t|s_t)$ is defined by the mean $f^\mu_\phi(s_t)$ and the standard deviation $f^\sigma_\phi(s_t)$ of an action, where ϕ is the trainable parameters of f^μ_ϕ and f^σ_ϕ. During the training, action a_t is sampled from this stochastic policy as follows:

$$a_t = f^\mu_\phi(s_t) + \epsilon_t \cdot f^\sigma_\phi(s_t), \quad (10)$$

where ϵ_t is sampled from a standard Gaussian distribution. During the evaluation, the deterministic policy is used, i.e., $a_t = f^\mu_\phi(s_t)$.

The soft objective $J(\pi)$ considers the maximization of both the expected reward and the entropy of the stochastic policy. Formally,

$$J(\pi) = \textstyle\sum_t \underset{(s_t,a_t)\sim\rho_\pi}{\mathbb{E}} \gamma^t[r_t(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t))], \quad (11)$$

where α is the temperature parameter determining the weight between the reward and the entropy.

For the SAC algorithm, we use three deep neural network models: the policy π_ϕ, the soft Q function Q_θ, and the soft value function V_ψ, where ϕ, θ, and ψ are the trainable parameters. These models are trained with the following loss functions:

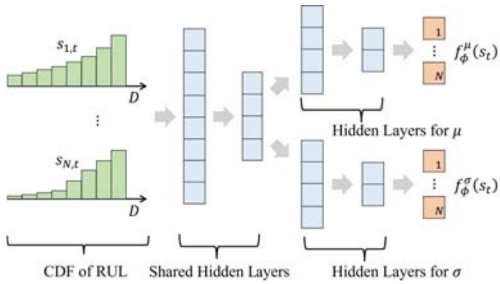

Fig. 4. Policy network π_ϕ for multi-component aircraft systems.

$$J_\pi(\phi) = \underset{\substack{s_t\sim\mathcal{D},\\ a_t\sim\pi_\phi}}{\mathbb{E}} \left[\log \pi_\phi(a_t|s_t) - \tfrac{1}{\alpha} Q_\theta(s_t, a_t)\right], \quad (12)$$

where $\mathcal{D}$ is the replay buffer. The architecture of policy net π_ϕ for multi-component aircraft system is proposed in Fig. 4. Its input is the CDF of RUL of individual components (matrix of size $N \times D$), and its output is the mean and standard deviation of the stochastic policy (vector of length N).

$$J_V(\psi) = \underset{s_t\sim\mathcal{D}}{\mathbb{E}} \left[\frac{1}{2}(V_\psi(s_t) - \hat{V}(s_t))^2\right], \quad (13)$$

where $\hat{V}(s_t) = [Q_\theta(s_t, a_t) - \alpha \log \pi_\phi(a_t|s_t)]$.

$$J_Q(\theta) = \underset{(s_t,a_t)\sim\mathcal{D}}{\mathbb{E}} \left[\tfrac{1}{2}\left(Q_\theta(s_t, a_t) - \hat{Q}(s_t, a_t)\right)\right], \quad (14)$$

where $\hat{Q}(s_t, a_t) = r_t(s_t, a_t) + \gamma \underset{s_{t+1}\sim p}{\mathbb{E}}[V_{\bar{\psi}}(s_{t+1})]$. The full implementation of the SAC algorithm for PdM is elaborated in (Lee and Mitici, 2023).

4. Case Study: Predictive Replacements of Two Turbofan Engines

The proposed method using probabilistic RUL prognostics and DRL approach is illustrated for the predictive replacements of two turbofan engines.

4.1. *Data description*

We use the C-MAPSS data set, especially FD002 subset, which is a simulated health condition data of aircraft turbofan engines considering 6 operating conditions and 1 failure mode (Saxena and Goebel, 2008). We use 14 non-constant sensor measurements, the current operating condition,

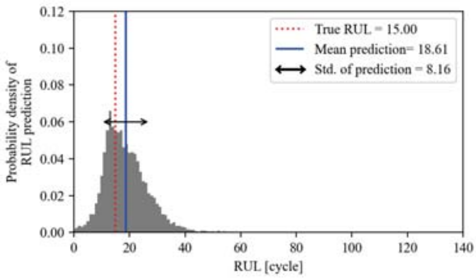

Fig. 5. Estimated RUL probability distribution of FD002 Testing Engine 158.

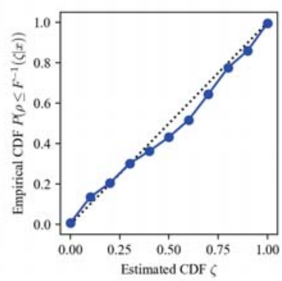

Fig. 6. Calibration plot of the estimated CDF of RUL for FD002 data subset.

and the number of flight cycles at 6 operating conditions, i.e., 21 features in total ($n_F = 21$). The time window is 21 ($n_W = 21$).

The FD002 data subset has run-to-failure data for 260 engines. We use the data from 130 engines to train the CNN model for probabilistic RUL prognostics. The rest 130 engines are used to generate maintenance episodes for the DRL approach (100 engines for the training and 30 engines for the evaluation).

4.2. *Estimating distribution of RUL*

The trained CNN model is tested for the testing engines of the FD002 data subset, which are independent of the engines used for training. An example of the estimated RUL probability distribution is shown in Fig. 5. The mean of the estimated RUL distribution is 18.61 cycles, while the true RUL is 15 cycles, i.e., the mean value overestimates the RUL. However, the probability distribution in Fig. 5 gives more information than just a mean value. The standard deviation of the predicted RUL (8.16) is relatively large. The peak probability is at 13 cycles. The distribution is right-skewed. As such, using the probability distribution, better maintenance decisions can be made.

The quality of the estimated RUL distribution is analyzed using calibration plot (Kuleshov et al., 2018). Let $F(R|x)$ be the cumulative distribution function (CDF) of the estimated RUL R, given input data x. Let $F^{-1}(\zeta|x) = \inf\{R : \zeta \leq F(R|x)\}$, i.e., the quantile function of R. The estimated RUL distribution is perfectly calibrated if

$$P(\rho \leq F^{-1}(\zeta|x)) = \zeta \quad \forall \zeta \in [0,1], \tag{15}$$

where ρ is the true RUL. The calibration plot $P(\rho \leq F^{-1}(\zeta|x))$ against ζ in Fig. 6 is linear, and we conclude that the estimated probability distribution is well calibrated to the empirical probability distribution of RUL. The current approach does not distinguish aleatoric uncertainty and epistemic uncertainty, but these can be distinguished using the method proposed in (Valdenegro-Toro and Mori, 2022), which will provide further uncertainty information for future work.

4.3. *Planning predictive maintenance*

We train the DRL agent using the SAC algorithm. At each training episode, health condition data and true RUL are sampled from the C-MAPSS data set, our CNN model generates an observed state s_t (RUL distribution), and a reward r_t is given based on the true RUL and the action a_t of the DRL agent. The reward model parameters are assumed as follows: $c_0 = 1.0$, $c_1 = 0.01$, $c_{\text{uns}} = 3.0$, and $c_{\text{slot}} = 0.5$.

For the evaluation, a deterministic policy of the trained DRL agent is used, i.e., $a_t = f^{\mu}_{\phi}(s_t)$. If the predicted RUL of engine i is large, the trained policy suggests no maintenance, i.e., $\lceil a_{i,t} \rceil > D$. Otherwise, it suggests scheduling a replacement after $\lceil a_{i,t} \rceil$ flight cycles. In Fig. 7, only engine 2 is scheduled for a replacement because the CDF of RUL of engine 1 is low, $p_{1,30,t} < 0.003$.

If the RUL of two engines are close enough, the DRL policy may suggest grouping two replace-

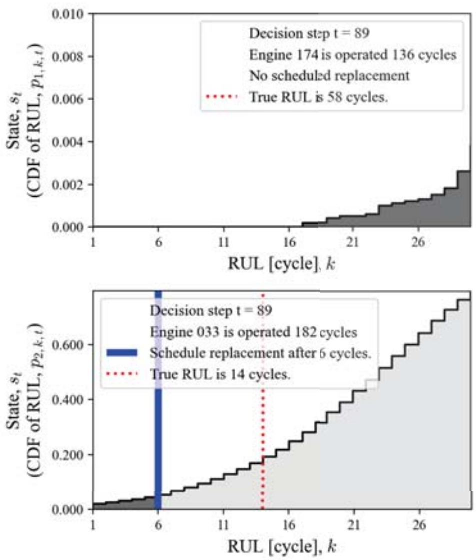

Fig. 7. The DRL agent scheduled a replacement for only one engine. Maintenance of another engine will be scheduled in the later decision steps.

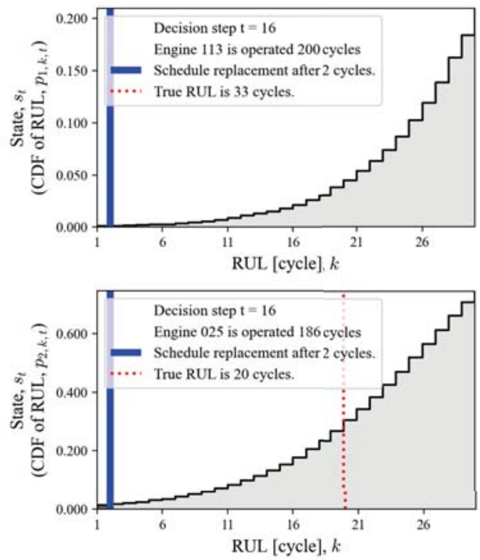

Fig. 8. The DRL agent scheduled replacements for both engines after 2 cycle. This reduces the maintenance slot cost.

ments at the same cycle to save the maintenance slot cost. In Fig. 8, two replacements are scheduled together after 2 flight cycle.

4.4. *Comparison of performance*

The long-term maintenance cost of the DRL-based PdM is compared to other maintenance methods. One is time-based maintenance (TBM), where engines are replaced at the age of A cycles. The age threshold A can be optimized to avoid unscheduled replacements ($A = 125$) or minimize cost ($A = 175$). We also consider the PdM without using DRL, where engines are replaced when the CDF of RUL exceeds C. The CDF threshold C is optimized to minimize cost ($T = 0.01$).

Table 1 shows that the PdM methods significantly reduce the total maintenance cost compared to the TBM methods. Compared to the cost-minimal TBM ($A = 175$), the DRL-based PdM reduces 26% of the total cost. Using the RUL prognostics, instead of a fixed interval of replacements, engines are timely replaced when their RUL is small.

Moreover, the DRL approach further reduces 27% of unscheduled maintenance cost compared to the PdM using the fixed CDF threshold ($C = 0.01$). Setting a fixed CDF threshold is not effective because the distributions of RUL vary over different engines (see Fig. 7 –8). Thus, it is beneficial to train the maintenance agent based on the probability distributions.

The DRL policy reduces the maintenance slot cost by grouping engine replacements when it is beneficial, i.e., when the cost saving from grouping is higher than the cost saving from using another engine for a few more cycles. Overall, under the DRL policy, 5% of engine replacements are grouped. On the other hand, under TBM, only 1% of replacements are grouped by chance. As a result, Table 1 shows that the DRL-based PdM uses the smallest maintenance slot cost.

5. Conclusion

In this study, we propose a predictive maintenance method for multi-component aircraft systems, considering both Remaining-Useful-Life (RUL) prognostics and maintenance planning. Especially our RUL prognostics method predicts the probability distribution of RUL using convolutional neural networks and Monte Carlo dropouts. The

Table 1. Comparison of engine maintenance cost for 3000 flight cycles.

	Maintenance Cost			
Maintenance Method	Total	Sch.	Uns.	Slot
PdM (DRL)	56.6	37.8	3.2	15.6
PdM ($C = 0.01$)	58.2	37.7	4.3	16.2
TBM ($A = 175$)	76.7	32.7	25.6	18.4
TBM ($A = 125$)	86.5	60.4	0.0	26.1

RUL distribution is used by a deep reinforcement learning (DRL) agent to plan cost-minimal maintenance. The benefit of the proposed method is illustrated for the predictive replacement of two turbofan engines. Finally, this study illustrates the potential of the DRL approach as a predictive maintenance planning method.

Future work is needed in the disentanglement of aleatoric and epistemic uncertainty, and the application of the framework in other systems and maintenance actions.

References

Babu, G. S., P. Zhao, and X.-L. Li (2016). Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life. In *DASFAA 2016: Database Systems for Advanced Applications*, pp. 214–228.

Gal, Y. and Z. Ghahramani (2015). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *arXiv*.

Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *arXiv*.

Kaelbling, L. P., M. L. Littman, and A. W. Moore (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research* (4), 237–285.

Kuleshov, V., N. Fenner, and S. Ermon (2018). Accurate Uncertainties for Deep Learning Using Calibrated Regression. *arXiv*.

Lee, J. and M. Mitici (2020). An integrated assessment of safety and efficiency of aircraft maintenance strategies using agent-based modelling and stochastic Petri nets. *Reliability Engineering & System Safety 202*, 107052.

Lee, J. and M. Mitici (2023). Deep reinforcement learning for predictive aircraft maintenance using probabilistic Remaining-Useful-Life prognostics. *Reliability Engineering & System Safety 230*, 108908.

Lee, J., I. d. Pater, S. Boekweit, and M. Mitici (2022). Remaining-Useful-Life prognostics for opportunistic grouping of maintenance of landing gear brakes for a fleet of aircraft. In *PHM Society European Conference*, Volume 7, pp. 278–285.

Li, H., Z. Wan, and H. He (2020). Constrained EV Charging Scheduling Based on Safe Deep Reinforcement Learning. *IEEE Transactions on Smart Grid 11*(3), 2427–2439.

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

Mitici, M. and I. d. Pater (2021). Online Model-Based Remaining-Useful-Life Prognostics for Aircraft Cooling Units Using Time-Warping Degradation Clustering. *Aerospace 8*(6), 168.

Pater, I. d., A. Reijns, and M. Mitici (2022). Alarm-based predictive maintenance scheduling for aircraft engines with imperfect Remaining Useful Life prognostics. *Reliability Engineering & System Safety 221*, 108341.

Saxena, A. and K. Goebel (2008). Turbofan Engine Degradation Simulation Data Set.

Sprong, J. P., X. Jiang, and H. Polinder (2020). Deployment of Prognostics to Optimize Aircraft Maintenance – A Literature Review. *Journal of International Business Research and Marketing 5*(4), 26–37.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout A simple Way to Prevent Neural networks from Overfitting. *Journal of Machine Learning Research 15*, 1929–1958.

Valdenegro-Toro, M. and D. S. Mori (2022). A Deeper Look into Aleatoric and Epistemic Uncertainty Disentanglement. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) 00*, 1508–1516.

Wang, Y., C. Gogu, N. Binaud, C. Bes, R. T. Haftka, and N.-H. Kim (2018). Predictive airframe maintenance strategies using model-based prognostics. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability 232*(6), 690–709.

Zheng, Y., Q. Liu, E. Chen, Y. Ge, and J. L. Zhao (2014). Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks. In *International Conference on Web-Age Information Management*, Volume 10, pp. 96–112.