Alarm-based predictive maintenance scheduling for aircraft engines with imperfect Remaining Useful Life prognostics

de Pater, Ingeborg; Reijns, Arthur; Mitici, Mihaela

DOI
10.1016/j.ress.2022.108341

Publication date
2022

Document Version
Final published version

Published in
Reliability Engineering and System Safety

Citation (APA)
de Pater, I., Reijns, A., & Mitici, M. (2022). Alarm-based predictive maintenance scheduling for aircraft engines with imperfect Remaining Useful Life prognostics. *Reliability Engineering and System Safety*, *221*, Article 108341. https://doi.org/10.1016/j.ress.2022.108341

Reliability Engineering and System Safety 221 (2022) 108341

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Alarm-based predictive maintenance scheduling for aircraft engines with imperfect Remaining Useful Life prognostics

Ingeborg de Pater *, Arthur Reijns, Mihaela Mitici

Faculty of Aerospace Engineering, Delft University of Technology, HS 2926 Delft, The Netherlands

ARTICLE INFO

Keywords:
Predictive maintenance planning
RUL prognostics
Aircraft maintenance
Turbofan engines
Fleet of aircraft

ABSTRACT

The increasing availability of condition monitoring data for aircraft components has incentivized the development of Remaining Useful Life (RUL) prognostics in the past years. However, only few studies consider the integration of such prognostics into maintenance planning. In this paper we propose a dynamic, predictive maintenance scheduling framework for a fleet of aircraft taking into account imperfect RUL prognostics. These prognostics are periodically updated. Based on the evolution of the prognostics over time, alarms are triggered. The scheduling of maintenance tasks is initiated only after these alarms are triggered. Alarms ensure that maintenance tasks are not rescheduled multiple times. A maintenance task is scheduled using a safety factor, to account for potential errors in the RUL prognostics and thus avoid component failures. We illustrate our approach for a fleet of 20 aircraft, each equipped with 2 turbofan engines. A Convolution Neural Network is proposed to obtain RUL prognostics. An integer linear program is used to schedule aircraft for maintenance. With our alarm-based maintenance framework, the costs with engine failures account for only 7.4% of the total maintenance costs. In general, we provide a roadmap to integrate imperfect RUL prognostics into the maintenance planning of a fleet of vehicles.

1. Introduction

The cost of aircraft maintenance is estimated to be 10.3% of the total airline operating costs, with approximately 3.3 million dollars spent on maintenance per aircraft in 2019 [1]. Striving to reduce these costs, aircraft maintenance is shifting to data-driven, predictive maintenance where on-board sensors are increasingly used to monitor the health condition of the aircraft components. Based on these sensor measurements, dedicated algorithms are developed to estimate the Remaining Useful Life (RUL) of components. Using RUL prognostics, the aim is to anticipate failures and optimize the deployment of maintenance tasks. One of the main challenges in predictive maintenance is to obtain reliable RUL prognostics and to integrate them into maintenance planning [2].

Most existing studies focus solely on developing RUL prognostics, using either a model-based or a machine learning approach [3]. Model-based RUL prognostics assume that the degradation of components is characterized by a stochastic process. For instance, in [4,5] the RUL of aircraft Cooling Units is estimated using particle filtering with an exponential degradation model and a Wiener linear process, respectively. In [6], RUL prognostics for aircraft landing gear brakes are obtained using a linear regression, while a Gamma process characterizes the degradation of the brakes. Machine learning algorithms have been proposed to estimate the RUL of, for instance, aircraft turbofan engines [7–9] and bearings [10,11]. In [7–9], a Convolutional neural network (CNN) is used to predict the RUL of turbofan engines. To predict the RUL of rolling element bearings, a CNN with a residual is proposed in [10], while a CNN with multi-scale feature extraction is proposed in [11]. We refer to [3,12] for an extensive overview of recent studies about RUL prognostics.

Several studies focus mainly on condition-based or predictive maintenance planning, where the RUL prognostics are based on simple, generic probability distributions. In [13,14], maintenance is planned for a railway network and a steel bridge structure respectively, using a Markov Decision Process. In [15], a Large Neighborhood Search algorithm is proposed for the maintenance planning of a fleet of aircraft.

Few studies develop RUL prognostics and subsequently integrate these prognostics into maintenance planning [2,16]. Even so, these studies focus on maintenance planning for one (multi-component) system. In [16], multi-class RUL prognostics for aircraft turbofan engines are generated using a Long Short-Term Memory neural network. Based on these prognostics, engine replacements are planned and spare parts are ordered. In [17], prognostics for aircraft airframe cracks are developed using an extended Kalman filter. These prognostics are further used to determine which panels of a single aircraft are maintained, if

* Corresponding author.
E-mail address: i.i.depater@tudelft.nl (I. de Pater).

https://doi.org/10.1016/j.ress.2022.108341
Received 15 October 2021; Received in revised form 12 January 2022; Accepted 13 January 2022
Available online 29 January 2022

any. In contrast to these studies, we consider the maintenance of a *fleet* of aircraft, integrating data-driven RUL prognostics.

Even fewer studies develop RUL prognostics and subsequently integrate these prognostics in the maintenance planning for multiple assets/systems. In [18], a particle filtering algorithm is used to determine RUL prognostics for aircraft cooling units. With these prognostics, maintenance for a fleet of aircraft is planned using linear programming, taking into account the availability of spare parts. In [19], the maintenance of multiple aircraft brakes is considered. An aircraft brake is replaced as soon as the predicted RUL falls below a threshold. Multiple objectives, such as minimizing flight delays, minimizing the number of unscheduled maintenance tasks and minimizing the total number of maintenance tasks, are considered. In [20], maintenance is planned for a fleet of vehicles using a multi-objective genetic algorithm. The aim is to minimize the total maintenance costs, the total workload, the expected number of failures and the changes in the maintenance schedule.

In general, these studies show that integrating RUL prognostics into maintenance planning models leads to lower maintenance costs and a more efficient use of spare parts [16,18]. However, when planning maintenance, the errors (e.g., RMSE, MAE, false negatives, false positives) of the RUL prognostics are not considered. To account for such potential errors when planning maintenance, we propose a system of alarms to initiate maintenance task scheduling, together with a safety margin that adjusts the moment when maintenance tasks are scheduled.

This paper proposes a dynamic, predictive maintenance planning framework for a fleet of aircraft that integrates machine-learning RUL prognostics for aircraft components. These prognostics are periodically updated as more measurements become available. Alarms are triggered based on the evolution of the prognostics over time. Triggering an alarm for a component initiates the scheduling of a maintenance task for this component. The ideal time to schedule such a task is determined based on the RUL prognostics and a safety margin, to account for potential errors in the RUL prognostics. Once a maintenance task for a component is scheduled, we continue to periodically update the RUL prognostics of this component. Based on the evolution of the prognostics over time, maintenance tasks may be rescheduled at a higher cost.

The time when alarms are triggered is crucial. Triggering alarms when the predicted RUL is large may result in the initiated maintenance task being re-scheduled several times, as RUL prognostics are updated over time. Triggering alarms when the predicted RUL is small may result in component failures as there may not be enough time and resources left to perform maintenance. Using a genetic algorithm, we optimize the parameters of our alarm policy (the frequency of alarms, the threshold for triggering alarms and the safety margin).

We illustrate our approach for the maintenance planning of a fleet of aircraft, each equipped with two turbofan engines. By employing our alarm-based maintenance framework, the costs with engine failures account for only 7.4% of the total maintenance costs. Overall, our framework provides an end-to-end approach for maintenance scheduling of multiple components considering imperfect RUL prognostics.

The remainder of this paper is organized as follows. In Section 2 we propose a CNN to obtain RUL prognostics for turbofan engines. In Section 3, we develop an alarm-based maintenance planning framework that integrates RUL prognostics in the maintenance schedule. These prognostics are updated over time. In Section 4 we illustrate our CNN for RUL prognostics together with our alarm-based maintenance framework for a fleet of aircraft equipped with turbofan engines. Conclusions are provided in Section 5.

2. Remaining useful life prognostics using a convolutional neural network

In this section, we develop RUL prognostics for turbofan engines using Convolutional neural networks (CNNs) and the C-MAPSS turbofan engine degradation dataset [21]. CNNs have successfully been applied for RUL prognostics for turbofan engines in, for instance, [7–9].

Table 1
C-MAPSS datasets for turbofan engines [21].

	FD 001	FD 002	FD 003	FD 004
Training instances	100	260	100	249
Testing instances	100	259	100	248
Operating conditions	1	6	1	6
Fault conditions	1	1	2	2

The C-MAPSS dataset consists of simulated data on the degradation of turbofan engines. This data was generated by NASA using the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). The dataset contains multi-variate temporal data of 21 sensors. There are 4 data subsets, FD001, FD002, FD003 and FD004, each with specific operating and fault conditions. Each subset has one training set where measurements are recorded until the failure of the engine (run-to-failure instances), and one test set. In the test set, the sensor recordings are terminated somewhere before failure, and the aim is to predict the RUL at that moment. In all cases, each engine has a different level of initial wear. Over time, the condition of an engine degrades as it approaches failure. A description of the 4 subsets is given in Table 1.

From the 21 sensors considered, 7 sensors have constant values over time. As such, we select the remaining 14 sensors with non-constant measurements for the input of the CNNs. We normalize the sensor measurements with min–max normalization [7] with respect to the operating condition [8] in each subset as follows:

$$\hat{m}_{ij} = \frac{2(m_{ij}^{k} - m_{jk}^{\min})}{m_{jk}^{\max} - m_{jk}^{\min}} - 1, \tag{1}$$

with m_{ij}^{k} the sensor measurement of sensor j during flight i, where flight i was performed under operating condition k, $m_{jk}^{\min}$ and $m_{jk}^{\max}$ the minimum and maximum value in the training set of sensor j under operating condition k respectively, and $\hat{m}_{ij}$ the normalized measurement of sensor j during flight i.

2.1. Architecture of the CNN

As input for the CNN, we consider multi-dimensional data samples X:

$$X = [x_1, x_2, \dots, x_N], \tag{2}$$

where N is the number of flights included. For each flight $i \in \{1, 2, \dots, N\}$, x_i contains the sensor measurements obtained during the flight and history of the operating conditions at that flight:

$$x_i = [\hat{m}_{i1}, \hat{m}_{i2}, \dots, \hat{m}_{iM}, o_{i1}, o_{i2}, \dots, o_{iO}], \tag{3}$$

with $\hat{m}_{ij}$ the normalized measurement of the jth sensor during flight i, M the total number of sensors considered, o_{ir} the history of operating condition r at flight i, and O the number of operating conditions of the considered subset. The history of operating condition r denotes the number of flights, since the first flight of the considered engine up to flight i, an engine has performed in operating condition r [8].

Fig. 1 shows the architecture of the proposed CNN. We consider L convolutional layers. The first $L-1$ convolutional layers each have K_f kernels, and thus generates K_f feature maps. Each kernel has a size of $K_s \times 1$. We thus use one-dimensional kernels [7]. The convolutional operation in the lth convolutional layer for the nth kernel k_n^l is [22]:

$$z_n^l = \sigma(k_n^l * z^{l-1} + b_n^l) \tag{4}$$

where z_n^l is the nth feature map of layer l, $*$ is the convolutional operator, z^{l-1} are the feature maps in layer $l-1$, b_n^l is the bias of the nth feature map of layer l, and $\sigma(\cdot)$ is the activation function of the

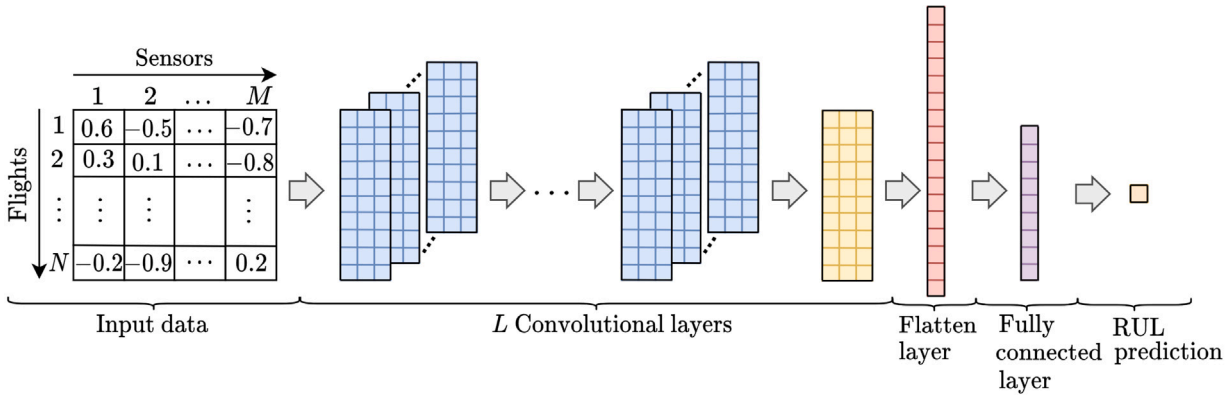


Fig. 1. Schematic overview of the CNN.

convolutional layer. The Lth convolutional layer has one kernel with a size of $K'_s \times 1$, combining all the K_f feature maps into one single feature map.

Using the extracted features of the convolutional layers, the CNN predicts the RUL. The 2-dimensional, final feature map of the last convolutional layer is flattened. In this layer, we apply a drop-out rate ρ to prevent overfitting. The flatten layer is connected with a fully connected layer, where each neuron is connected to all neurons in the flatten layer. Let z^{fl} be the output of the flatten layer, and let w^f be the weights of the fully connected layer. The output z^f of the fully connected layer is then [22]:

$$z^f = \sigma(w^f z^{\text{fl}} + b^f), \tag{5}$$

where b^f is the bias of the fully connected layer, and $\sigma(\cdot)$ is the activation function of this layer. Last, the final layer with one neuron outputs a RUL prediction using a linear activation function.

Following a grid-search hyper-parameter tuning, with the hyper-parameters of [7] as starting point, we consider $L = 5$ convolutional layers. The first 4 convolutional layers contain $K_f = 10$ kernels, each with a size of $K_s = 10 \times 1$, while the last convolutional layer contains 1 kernel with a size of $K'_s = 3 \times 1$. Same padding is implemented in all convolutional layers to ensure that the feature maps have a constant dimension. In the flatten layer, we apply a drop-out rate of $\rho = 0.5$ during training. Finally, the fully connected layer contains 100 neurons. All layers apply a hyperbolic tangent activation (tanh) function.

We optimize the weights of the CNN using the Adam optimizer [23] with a batch size of 256 samples, and a maximum of 250 training epochs. The initial learning rate is 0.001, which is multiplied by 0.6 after 10 consecutive epochs without improvement, for a stable convergence of the weights. We use a window size of 30 flights for subsets FD001 and FD003, and of 21 and 19 flights for subsets FD002 and FD004 respectively, i.e., the number of flights available for the shortest test instance in the test sets of FD002 and FD004.

2.1.1. RUL prognostics for aircraft engines

We apply our CNN to each of the 4 test data subsets FD001, FD002, FD003 and FD004. We evaluate the obtained RUL prognostics by means of the Root Mean Square Error (RMSE) metric:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{w=1}^{n} {e_w}^2}, \tag{6}$$

where n is the number of testing instances in the considered data subset and e_w is the RUL prediction error (in flights) for an engine w, $e_w = \text{RUL}_w^{\text{actual}} - \text{RUL}_w^{\text{predicted}}$. Moreover, we use a piece-wise linear RUL target function [7,9,24] with $R_{\text{early}} = 125$ flights, i.e, when the actual RUL is larger than $R_{\text{early}} = 125$ flights, the target RUL of the neural network is $R_{\text{early}} = 125$ flights.

Table 2
RMSE for the RUL prognostics using C-MAPSS and a (variant of) a CNN.

	R_{early}	FD 001	FD 002	FD 003	FD 004
Our approach	125	12.22	15.07	12.72	18.10
CNN [7]	125	12.61	22.36	12.64	23.31
MS-DCNN [9]	125	11.44	19.35	11.67	22.22
CNN with pooling [8]	Varies	18.45	30.29	19.82	29.16
CNN with pyramid pooling [25]	–	12.64	25.92	12.39	26.84

Table 3
RMSE for RUL prognostics using C-MAPSS and various machine learning algorithms.

	R_{early}	FD 001	FD 002	FD 003	FD 004
Our approach	125	12.22	15.07	12.72	18.10
LSTM-MLSA [24]	125	11.57	14.02	12.13	17.21
CNN-LSTM [26]	125	11.17	–	9.99	–
HAGCN [27]	130	11.93	15.05	11.53	15.74
HDNN [28]	125	13.02	15.24	12.22	18.17
Semi-supervised [29]	130	12.56	22.73	12.10	22.66
CapsNets [30]	125	12.58	16.30	11.71	18.96

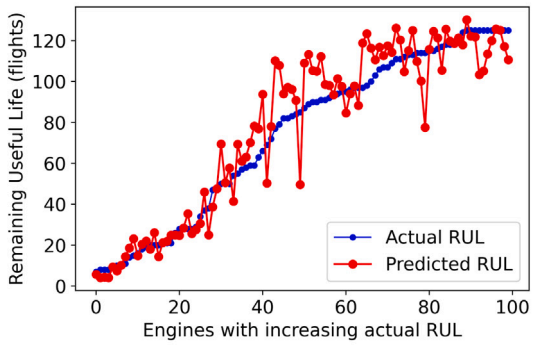
Table 2 shows the RMSE for each of the 4 test data subsets of C-MAPSS. The RMSE is highest for subsets FD002 and FD004, where the engines are subject to multiple operating conditions. Fig. 2 shows the RUL prognostic versus the actual RUL for the individual engines in the 4 test data subsets. As expected, the results show that the RUL prognostics are indeed imperfect with non-zero errors.

Table 2 also compares the performance of our RUL prognostic model with existing studies that employ CNNs for RUL prognostics as well. The results show that we obtain a lower RMSE for subsets FD002 and FD004 compared to [7–9,25]. A more advanced CNN in [9] results in a lower RMSE for subsets FD001 and FD003. Overall, our results are comparable to the best existing results for this dataset when using a CNN. Table 3 gives an overview of the performance of RUL prognostic models for the C-MAPSS dataset when considering various machine learning algorithms. A more extensive overview of such models can be found in [24,31]. The lowest RMSE is obtained when using a LSTM neural network with multi-layer self-attention [24], or a hierarchical attention graph convolutional network [27]. Also here, the performance of our RUL prognostic method is comparable with existing methods.

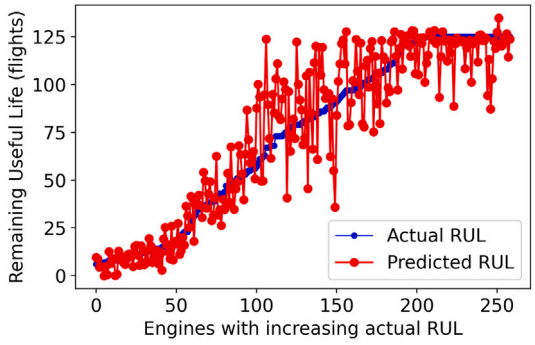
In the next section, we integrate these imperfect RUL prognostics into a predictive maintenance scheduling model for a fleet of aircraft.

3. Maintenance scheduling with imperfect RUL prognostics

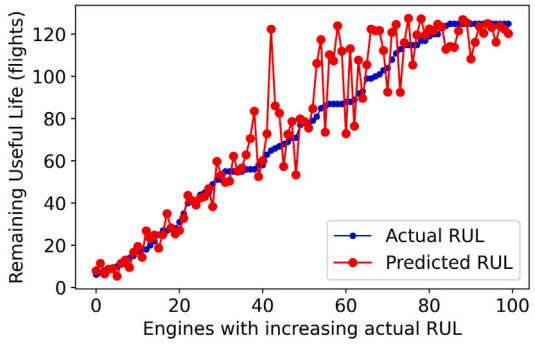
In this section, we propose a generic, alarm-based maintenance planning framework for a fleet of aircraft where imperfect RUL prognostics are available for aircraft components.



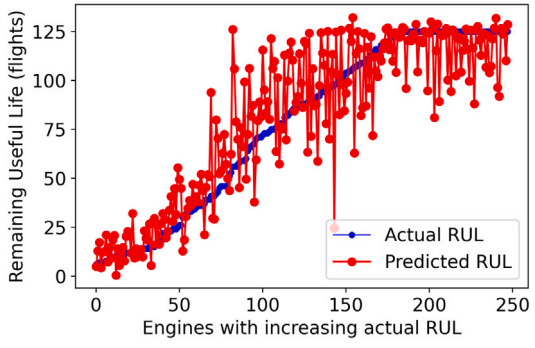
(a) FD001



(b) FD002

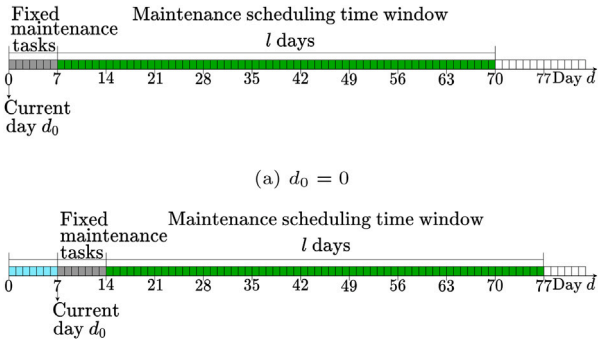


(c) FD003



(d) FD004

Fig. 2. RUL predictions of the engines in the four C-MAPSS data subsets. The engines are sorted in an increasing order of their actual RUL.



(a) $d_0 = 0$

(b) $d_0 = 7$

Fig. 3. Illustration of two sequential time windows of the rolling horizon approach with $\tau = 7$, $k = 7$ and $l = 63$ days.

3.1. Problem description

Fleet of aircraft with degrading components

Let A denote a fleet of aircraft. Each aircraft $a \in A$ is equipped with a set V_a of identical components. The health of each component degrades over time. A RUL prognostic for each component $v \in V_a$ is obtained every day.

Maintenance slots

Each aircraft $a \in A$ has allocated a set of maintenance slots S_a during which the aircraft is on the ground and maintenance is performed. These slots are scheduled months in advance. During these slots, periodic maintenance tasks and inspections are scheduled in advance as well, as described in the maintenance manuals of the aircraft [32]. Let d_s denote the day during which a maintenance slot $s \in S_a$ is planned.

Additional maintenance tasks driven by RUL prognostics

During a maintenance slot, *additional* maintenance tasks may be scheduled based on RUL prognostics, in anticipation of a failure. Performing an additional maintenance task for a component $v \in V_a$ costs c_p. We assume that at most h additional maintenance tasks can be performed every day, due to the limited availability of the maintenance engineers, specialized tools and equipment.

When a component $v \in V_a$ fails, we assume that an *unscheduled* maintenance task for this component is immediately performed at a cost $c_f > c_p$.

Rolling horizon approach and rescheduling additional maintenance tasks

We assume a discrete-time problem with time steps of 1 day. Additional maintenance tasks are planned using a rolling horizon approach with time windows. At current day d_0, we consider a scheduling time window $D_{d_0} = [d_0 + k, d_0 + k + l)$, where k is the minimum number of days required to prepare an additional maintenance task and $k + l$ is the moment in time up to which maintenance slots are known. At day d_0, additional maintenance tasks may be scheduled within this time window D_{d_0} based on the available RUL prognostics. We then advance to the next moment of maintenance scheduling at day $d_0 + \tau$ (new current day) with time window $D_{d_0+\tau} = [d_0 + \tau + k, d_0 + \tau + k + l)$ for which new, updated RUL prognostics are generated. Again, additional maintenance tasks may be scheduled in time window $D_{d_0+\tau}$, based on the updated RUL prognostics. This process is repeated for future time windows.

Fig. 3 shows an example of a sequence of two maintenance scheduling time windows with $\tau = 7$, $k = 7$ and $l = 63$ days. At current day $d_0 = 0$, we schedule additional maintenance tasks for the time window $[d_0 + k, d_0 + k + l) = [7, 70)$. We fix all additional tasks scheduled up to day $d_0 + \tau + k = 14$, and advance in time to a new current day $d_0 + \tau = 7$.

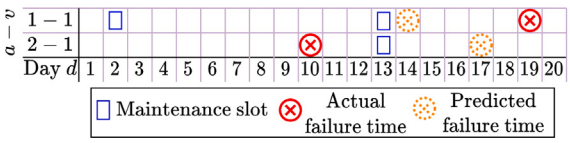


Fig. 4. Example — Maintaining 2 aircraft in one time window of the rolling horizon approach, with RUL prognostics.

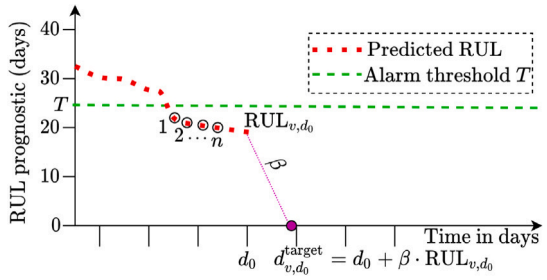


Fig. 5. Example — RUL-based alarm for a component v.

For this new current day, we now consider the scheduling time window [14, 77).

A *rescheduling* of an additional maintenance task occurs when this task has been assigned at day d_0 to a slot s in time window D_{d_0}, and the same task is re-assigned at day $d_0 + \tau$ to another slot $s' \neq s$ in a subsequent time window $D_{d_0+\tau}$. Rescheduling additional tasks may occur due to updated RUL prognostics or due to exceeding the limit h of additional tasks per day. Rescheduling an additional maintenance task costs $c_r < c_p$.

Fig. 4 shows an example of maintenance for two aircraft in one time window of the rolling horizon approach, when considering RUL prognostics. We consider the maintenance planning for aircraft 1, component 1 (1 − 1), and aircraft 2, component 1 (2 − 1). Our time window consists of 20 days. For aircraft 1, there are two maintenance slots in which we can plan an additional maintenance task for component 1: at day 2, and at day 13. Aircraft 2 has only one maintenance slot, at day 13. Component 1 of aircraft 1 is expected to fail at day 14. However, the actual failure time of this component is day 19. The RUL is thus underestimated. Component 1 of aircraft 2 is expected to fail at day 17, while it actually fails at day 10. The RUL is thus overestimated.

3.2. Maintenance scheduling for a fleet of aircraft using RUL-based alarms

We propose to schedule maintenance for a fleet of aircraft using RUL-based alarms. A schematic overview of this framework is given in Fig. 5. Every day, the RUL prognostic for a component v is updated. In case the predicted RUL falls below an alarm threshold T for n consecutive days (i.e., n days in a row), an alarm is triggered. We refer to this component as an *alarmed* component. At the next maintenance scheduling moment in the rolling horizon approach, an additional maintenance task is scheduled for this component.

We require the RUL prognostic to fall below the alarm threshold T for n *consecutive* days due to the non-monotonic trend of the prognostics. As a result, the times at which the RUL prognostic falls below an alarm threshold can be far apart (false positives). Acting on these false positives would lead to unnecessary maintenance and costs. We define n to identify a consistent behavior of degradation in a short period of time instead.

Consider current day d_0 when a RUL prognostic RUL_{v,d_0} for an alarmed component v is available. Ideally, we would schedule maintenance at day $d_0 + \text{RUL}_{v,d_0}$ to minimize the wasted life of the component, while avoiding a failure. However, since we consider imperfect RUL prognostics, we assume a safety factor β, $0 \leq \beta \leq 1$, such that we aim to schedule maintenance before a *target* day instead:

$$d^{\text{target}}_{v,d_0} = d_0 + \beta \cdot \text{RUL}_{v,d_0}.$$

Scheduling maintenance for alarmed components: one time window

Given an alarm threshold T, an n number of times the RUL prognostic falls below T before an alarm is triggered, and a safety factor β, we propose the following maintenance scheduling model for a fleet of aircraft. The objective is to minimize the total maintenance costs. This model is applied for each time window of the rolling horizon approach.

Let V^{alarm} denote the set of alarmed components in the fleet of aircraft at current day d_0, i.e., the set of components for which we should schedule maintenance. Let $S_{a,D_{d_0}} \subseteq S_a$ denote the set of all available maintenance slots for aircraft $a \in A$ such that the slot is within the scheduling time window D_{d_0}, i.e., $d_s \in D_{d_0}$, $\forall s \in S_{a,D_{d_0}}$.

Due to the limited capacity h and the limited number of maintenance slots, it may be that not all alarmed components $v \in V^{\text{alarm}}$ can be maintained in the scheduling time window D_{d_0}. In this case, we assume that a buffer, generic slot is available to maintain the aircraft [18]. Let s^{gen} be a generic slot at day $d_{s^{\text{gen}}} = d_0$. When using this generic slot, a very high cost $c_g \gg c_f$ is incurred. This slot does not have capacity constraints. All aircraft may be maintained in this generic slot. The set of slots $S_{a,D_{d_0}}$ available for each aircraft $a \in A$ thus includes this generic slot.

We consider the following integer linear program to schedule additional maintenance tasks at day d_0 in the scheduling time window D_{d_0}:

Decision variable. The decision variable is defined as follows:

$$x_{avs} = \begin{cases} 1, & \text{component v} \in V_a \cap V^{\text{alarm}} \text{ of} \\ & \text{aircraft } a \in A \text{ is maintained in} \\ & \text{slot} s \in S_{a,D_{d_0}}, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

Objective function. Let c_{avs} denote the costs of scheduling an additional maintenance task for component $v \in V_a \cap V^{\text{alarm}}$ of aircraft $a \in A$ during slot $s \in S_{a,D_{d_0}}$. We define c_{avs} as:

$$c_{avs} = p^{\text{late}}(d_s - d^{\text{target}}_{v,d_0})^+ + p^{\text{early}}(d^{\text{target}}_{v,d_0} - d_s)^+ + p^{\text{res}} I^{\text{res}}_v + p^{\text{gen}} I^{\text{gen}}_s,$$

where p^{late} is a penalty for each day an additional maintenance task is scheduled after the target day $d^{\text{target}}_{v,d_0}$, p^{early} is a penalty for each day an additional maintenance task is scheduled before the target day $d^{\text{target}}_{v,d_0}$ (wasting component life), p^{res} is a penalty for rescheduling an additional maintenance task for a component v to a new slot and $p^{\text{gen}} \gg p^{\text{late}}, p^{\text{early}}, p^{\text{res}}$ is the penalty for using the generic slot s^{gen}. Last, I^{res}_v and I^{gen}_s are indicator functions:

$$I^{\text{res}}_v = \begin{cases} 1, & \text{additional maintenance task for} \\ & \text{component } v \text{ is rescheduled,} \\ 0, & \text{otherwise,} \end{cases}$$

$$I^{\text{gen}}_s = \begin{cases} 1, & \text{slot } s \in S_{a,Dd_0} \text{ is the generic slot } s^{\text{gen}} \\ 0, & \text{otherwise.} \end{cases}$$

The objective is to minimize the total cost of scheduling additional maintenance tasks:

$$\text{min.} \quad \sum_{a \in A} \sum_{v \in V_a \cap V^{\text{alarm}}} \sum_{s \in S_{a,D_{d_0}}} c_{avs} x_{avs}. \tag{8}$$

Constraints. We consider the following constraints:

$$\sum_{s \in S_{a,D_{d_0}}} x_{avs} = 1, \quad \forall a \in A, \forall v \in V_a \cap V^{\text{alarm}} \tag{9}$$

$$\sum_{a\in A}\sum_{v\in V_a\cap V^{\text{alarm}}}\sum_{s\in S_{a,D_{d_0}}\setminus\{s^{\text{gen}}\}:\,d_s=d} x_{avs}\le h,\ \forall d\in D_{d_0} \tag{10}$$

Constraint (9) ensures that an additional maintenance task is scheduled for each alarmed component $v\in V^{\text{alarm}}$. Constraint (10) ensures that at most h additional tasks are scheduled every day, aside from the generic slot.

This model assumes known values for T, n and β. We next determine these values.

3.3. Defining an alarm policy (T, n, β) using a genetic algorithm

We now optimize the alarm threshold T, the number of times n that the RUL prognostic falls below T before an alarm is triggered, and the safety factor β of our maintenance planning framework using a genetic algorithm (GA) [33]. The aim is to minimize the long-term maintenance costs.

Agent chromosomes and initialization. Let Θ_i be the population of $|\Theta_i| = N$ agents in iteration i of the GA. Let θ_p be the chromosome with the parameters of agent $\theta\in\Theta_i$:

$$\theta_p = (T^\theta, n^\theta, \beta^\theta), \tag{11}$$

with T^θ the alarm threshold, n^θ the number of times the RUL prognostic falls below T before an alarm is triggered, and β^θ the safety factor. These parameters define agent $\theta\in\Theta_i$. We initialize T^θ with a random integer in $[k, l]$, n^θ with a random integer in $[1, 5]$ and β^θ with a random value from $[0.01, 0.02, \dots, 1.00]$ for all agents in the initial generation Θ_0.

Selection of the parents. Let θ_f denote the fitness of agent $\theta\in\Theta_i$ of iteration i. We select N parents for the population of the $i+1$th generation with tournament selection: For each parent, we first randomly select r individual agents from Θ_i. Then, the agent with the highest fitness θ_f of these r agents is selected as a parent.

Reproduction: crossover and mutation. Each pair of 2 parents selected from Θ_i generates 2 new child agents, that become part of the population Θ_{i+1} of generation $i+1$. We use one-point crossover [33] to generate the two new child agents. We mutate each element of the new child agent chromosome with probability $\tilde{p}$ using random resetting [33].

Termination. The GA is terminated after M generations. The agent with the highest fitness across all generations is selected as the final agent.

Monte Carlo simulation to evaluate the fitness of a GA agent

We evaluate the fitness θ_f of an agent θ with parameters $\theta_p = (T^\theta, n^\theta, \beta^\theta)$ by calculating the expected maintenance costs over a period of 10 years using Monte Carlo simulation.

We perform 100 Monte Carlo simulation runs (iterations). For each iteration $i\in\{1, 2, \dots, 100\}$, we generate a maintenance planning over a period of 10 years using the rolling horizon approach (see Section 3.1). At the beginning of each scheduling time window D_{d_0} in the rolling horizon approach, we update the RUL prognostics. With these RUL prognostics and the alarm threshold T^θ and n^θ of the agent θ under consideration, we determine the set of alarmed components V^{alarm}. We also determine the target day to maintain each alarmed component using the safety factor β^θ and the updated RUL prognostics. We then assign each alarmed component to exactly one maintenance slot in the scheduling time window D_{d_0} (see Eqs. (7)–(10)).

After each iteration i, we calculate the costs $\theta_{c,i}$ of the generated maintenance planning using c_f, the costs of an engine failure, c_p, the costs of an additional maintenance task, c_r, the costs of rescheduling a maintenance task and c_g, the costs of using a generic slot. We define the fitness of agent θ as:

$$\theta_f = \frac{1}{\frac{1}{100}\sum_{i=1}^{100}\theta_{c,i}}, \tag{12}$$

i.e., the higher the expected costs, the lower the fitness of the agent.

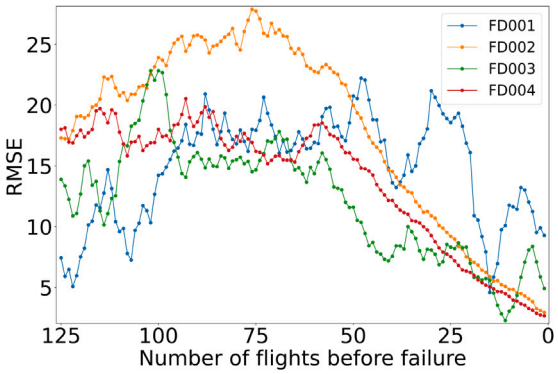


Fig. 6. RMSE of the RUL predictions over time for the engines in E, splitted over the four C-MAPSS subsets.

4. Case study — engine maintenance scheduling for a fleet of aircraft

We apply our maintenance planning framework in Section 3 to the engines in the C-MAPSS data set [21]. Since in the test set of C-MAPSS the sensor measurements terminate at some time before engine failure, RUL prognostics cannot be generated after every flight until failure. Therefore, we apply our maintenance framework for 14% of the training instances of C-MAPSS, which are run-to-failure instances. A similar approach has been taken in [16]. The 14% instances are randomly selected from the C-MAPSS training subsets, i.e., we randomly select 14% of the engines from subset FD001, resulting in 14 engines selected, 14% of the engines from subset FD002, resulting in 37 engines selected, 14% of the engines from subset FD003, resulting in 14 engines selected, and 14% of the engines from subset FD004, resulting in 35 engines selected. In total, we select 100 engines for which we apply our proposed maintenance framework. For these 100 instances, we obtain RUL prognostics using CNNs, as discussed in Section 2. For this, we do not use any knowledge about the actual failure times of these engines. The remaining 86% of the training instances of each subset are used to train the CNNs.

Let E denote the set of the selected 100 turbofan engines. These engines have an average lifespan of 204 flights, with a minimum lifespan of 110 flights, and a maximum lifespan of 430 flights.

4.1. Imperfect remaining useful life prognostics for turbofan engines

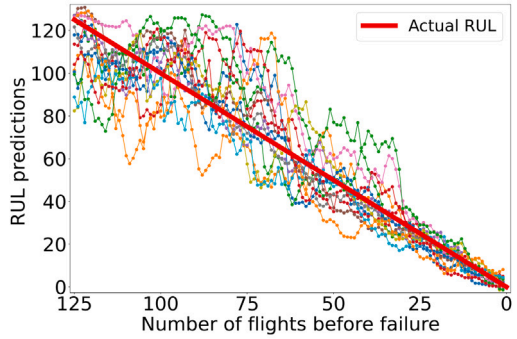
Using the CNN as discussed in Section 2, we obtain a RUL prognostic after every flight for each engine in the set E. Fig. 7 shows the obtained RUL predictions up to $R_{\text{early}} = 125$ flights before failure for each engine in E. Fig. 6 shows the RMSE for the obtained prognostics up to 125 flights before failure. The accuracy of the RUL prognostics varies over time and across the 4 subsets of the C-MAPSS dataset.

We evaluate the obtained series of RUL prognostics using the RMSE, the Cumulative Relative Accuracy (CRA) and the Convergence of the RMSE metrics [34]. The CRA_λ is defined as follows [34]:

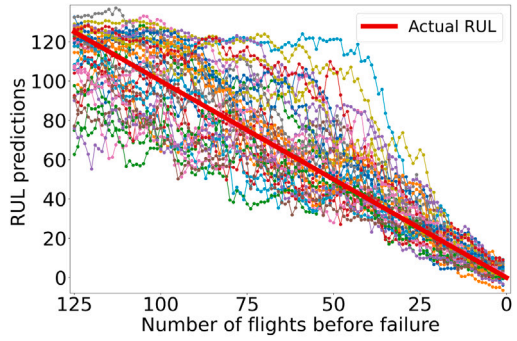
$$\text{CRA}_\lambda = \frac{1}{n}\sum_{w=1}^{n} 1 - \frac{|\text{RUL}_{w,\lambda}^{\text{actual}} - \text{RUL}_{w,\lambda}^{\text{predicted}}|}{\text{RUL}_{w,\lambda}^{\text{actual}}},$$

where n is the number of components in the considered data subset, $\text{RUL}_{w,\lambda}^{\text{actual}}$ is the actual RUL of engine w at λ percent of its lifetime (i.e., $\lambda = 0.5$ gives the actual RUL of engine w halfway its lifetime, and $\lambda = 0.9$ gives the actual RUL of engine w at 90% of its lifetime), and $\text{RUL}_{w,\lambda}^{\text{predicted}}$ is the predicted RUL of engine w at λ percent of its lifetime.

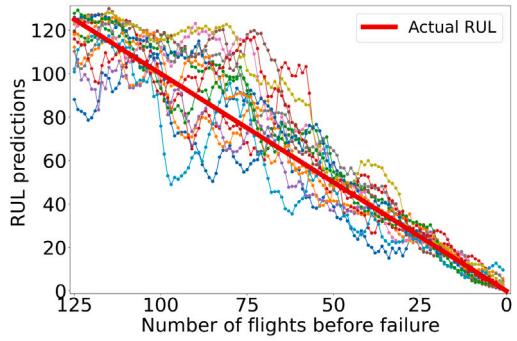
The convergence of the RMSE metric [34] quantifies how fast the RMSE metric converges over time to its minimum value, assuming that



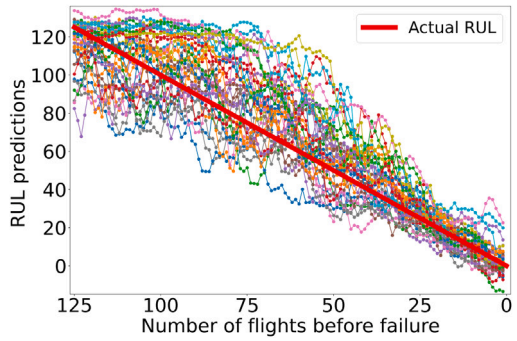
(a) The 14 test engines of data set FD001



(b) The 37 test engines of data set FD002



(c) The 14 test engines of data set FD003



(d) The 35 test engines of data set FD004

Fig. 7. RUL predictions over time for the engines in E, splitted over the four C-MAPSS subsets.

the RUL predictions improve over time. The convergence of the RMSE is defined as:

$$\text{Convergence} = \sqrt{(x_c - R_{\text{early}})^2 + y_c^2},$$

Table 4
RMSE, Cumulative Relative Accuracy (CRA), and Convergence of the RMSE, for the RUL prognostics results for the run-to-failure data of the engines in set E.

	FD 001	FD 002	FD 003	FD 004
RMSE	14.35	17.98	12.05	15.15
$\text{CRA}_{0.5}$	0.79	0.74	0.79	0.80
$\text{CRA}_{0.9}$	0.93	0.97	0.97	0.98
Convergence (flights)	64.64	51.96	52.07	51.00

where (x_c, y_c) is the centroid of the area under the RMSE curve in Fig. 6. Notice that this curve starts only at $R_{\text{early}} = 125$ flights before failure. The lower the convergence, the faster the RMSE converges.

Table 4 shows the RMSE, the $\text{CRA}_\lambda, \lambda \in \{0.5, 0.9\}$ and the Convergence of the RMSE, obtained for the engines in the set E. The results show that CRA_λ improves with increasing λ, i.e., the RUL prognostics improve as the engines approach the actual time of failure. The Convergence of the RMSE ranges between 51.00 and 64.64, and the highest Convergence of the RMSE is obtained for data subset FD001.

4.2. Results — alarm-based maintenance scheduling for aircraft engines

We consider a fleet of $|A| = 20$ aircraft, each equipped with $|V_a| = 2$ engines. The engines are randomly selected from E. For each aircraft $a \in A$, we label the two engines as $a-1$ (first engine of aircraft a) and $a-2$ (second engine of aircraft a).

For each aircraft, we consider maintenance slots with a frequency of 10–20 days, reflecting a realistic maintenance slots frequency [32]. Moreover, an additional maintenance task can be scheduled for only $h = 1$ engine per day, and at least $k = 7$ days (one week) are needed to prepare an additional maintenance task. Also, the available maintenance slots are known up to $k + l = 70$ days (10 weeks) in advance, and the maintenance schedule is updated every $\tau = 7$ days (one week). We assume that each aircraft performs one flight per day.

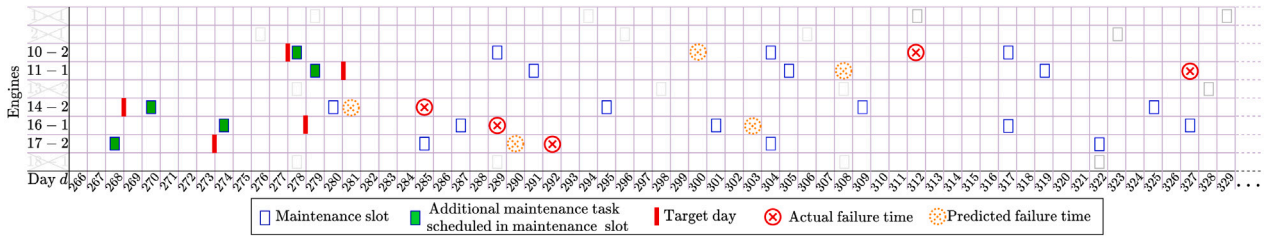
We assume a cost $c_r = 5000$ for rescheduling an additional maintenance task, $c_p = 10{,}000$ for performing an additional maintenance task, $c_f = 50{,}000$ for an engine failure and $c_g = 10^6$ for using a generic slot. For the objective function of the integer linear program in Section 3.2, we consider the maintenance penalties $p^{\text{early}} = 1$ for every day maintenance is scheduled earlier than the target day, $p^{\text{res}} = 100$ for rescheduling an additional maintenance task, $p^{\text{late}} = 1000$ for every day maintenance is scheduled after the target day and $p^{\text{gen}} = 10^6$ for using a generic slot.

With these penalties, we thus consider the target day of a maintenance task as a strict deadline; performing maintenance far before the target day is preferred over performing maintenance just after the target day.

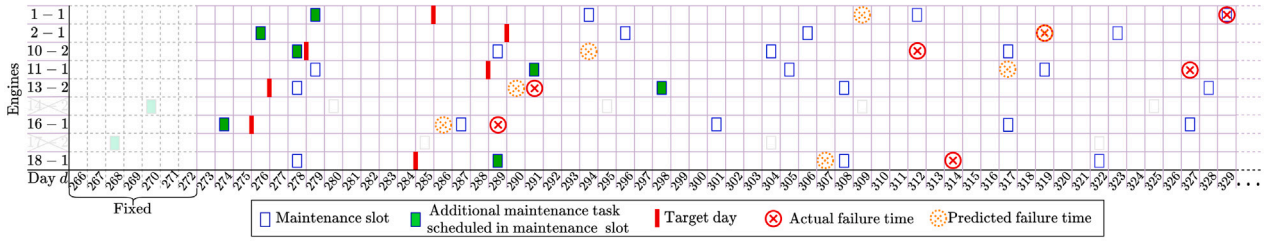
To determine the alarm policy (T, n, β) using the GA in Section 3.3, we consider $N = 30$ agents per population, $M = 20$ generations, $r = 5$ participants in each tournament and a mutation probability $\tilde{p} = 1/3$. As result, we obtain the alarm threshold $T = 49$ days, $n = 1$ and the safety factor $\beta = 0.44$.

Fig. 8 shows the maintenance schedule with this alarm policy for two sequential time windows of the rolling horizon approach, at day $d_0 = 259$ and day $d_0 = 266$. Here, from the total of 40 engines, we only show the alarmed engines. At day d_0, the beginning of scheduling window D_{d_0}, we update the RUL prognostics of the engines. With these updated RUL prognostics, the alarm threshold $T = 49$ days, $n = 1$ and the safety factor $\beta = 0.44$, we then determine the alarmed engines and the corresponding target days. These target days and the available maintenance slots are the input of the integer linear program in . An alarmed engine is assigned to exactly one maintenance slot by this integer linear program.

For example, aircraft 17, engine 2 (17-2) has four maintenance slots in Fig. 8: at days 268, 285, 304 and 322. The predicted failure time of



(a) Maintenance schedule created at day $d_0 = 259$.



(b) Maintenance schedule created at day $d_0 = 266$.

Fig. 8. Maintenance schedule in 2 sequential time windows of the rolling horizon approach, at day $d_0 = 259$ and day $d_0 = 266$.

engine 17-2 is at day 290, while the actual failure time is at day 292. A maintenance task for engine 17-2 is scheduled at day 268, well before the target day at day 273.

At day $d_0 = 259$, aircraft 10, 11, 14, 16 and 17 each have one alarmed engine. For each alarmed engine, an additional maintenance task is scheduled in the first maintenance slot available. For the alarmed engines of aircraft 11, 16 and 17, this maintenance slot is before the target day, while for the alarmed engines of aircraft 10 and 14, this maintenance slot is after the target day.

At day $d_0 = 266$, the additional maintenance tasks for aircraft 14 and 17 are fixed from the previous time window. However, the additional maintenance tasks for aircraft 10, 11 and 16 may still be rescheduled. Moreover, aircraft 1, 2 and 18 also have an alarmed engine now. For aircraft 10 and 16, the additional maintenance task remains planned at day 278 and day 274 respectively, as in the previous time window. However, the additional maintenance task for aircraft 11 is rescheduled from day 279 to day 291. This reschedule is because only $h = 1$ additional maintenance task can be planned per day, and at day 279 an additional maintenance task is now scheduled for engine 1-1. Moreover, the maintenance task of engine 13-2 is only planned at day 298, 22 days after its target day and also after its failure time. This is because the only maintenance slot available for this aircraft before day 298 is at day 278, when maintenance for engine 10-2 is already scheduled. At the next maintenance opportunity for engine 10-2, maintenance for engine 18-1 is already scheduled. If we maintain engine 13-2 at day 278, we would thus have to reschedule the maintenance for engine 10-2 to day 304.

Table 5 shows the alarmed engines at day $d_0 = 259$ and day $d_0 = 266$. At the moment of the alarm, the predicted RUL is between 39 and 48 days, while the actual RUL is between 30 to 70 days. The RUL prediction error at the moment of the alarm has an error between −22 days (underestimation of the failure time) to 14 days (overestimation of the failure time). The error in the RUL prognostics underlines the need for a safety factor β, with which a target day is determined.

The computational time to solve the maintenance schedule for the first and second time window using the integer linear program in Section 3.2 is 0.012 s and 0.022 s respectively, on a computer with an Intel Core i7 processor at 2.11 GHz and 8Gb RAM. The integer linear program is solved using the optimizer Gurobi version 9.0.2 with standard settings, implemented in Python.

Table 5
The day of the alarm, and the predicted and actual RUL at the moment of the alarm, for all engines that are alarmed at day $d_0 = 259$ and day $d_0 = 266$.

Engine $(a-1,\ a-2)$	Day of alarm	Predicted RUL at alarm (days)	Actual RUL at alarm (days)
1-1	265	46	64
2-1	261	44	58
10-2	259	41	53
11-1	257	48	70
13-2	263	39	28
14-2	243	47	42
16-1	259	44	30
17-2	254	42	38
18-1	262	48	52

Table 6
Failures occurring in 10 years of operations for a fleet of 20 aircraft using the maintenance framework. The day of the alarm is calculated since the beginning of the simulation, i.e., since day 0.

Engine $(a-1,\ a-2)$	Day of alarm	Predicted RUL at alarm (days)	Actual RUL at alarm (days)	Actual failure day	Target day d^{target} at alarm
13-1	103	45	21	124	122
11-2	229	48	25	254	250
12-1	227	48	28	255	248
13-2	263	40	28	291	280
6-2	358	45	35	393	377
4-1	1188	37	29	1217	1204
10-1	1859	48	25	1884	1880
1-2	2175	48	28	2203	2196
16-1	2531	44	23	2554	2550
20-1	2683	44	33	2716	2702
17-2	3143	48	28	3171	3164
1-1	3193	46	42	3235	3213

4.2.1. *Engine failures under the proposed alarm-based maintenance framework*

In this section, we analyze the engine failures that occur during a period of 10 years when applying our maintenance framework for a fleet of $|A| = 20$ aircraft with $|V_a| = 2$ engines.

A total of 12 engine failures occur in the considered time period of 10 years. These failures are described in Table 6. At the day of the alarm, the RUL is overestimated by 4–24 days. Using our proposed

Table 7
Long-term expected performance when considering imperfect and perfect RUL prognostics. The results are obtained with 95% confidence intervals that have a maximum width of 0.6 engine failures, 1.4 rescheduled maintenance tasks and 1.2 additional maintenance tasks, respectively.

	Imperfect RUL prognostics	Perfect RUL prognostics
Engine failures	13.61	0.10
Rescheduled maintenance tasks	67.26	2.15
Additional maintenance tasks	819.7	739.0

safety factor $\beta = 0.44$, the target day is 2–22 days before the actual failure time at the day of the alarm. Should additional maintenance tasks for these 12 engines have been scheduled before or at the initial target day, then these engines would thus not have failed.

However, the maintenance tasks are scheduled after the initial target day. For 6 out of the 12 engine failures, this is because no maintenance slot is available before the target day, i.e., there is a lack of maintenance slots. For the other failures, a maintenance slot is available before the target day. However, an additional maintenance task for another engine is already scheduled during the day of this maintenance slot, whereas at most $h = 1$ additional tasks can be scheduled per day. Increasing the number of maintenance slots or the maintenance capacity would thus help to decrease the number of engine failures and increase the reliability of the aircraft.

4.3. Maintenance with perfect RUL prognostics vs. imperfect RUL prognostics

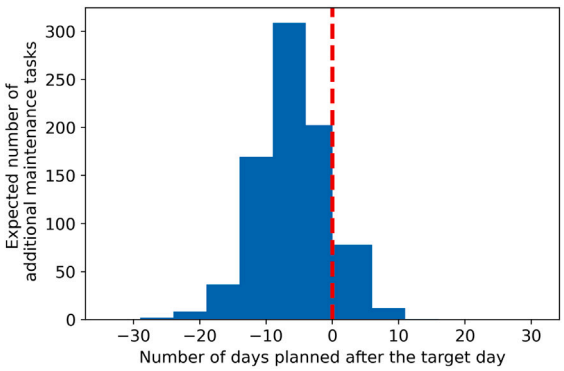
We evaluate the performance of our proposed maintenance framework for a fleet of $|A| = 20$ aircraft, each equipped with $|V_a| = 2$ engines, over a period of 10 years using Monte Carlo simulation. We compare the performance of our framework when considering perfect and imperfect RUL prognostics.

Predictive maintenance planning with perfect RUL prognostics

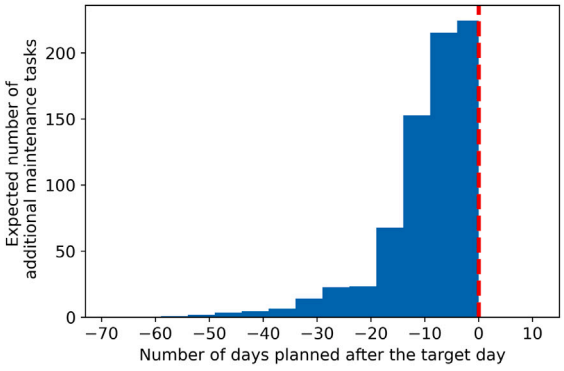
We apply our maintenance framework in Section 3.2 together with perfect RUL prognostics, i.e., the RUL predictions equal the actual RUL of the engines. As we have perfect RUL prognostics, we set the safety factor $\beta = 1$. Moreover, we do not postpone planning maintenance to obtain more accurate RUL prognostics or avoid reschedulements due to updated RUL prognostics. Instead, we define that an engine becomes alarmed as soon as its RUL prediction is below an alarm threshold $T = k + l = 70$ days, for $n = 1$ day in a row. Here, $k + l$ is the period of time up to which maintenance slots are known.

Using perfect RUL prognostics, more than 99.9% of the maintenance tasks are scheduled before the target day (see Fig. 9(b)). This is possible since an engine becomes alarmed $k + l = 70$ days before its target day. There are thus multiple possible maintenance slots in which the additional maintenance task can be scheduled. The expected number of engine failures is therefore nearly zero (see Table 7). In contrast, 11% of the additional maintenance tasks are planned after the target day when considering imperfect RUL prognostics (see Fig. 9(a)). This is because an engine becomes alarmed when the predicted RUL equals $T = 49$ days or less. Once a component becomes alarmed, only $\beta \cdot$ predicted RUL $\leq 0.44 \cdot 49 = 21$ days or less are thus available before the target day. There is, therefore, a smaller time window to schedule an additional maintenance task. This leads, in combination with the imperfect RUL prognostics, to a larger expected number of engine failures and thus to less reliable aircraft (see Table 7). We note that the generic slot is never used in our case study, showing that the slots were sufficient to perform the required maintenance tasks.

When using imperfect RUL prognostics, the expected number of rescheduled maintenance tasks equals 67.26, while it equals only 2.15



(a) Maintenance planning with imperfect RUL prognostics



(b) Maintenance planning with perfect RUL prognostics

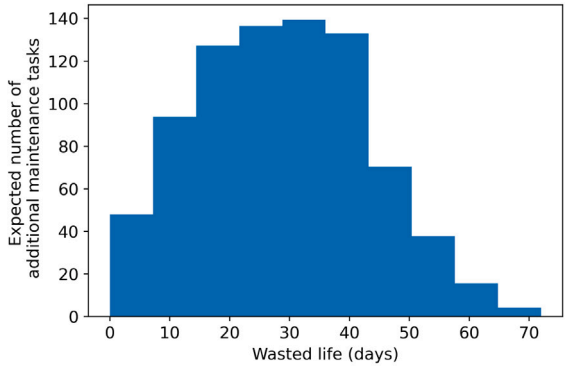
Fig. 9. Expected number of days an additional maintenance task is scheduled before (negative number) or after (positive number) the final target day, using perfect and imperfect RUL prognostics.

when considering perfect RUL prognostics (see Table 7). For imperfect RUL prognostics, the number of rescheduled maintenance tasks is higher since, (i) the target day changes over time due to updated RUL prognostics and (ii) the engines become alarmed $\beta \cdot$ predicted RUL $\leq 0.44 \cdot 49 = 21$ days or less before their target day, resulting in a smaller time window to find an optimal maintenance moment for each engine. When considering imperfect RUL prognostics, more engine life is wasted as well due to the safety factor β (see Fig. 10). As a consequence, more additional maintenance tasks are performed than when considering perfect RUL prognostics (see Table 7).

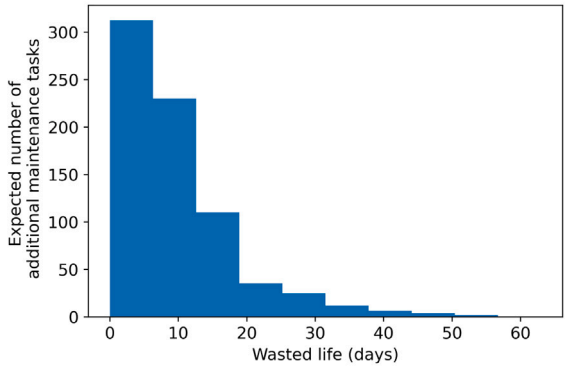
The total expected costs when performing maintenance using imperfect RUL prognostics is 24.3% higher than when using perfect RUL prognostics (see Fig. 11). In both cases, most of the costs are driven by the costs of performing additional maintenance tasks: 89.7% of the costs come from performing additional maintenance tasks when using imperfect RUL prognostics, while 99.8% of the costs come from performing additional maintenance tasks when using perfect RUL prognostics. When considering imperfect RUL prognostics, 7.4% from the costs are due to engine failures. Also, only 3.7% of the costs are a result of rescheduling maintenance tasks.

4.4. Sensitivity analysis — hyperparameters of the genetic algorithm

We perform a sensitivity analysis to evaluate the influence of the hyperparameters on the GA in Section 3.3. We consider the number of agents $N \in \{10, 30, 50\}$, the mutation probability $\tilde{p} \in \{0.1, 1/3, 0.5\}$, the number of participants in the tournament selection $r \in \{5, 10\}$ and $M = 50$ generations. Table 8 shows the fitness θ_f, the mean costs of the generated maintenance planning (see Eq. (12)), and iteration i of the



(a) Maintenance planning with imperfect RUL prognostics



(b) Maintenance planning with perfect RUL prognostics

Fig. 10. Expected engine wasted life, using perfect and imperfect prognostics.

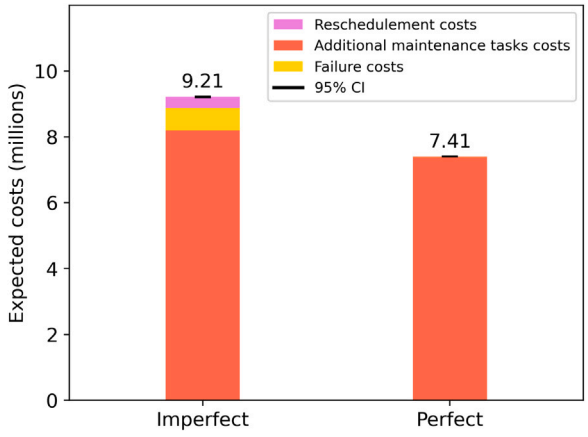


Fig. 11. Expected costs over a period of 10 years, using imperfect and perfect RUL prognostics.

GA when the best agent is found first. The fitness θ_f of the final agents θ ranges from $1.100 \cdot 10^{-7}$ to $1.106 \cdot 10^{-7}$, with a corresponding mean cost per iteration of the Monte Carlo simulation run between 9.043 to 9.092 million, i.e., a difference of 0.5%. The performance of the GA is thus robust: a similar solution is found with all considered combinations of the hyperparameters.

The final agent with the maximum fitness of $1.106 \cdot 10^{-7}$, and the corresponding costs of 9.043 million, is consistently found with a mutation probability of $\tilde{p} = \frac{1}{3}$ and $N = 30$ or $N = 50$ agents. With $\tilde{p} = \frac{1}{3}$, $N = 30$ and $r \in \{5, 10\}$, the best fitness is obtained in just 9 and 8 iterations of the GA, respectively.

Table 8
Sensitivity analysis — hyperparameters of the GA. The best agent is first found in iteration i.

$\tilde{p}$	N	r	Iteration i	$\theta_f \cdot 10^7$	Mean cost (millions)
0.1	10	5	12	1.103	9.068
	30	5	24	1.105	9.047
		10	15	1.103	9.067
	50	5	5	1.106	9.043
		10	48	1.105	9.047
$\frac{1}{3}$	10	5	43	1.100	9.092
	30	5	9	1.106	9.043
		10	8	1.106	9.043
	50	5	33	1.106	9.043
		10	6	1.106	9.043
0.5	10	5	22	1.101	9.085
	30	5	6	1.106	9.043
		10	43	1.104	9.058
	50	5	30	1.103	9.067
		10	18	1.105	9.047

5. Conclusions

We have proposed a dynamic maintenance framework for a fleet of aircraft where component RUL prognostics are updated periodically. Maintenance task scheduling is initiated as soon as an alarm is triggered. These alarms are based on the evolution of the RUL prognostics over time. Tasks are scheduled using a rolling horizon approach with time windows. In each time window, an integer linear program specifies the slots in which maintenance is scheduled. The ideal time to schedule a task is determined based on the RUL prognostics and a safety factor, to account for potential errors in the RUL prognostics. The parameters of the maintenance framework are obtained using a genetic algorithm.

We illustrate our maintenance framework for a fleet of 20 aircraft, each equipped with 2 turbofan engines. RUL prognostics of the turbofan engines are obtained using a CNN. These prognostics are updated every day. The results show that, using our maintenance framework, alarms are triggered early enough to enable the scheduling of additional tasks such that failures are prevented. The total cost savings with failure prevention outweigh the costs with potential task rescheduling due to an early alarm. The results also show that engine failures still occur due to the limited availability of maintenance slots or due to the limited number of maintenance tasks that can be performed per day. The long-term results show that the costs due to engine failures account for only 7.4% of the total maintenance costs. When comparing with the ideal case of perfect RUL prognostics, the maintenance costs are 24.4% higher.

The proposed maintenance planning framework is readily applicable to other aircraft components and systems as well. Of course, the costs considered should be adjusted accordingly. For example, the costs of a component failure c_f and the corresponding penalty p^{late} for planning a maintenance task after the target day may be lowered if a component is non-safety critical, or if many redundant components are present in a system.

In general, we can apply the proposed maintenance planning framework to generic, condition-monitored assets from other industries as well, e.g., fleet of trains, fleet of (electric) busses and fleet of ships. Additional industry-specific constraints may be considered for the maintenance of these assets.

CRediT authorship contribution statement

Ingeborg de Pater: Conceptualization, Formal analysis, Methodology, Software, Visualization, Writing – original draft, Writing – review

& editing. **Arthur Reijns:** Data curation, Methodology, Software. **Mihaela Mitici:** Conceptualization, Funding acquisition, Methodology, Project administration, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Markou C, Cros G. Airline maintenance cost executive commentary FY2019 data Public Version. Maintenance Cost Technical Group. IATA; 2020.

[2] Hu Y, Miao X, Si Y, Pan E, Zio E. Prognostics and health management: A review from the perspectives of design, development and decision. Reliab Eng Syst Saf 2021;108063.

[3] Lei Y, Li N, Guo L, Li N, Yan T, Lin J. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. Mecha Syst Signal Process 2018;104:799–834.

[4] de Pater I, Mitici M. Model-based remaining-useful-life prognostics for aircraft cooling units. In: PHM society european conference, vol. 6; 2021. p. 8.

[5] Mitici M, de Pater I. Online model-based Remaining-Useful-Life prognostics for aircraft cooling units using time-warping degradation clustering. Aerospace 2021;8(6):168.

[6] Lee J, Mitici M. An integrated assessment of safety and efficiency of aircraft maintenance strategies using agent-based modelling and stochastic Petri nets. Reliab Eng Syst Saf 2020;202:107052.

[7] Li X, Ding Q, Sun J-Q. Remaining Useful Life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf 2018;172:1–11.

[8] Babu GS, Zhao P, Li X-L. Deep convolutional neural network based regression approach for estimation of Remaining Useful Life. In: International conference on database systems for advanced applications. Springer; 2016, p. 214–28.

[9] Li H, Zhao W, Zhang Y, Zio E. Remaining Useful Life prediction using multi-scale deep convolutional neural network. Appl Soft Comput 2020;89:106113.

[10] Cao Y, Ding Y, Jia M, Tian R. A novel temporal convolutional network with residual self-attention mechanism for Remaining Useful Life prediction of rolling bearings. Reliab Eng Syst Saf 2021;107813.

[11] Li X, Zhang W, Ding Q. Deep learning-based Remaining Useful Life estimation of bearings using multi-scale feature extraction. Reliab Eng Syst Saf 2019;182:208–18.

[12] Jimenez JJM, Schwartz S, Vingerhoeds R, Grabot B, Salaün M. Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. J Manufact Syst 2020;56:539–57.

[13] Verbert K, De Schutter B, Babuška R. Timely condition-based maintenance planning for multi-component systems. Reliab Eng Syst Saf 2017;159:310–21.

[14] Andriotis C, Papakonstantinou K. Managing engineering systems with large state and action spaces through deep reinforcement learning. Reliab Eng Syst Saf 2019;191:106483.

[15] de Pater I, del Mar Carillo Galera M, Mitici M. Criticality-based predictive maintenance scheduling for aircraft components with a limited stock of spare components. In: Proceedings Of The 31st European safety and reliability conference; 2021.

[16] Nguyen KT, Medjaher K. A new dynamic predictive maintenance framework using deep learning for failure prognostics. Reliab Eng Syst Saf 2019;188:251–62.

[17] Yiwei W, Christian G, Binaud N, Christian B, Haftka RT, et al. A cost driven predictive maintenance policy for structural airframe maintenance. Chin J Aeronaut 2017;30(3):1242–57.

[18] de Pater I, Mitici M. Predictive maintenance for multi-component systems of repairables with Remaining-Useful-Life prognostics and a limited stock of spare components. Reliab Eng Syst Saf 2021;214:107761.

[19] Lee J, Mitici M. Multi-objective design of aircraft maintenance using Gaussian process learning and adaptive sampling. Reliab Eng Syst Saf 2022;218:108123.

[20] Wang Y, Limmer S, Van Nguyen D, Olhofer M, Bäck T, Emmerich M. Optimizing the maintenance schedule for a vehicle fleet: a simulation-based case study. Eng Optim 2021;1–14.

[21] Saxena A, Goebel K. Turbofan engine degradation simulation data set. NASA Ames Prognost Data Reposit 2008;878–87.

[22] Wang B, Lei Y, Li N, Yan T. Deep separable convolutional network for Remaining Useful Life prediction of machinery. Mech Syst Signal Process 2019;134:106330.

[23] Kingma DP, Ba J. Adam: A method for stochastic optimization. 2014, arXiv preprint arXiv:1412.6980.

[24] Xia J, Feng Y, Lu C, Fei C, Xue X. LSTM-based multi-layer self-attention method for Remaining Useful Life estimation of mechanical systems. Eng Fail Anal 2021;125:105385.

[25] Song Y, Zhang Y, Bliek L, Xia T. A temporal pyramid pooling-based convolutional neural network for Remaining Useful Life prediction. In: European safety and reliability conference; 2021.

[26] Peng C, Chen Y, Chen Q, Tang Z, Li L, Gui W. A Remaining Useful Life prognosis of turbofan engine using temporal and spatial feature fusion. Sensors 2021;21(2):418.

[27] Li T, Zhao Z, Sun C, Yan R, Chen X. Hierarchical attention graph convolutional network to fuse multi-sensor signals for Remaining Useful Life prediction. Reliab Eng Syst Saf 2021;215:107878.

[28] Al-Dulaimi A, Zabihi S, Asif A, Mohammadi A. A multimodal and hybrid deep neural network model for Remaining Useful Life estimation. Comput Industry 2019;108:186–96.

[29] Ellefsen AL, Bjø rlykhaug E, Æ søy V, Ushakov S, Zhang H. Remaining Useful Life predictions for turbofan engine degradation using semi-supervised deep architecture. Reliab Eng Syst Saf 2019;183:240–51.

[30] Ruiz-Tagle Palazuelos A, Droguett EL, Pascual R. A novel deep capsule neural network for remaining useful life estimation. Proc Inst Mech Eng Part O: J Risk Reliab 2020;234(1):151–67.

[31] Zhao Y, Wang Y. Remaining Useful Life prediction for multi-sensor systems using a novel end-to-end deep-learning method. Measurement 2021;109685.

[32] Ackert SP. Basics of aircraft maintenance programs for financiers. Eval Insights Commerc Aircraft Mainten Programs 2010;1–23.

[33] Kramer O. Genetic algorithms. In: Genetic algorithm essentials. Springer; 2017, p. 11–9.

[34] Saxena A, Celaya J, Saha B, Saha S, Goebel K. On applying the prognostic performance metrics. In: Annual conference of the PHM society, vol. 1; 2009.