

Aircraft engine maintenance planning using model-based remaining useful life prognostics

A Master of Science Thesis

N. Schiettekatte



Aircraft engine maintenance planning using model-based remaining useful life prognostics

A Master of Science Thesis

by

N. Schiettekatte

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday February 10th, 2022 at 14:00.

Student number: 4485610
Project duration: 22/02/2021 - 10/02/2022
Thesis committee: Prof. Dr. H.A.P. Blom TU Delft, Chair
Dr. M.A. Mitici TU Delft, Supervisor
Dr. D. Zappalá TU Delft, External committee member

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

The front page image is taken from <https://ammroc.edggroup.ae/>

Acknowledgements

This thesis concludes my studies and it is my final deliverable to obtain a MSc degree in Aerospace Engineering at Delft University of Technology. In this thesis you will find my research about predictive maintenance and how this can be applied for aircraft turbofan engines. I greatly enjoyed working on this research in the past year where I was able to combine my interests in modelling, statistics and aircraft. The research was a challenge where I learned many new things about myself, the research topic and I improved my academic skills. The thesis is a great way to conclude six and a half years of studying in Delft.

I would like to sincerely thank my supervisor Mihaela Mitici, who helped me a lot during this research. We had many interesting and inspiring discussions which were of great help for improving my research. She was supportive and always provided constructive feedback, which improved the quality of my research significantly.

I would also like to thank my friends and family for their support through this journey. My girlfriend, Nienke, who was always there for me and ready to listen and support me. I would like to thank my family for their trust and believing in me and finally my friends, who offered great distraction during our coffee and lunch breaks and for entertainment and support during the weekends and evenings. This research wouldn't have been possible without all of you.

*N. Schiettekatte
Delft, January 2022*

Contents

List of Figures	vii
List of Tables	ix
Nomenclature	xi
Introduction	xvii
I Scientific Paper	1
II Literature Study previously graded under AE4020	27
1 Abstract	29
2 Introduction	31
3 Literature Review	33
3.1 Prognostics in Aircraft Maintenance	33
3.1.1 Current Situation	33
3.1.2 A Brief Revision of Particle Filter Theory	34
3.1.3 Particle Filter Literature Review	35
3.1.4 A Brief Revision of Polynomial Chaos Expansion Theory	41
3.1.5 Polynomial Chaos Expansion Theory Literature Review	42
3.2 Degradation Modelling of Multi-Sensor Systems	46
3.2.1 Introduction to Data	47
3.2.2 Current Solution Methods	47
3.3 Maintenance Optimisation Methods under Uncertainty	52
3.3.1 Stochastic Optimisation Methods	52
3.3.2 Markov Decision Process Optimisation Methods	56
4 Research Framework	61
4.1 Research Gap	61
4.2 Research Aim	63
4.3 Research Questions	63
4.4 Research Scope	64
5 Thesis Planning	65
5.1 Gantt Chart	65
6 Conclusion	67
III Supporting work	69
1 Additional results: health indicator modelling & RUL prognostics	71
1.1 PHM Challenge data set	71
1.1.1 Sensor measurements split per operational mode	71
1.1.2 Principal component analysis	74
1.1.3 Failure clusters per operational mode	74
1.1.4 Health indicator progression example	76
1.1.5 RUL estimation example	77
1.1.6 Results aPCE model	77

1.2	FD001 data set	78
1.2.1	Sensor measurements	78
1.2.2	Principal component analysis	78
1.2.3	Failure clusters.	79
1.2.4	Health indicator progression example	79
1.2.5	RUL estimation example	80
1.2.6	Results aPCE model	80
1.3	FD002 data set	81
1.3.1	Sensor measurements split per operational mode	81
1.3.2	Principal component analysis	84
1.3.3	Failure clusters per operational mode	85
1.3.4	Health indicator progression example	86
1.3.5	RUL estimation example	87
1.3.6	Results aPCE model	87
2	Additional results: component level optimisation	89
3	Additional results: full case study results	99
4	Verification and Validation	101
4.1	Model verification.	101
4.2	Model validation	102
	Bibliography	105

List of Figures

2.1	Academic publications per year in the field of prognostics[25]	31
3.1	Different maintenance strategies with their corresponding relative cost indication and number of failures[52]	34
3.2	Overview of the PF process [1]	35
3.3	Relation between states and observations in a HMM framework	39
3.4	Possible predictions of the oil flow depending on the model order [30]	39
3.5	Estimation of the UtUV parameter using regular PF (left) and a fuzzy resampling PF (right) [25]	40
3.6	Similarity-based matching of degradation trajectories in a library to test data [57]	49
3.7	Failure space consisting of principal components with an indication of the barycenter of a certain operational mode [24]	50
3.8	PoF and risk of unavailability second engine if first engine is being repaired [4]	54
3.9	Sequential decision problem [44]	56
3.10	Maintenance policy map where A defines the action to be taken [31]	58
5.1	Project Gantt chart	66
1.1	Sensor measurements of all sensors - PHM Challenge training data set operational mode 1.	71
1.2	Sensor measurements of all sensors - PHM Challenge training data set operational mode 2.	72
1.3	Sensor measurements of all sensors - PHM Challenge training data set operational mode 3.	72
1.4	Sensor measurements of all sensors - PHM Challenge training data set operational mode 4.	73
1.5	Sensor measurements of all sensors - PHM Challenge training data set operational mode 5.	73
1.6	Sensor measurements of all sensors - PHM Challenge training data set operational mode 6.	74
1.7	Engines failing per operational mode projected in the 2D principal component space for the PHM Challenge training data set. 7 sensor measurements are used to develop the 2D space, see Part I.	75
1.8	Projection of all failure clusters identified in Figure 1.7 in the principal component space.	76
1.9	Constructed health indicator for the first 2 engines of the PHM challenge training data set.	76
1.10	RUL estimation for engines 1 and 2 of the PHM Challenge testing data set	77
1.11	Sensor measurements of all sensors - FD001 training data set operational mode 1.	78
1.12	Projection of the failure cluster in the principal component space using the FD001 training data set.	79
1.13	Constructed health indicator for the first 2 engines of the FD001 training data set.	79
1.14	RUL estimation for engines 1 and 2 of the FD001 testing data set	80
1.15	Sensor measurements of all sensors - FD002 training data set operational mode 1.	81
1.16	Sensor measurements of all sensors - FD002 training data set operational mode 2.	81
1.17	Sensor measurements of all sensors - FD002 training data set operational mode 3.	82
1.18	Sensor measurements of all sensors - FD002 training data set operational mode 4.	82
1.19	Sensor measurements of all sensors - FD002 training data set operational mode 5.	83
1.20	Sensor measurements of all sensors - FD002 training data set operational mode 6.	83
1.21	Engines failing per operational mode projected in the 2D principal component space for the FD002 training data set. 7 sensor measurements are used to develop the 2D space, see Part I.	85
1.22	Projection of all failure clusters identified in Figure 1.21 in the principal component space.	86
1.23	Constructed health indicator for the first 2 engines of the FD002 challenge training data set.	86
1.24	RUL estimation for engines 1 and 2 of the FD002 testing data set	87
4.1	RUL prediction errors for the FD002 data set.	102

4.2	Sensitivity analysis results by Nguyen and Medjaher [35]. CR is short for cost rate, DPM for dynamic predictive maintenance, PeM for periodic maintenance and IPM ideally predicted maintenance.	103
4.3	Sensitivity analysis results by Zhang and Zhang [62] by varying the cost of corrective repair C_c and cost of downtime C_d . DCBM is short for dynamic condition based maintenance, PM for periodic maintenance and IPM for ideally predicted maintenance.	104

List of Tables

3.1	Literature Review on Particle Filters	35
3.2	Literature Review on Polynomial Chaos Expansions	42
3.3	Properties of the polynomial basis of the moment-based aPC method[37]	44
3.4	Literature review on degradation modelling	48
3.5	Evaluation of different stochastic and statistical approaches on the 2008 PHM data set [29]	52
3.6	Literature review on maintenance optimisation methods	53
5.1	Thesis planning of milestones	65
1.1	Explained variance of the principal components of the PHM Challenge training data set.	74
1.2	Results of the aPCE prognostic model applied to the first 12 aircraft engines of the PHM Challenge testing data set.	77
1.3	Explained variance of the principal components of the FD001 training data set.	78
1.4	Results of the aPCE prognostic model applied to the first 12 aircraft engines of the FD001 testing data set.	80
1.5	Explained variance of the principal components of the FD002 training data set.	84
1.6	Results of the aPCE prognostic model applied to the first 12 aircraft engines of the FD002 testing data set.	87
2.1	Optimal aircraft engine maintenance times for engines 201 to 203 from the FD002 training data set	89
2.2	Optimal aircraft engine maintenance times for engines 204 to 209 from the FD002 training data set	90
2.3	Optimal aircraft engine maintenance times for engines 210 to 216 from the FD002 training data set	91
2.4	Optimal aircraft engine maintenance times for engines 217 to 223 from the FD002 training data set	92
2.5	Optimal aircraft engine maintenance times for engines 224 to 230 from the FD002 training data set	93
2.6	Optimal aircraft engine maintenance times for engines 231 to 237 from the FD002 training data set	94
2.7	Optimal aircraft engine maintenance times for engines 238 to 244 from the FD002 training data set	95
2.8	Optimal aircraft engine maintenance times for engines 245 to 250 from the FD002 training data set	96
2.9	Optimal aircraft engine maintenance times for engines 251 to 256 from the FD002 training data set	97
2.10	Optimal aircraft engine maintenance times for engines 257 to 260 from the FD002 training data set	98
3.1	Optimised maintenance schedule for a pool of aircraft engines for 100 days.	99

Nomenclature

List of Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
AOG	Aircraft On Ground
AOPF	Adaptive Order Particle Filter
aPC	Arbitrary Polynomial Chaos
ASIS	Auxiliary Sampling Importance Sampling
CBM	Condition-Based Maintenance
CDF	Cumulative Distribution Function
CI	Confidence Interval
CM	Corrective Maintenance
CRPS	Continuous Ranked Probability Score
DBN	Deep Belief Network
DM	Do Maintenance
DN	Do Nothing
ELM	Extreme Learning Machine
FC	Flight Cycle
FPCA	Functional Principal Component Analysis
gPC	Generalised Polynomial Chaos
HI	Health Indicator
HMM	Hidden Markov Model
HPC	High Pressure Compressor
IS	Importance Sampling
JIT	Just-In-Time
KF	Kalman Filter
LLP	Life Limit Part
MB	Model-Based
MC	Monte Carlo
MDP	Markov Decision Process

MFPCA	Multivariate Functional Principal Component Analysis
MLE	Maximum Likelihood Estimation
MLP	Multi-Layer Perceptron
MRO	Maintenance Repair & Overhaul
MSDFM	Multi-Sensor Data Fusion Model
MTTF	Mean-Time-To-Failure
MU	Monetary Units
NASA	National Aeronautics and Space Administration
OM	On-condition Maintenance
OM	Operational Mode
OM	Opportunistic Maintenance
PC	Principal Component
PCA	Principal Component Analysis
PCE	Polynomial Chaos Expansion
PCM	Probabilistic Collocation Method
PCoE	Prognostics Centre of Excellence
PDF	Probability Density Function
PF	Particle Filter
PHM	Prognostics and Health Management
PM	Preventive Maintenance
PM-MTTF	Periodic Maintenance - Mean-Time-To-Failure
RBF	Radial Basis Function
RMSE	Root Mean Squared Error
RTF	Run-To-Failure
RUL	Remaining Useful Life
SIR	Sequential Importance Resampling
SMC	Sequential Monte Carlo
SO	Stochastic Optimisation
SSM	State Space Model
TBM	Time-Based Maintenance
TTM	Time-To-Maintenance
UQ	Uncertainty Quantification
UtUV	Unit to Unit Variability

List of Symbols

α	Matrix storing the combinatoric enumeration information of the multivariate PCE [-]
α_{early}	Penalty for preponing maintenance [-]
α_{fail}	Engine failure penalty for the MDP [-]
α_{late}	Penalty for postponing maintenance [-]
ϵ_i	RUL prediction error for engine i [FC]
ϵ_i^m	MDP maintenance planning error for engine i [FC]
γ	Discount factor [-]
Λ	Support of a random variable ξ [-]
$\mathbb{D}(x)$	Distribution of a parameter x [-]
$\mu_{\bar{k}}$	\bar{k}^{th} statistical moment [-]
Φ_i	Multivariate orthogonal polynomial basis [-]
σ_i^p	Standard deviation of the RUL prediction [-]
ξ	Model parameter [-]
A	Set of actions [-]
a	Action [-]
a^*	Optimal maintenance action corresponding to time t^* [-]
C	Costs of doing maintenance at a potential time t [MU]
C^*	Costs corresponding to a^* and t^* [MU]
C_f	Costs for corrective maintenance [MU]
C_m	Costs for preventive maintenance [MU]
C_r	Costs associated to risk [MU]
C_{AOG}	Costs for an AOG situation [MU]
C_{lp}	Costs of LP problem [MU]
C_{PerfM}	Costs of the perfect RUL maintenance strategy [MU]
C_{PM}	Costs of the PM-MTTF maintenance strategy [MU]
$C_{r,dn}$	Costs associated to risk of doing nothing [MU]
$C_{r,p}$	Costs associated to risk of performing maintenance [MU]
C_{RTF}	Costs of the run-to-failure maintenance strategy [MU]
C_{RUL-M}	Costs of the RUL-maintenance strategy [MU]
C_{wp}	Costs associated to wasting useful engine life cycles [MU]
$c_{i,t}$	Costs of planning maintenance of engine i on time t for LP formulation [MU]
D_f	Failure threshold [-]
D_k	Dispersion of failure mode k [-]

D_s	Set of failure thresholds [-]
\bar{d}	Order of expansion of the PCE model [-]
$d_{i,t}^k$	Distance of point a point in the PC space to the failure centre [-]
$d_{i,t}^{k(t)}$	Distance of engine i at time t for operational mode k corresponding to time t [-]
E	Number of operational engines at a time [-]
$f(t, \xi)$	Function to model the health indicator as a function of input ξ [-]
$f_i(t)$	Time dependent coefficients of the PCE [-]
$g(\xi)$	Time independent coefficients [-]
$H^m(t)$	Model function of the health indicator [-]
$H^{PCE,P}$	Posterior response surface PDF of the PCE [-]
H^{PCE}	PCE model response of the health indicator model [-]
$H_i(t)$	Health indicator of engine i at time t [-]
$\bar{H}^{PCE,P}$	Average of the posterior response surface PDF of the HI PDF $H^{PCE,P}$ [-]
K	Total number of operational modes [-]
M	Expansion terms of the PCE [-]
\mathcal{M}	Markov Decision Process [-]
N	Number of aircraft engines [-]
N_k	Number of engines failing in operational mode k [-]
N_s	Number of states [-]
n	Number of independent random variables [-]
n_i^p	Sample size of the PCE posterior response surface [-]
P	State transition probabilities [-]
$\hat{p}^{(\bar{k})}$	Orthonormal polynomial of degree \bar{k} [-]
$P^i(\xi)$	Orthogonal polynomial basis of the PCE of degree i [-]
P_t	Maintenance planning horizon [days]
p_{early}^i	Costs for preponing maintenance [MU]
p_{late}^i	Costs for postponing maintenance [MU]
$p_i^{(\bar{k})}$	Orthogonal polynomial coefficient of polynomial $P^{(\bar{k})}$ of degree \bar{k} [-]
Q	Hangar availability [-]
R	Reward function [-]
R_v	Variance matrix [-]
RUL_i	Predicted remaining useful life of engine i [FC]
RUL_i^a	Actual remaining useful life of engine i [FC]
S	State space [-]

S_i	PHM challenge score of engine i [-]
s_t	State corresponding to time t [-]
T	Set of potential maintenance dates [day]
\bar{T}_f	Mean-time-to-failure [FC] or [day]
T_{hor}	Maximum simulation time [days]
T_i	Failure time of an engine i [FC] or [day]
$T_{PM,i}$	Scheduled maintenance date for the MTTF-PM strategy [day]
T_{PM}	Periodic maintenance time [FC] or [day]
$t_{post,i}$	New maintenance date in case of an AOG event [day]
$t_{pre,i}$	Preponed maintenance date for perfect RUL maintenance strategy [day]
t	Time [FC] or [day]
t^*	Optimal maintenance time [FC] or [day]
t_m	Maintenance duration [days]
t_{ins}	Time between RUL prognostic updates/inspections [days]
V	Value of the Bellman equation [-]
W	Number of flights per week [-]
w_i	Sample weight [-]
$w_{i,t}$	LP decision variable for planning engine i on day t [-]
\bar{x}^k	x-coordinate of the failure centre in mode k [-]
x_{i,T_i}^k	x-coordinate in the PC space of engine i failing in mode k at time T_i [-]
$x_{i,t}^k$	x-coordinate of sensor measurement data vector $z_{i,t}^k$ in the PCA space [-]
Y	Model output [-]
\bar{y}^k	y-coordinate of the failure centre in mode k [-]
\mathbf{y}	Vector containing health indicator measurements [-]
y_{i,T_i}^k	y-coordinate in the PC space of engine i failing in mode k at time T_i [-]
$y_{i,t}^k$	y-coordinate of sensor measurement data vector $z_{i,t}^k$ in the PCA space [-]
y_m	Model output [-]
Z^k	Matrix containing all sensor measurements of engines failing in operational mode k [-]
z_{i,T_i}^k	Sensor measurement data vector of engine j at time T_i failing in mode k [-]
$z_{i,t}^k$	Sensor measurement data vector of engine i at time t in operational mode k [-]

Introduction

Aircraft maintenance has been required since the beginning of the aviation era. Historically seen, aircraft maintenance repair & overhaul companies have had a rather conservative approach towards maintenance as high safety standards must be adhered to, meaning that components are often repaired or replaced before their end of life is reached. In recent years, the interest of maintenance companies is shifting towards implementing prognostics & health management (PHM) methods which make full use of components, leading to less costs, less unexpected aircraft-on-ground events and a significant reduction of waste of components [15].

This research focuses on developing a new prognostic method for aircraft turbofan engines based on run-to-failure data. This method is then used to implement it in a predictive maintenance framework. A thorough literature study of state-of-the-art research is performed during the starting phase of this thesis to identify research gaps and formulate the research aim and research questions. 2 aims have been identified for this research:

1. Developing a model-based prognostic model which is able to estimate the remaining useful life of an aircraft engine based on implicit multi-sensor measurements obtained during the lifetime of that engine, as well as quantify the uncertainty of the remaining useful life estimation in the form of a probability density function.
2. Developing a maintenance optimisation method which is able to implement the result of the prognostic model (the probability density function of the remaining useful life of an aircraft engine) in order to obtain an optimal maintenance policy which reduces maintenance costs compared to other maintenance policies.

This research contributes to the development of new prognostic and maintenance optimisation methods in the aircraft maintenance industry by providing new insights and a proof of concept of the developed approach. Furthermore, whereas most research either focuses on either developing a new prognostic method or developing a new maintenance optimisation framework, this research aims at combining the two using a new approach.

This thesis report is organised as follows. In [Part I](#) the scientific paper is presented. Next, in [Part II](#) the literature study is presented, which has been graded under the course code AE4020. Finally, in [Part III](#) the supporting work of the thesis is provided.

I

Scientific Paper

Aircraft engine maintenance planning using model-based remaining useful life prognostics

Niels Schiettekatte*

Delft University of Technology, Delft, The Netherlands

January 2022

Abstract

Aircraft maintenance methods are shifting from conservative maintenance approaches such as a periodic maintenance approach towards predictive maintenance approaches, leading to a reduction of costs, less unexpected aircraft-on-ground events and less wasted useful life of components. In this paper, we propose a new remaining useful life prognostics approach for aircraft engines and integrate this into a maintenance planning framework. First, an explicit health indicator is constructed from an implicit multi-sensor aircraft engine degradation data set using principal component analysis. Then, the remaining useful life prognostic model is developed using a polynomial chaos expansion approach, which allows for uncertainty quantification in the form of a probability density function which is faster than Monte Carlo simulation. A Markov decision process is used to determine the optimal time of maintenance for aircraft engines. During a case study, these optimal times for individual engines being part of a pool of operational engines are integrated using a linear programming model and a rolling horizon approach to obtain an optimised maintenance schedule. The prognostic model performs in the mid-range compared to other papers using the same data sets. Furthermore, with the current cost parameters the integrated remaining useful life maintenance planning strategy reduces costs by a factor of 3 and 2.5 compared to a periodic maintenance strategy or a run-to-failure maintenance strategy, respectively. The waste is reduced by a factor of 2.5 compared to periodic maintenance, while no failures occur. This research has demonstrated that the polynomial chaos expansion remaining useful life prognostic approach can be used for optimal maintenance planning for aircraft engines using a Markov decision process, showing benefits in terms of costs, waste and unexpected failures.

Keywords: Remaining useful life prognostics, Polynomial Chaos Expansion, Aircraft Turbofan Engines, Predictive Maintenance

1 Introduction

Aircraft maintenance has been a necessity since the beginning of the aviation era. It is required to ensure that the airworthiness of aircraft is preserved and that requirements of system reliability, availability, maintainability and safety are met at all times. Historically seen, maintenance repair and overhaul (MRO) companies have had a rather conservative attitude regarding maintenance scheduling during the last century and the current first decade of the current century [1]. During this time frame, components were often replaced at fixed intervals without taking into account the degradation state of the component[2]. Currently, MRO's are continuously trying to optimise their operations and because condition monitoring and modelling techniques have improved significantly in the last decades, the interest of MRO companies is shifting from conservative methods to the investigation of implementing prognostic methods in order to optimise their operations [3].

With the increasing availability of big data and operational experience of airlines, MRO companies and aircraft manufacturers, more and more diagnostics and prognostics methods are developed to predict remaining lifetimes of aircraft components. Using diagnostics and prognostics, optimal aircraft maintenance dates can be dynamically predicted instead of using periodic maintenance inspections, leading to less air-

craft on ground (AOG) time due to unexpected faults, avoiding early replacements leading to less waste and a significant reduction in costs.

Three different main maintenance strategies can currently be identified: preventive maintenance, predictive maintenance and corrective maintenance [4]. Preventive maintenance is associated with low repair costs but high preventive costs due to waste. On the other hand, corrective maintenance is associated with high repair costs due to unexpected faults. Predictive maintenance tries to balance the two by minimising costs in terms of waste and unexpected failures. Prognostics and health management (PHM) is a term often referred to in literature which describes the discipline linking the remaining useful life (RUL) of a component to appropriate decision making to maintain the component in time[5]. In this paper we will develop a prognostic method to determine the RUL of aircraft turbofan engines and use this to develop an optimal maintenance schedule by applying predictive maintenance.

Prognostic models can be classified in three types of models: data-driven, model-based and hybrid [6]. Model-based methods depend on the incorporation of a physical model to estimate the RUL, such as the physical crack growth model by Paris and Erdogan [7]. Data-driven methods do not rely on any knowledge of the physical behaviour of the component which is analysed. They depend largely on measured data. Furthermore, data-driven models can be split into models using artificial intelligence (AI) and models which use statistical

*MSc Student, Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology

and stochastic approaches. Finally, hybrid models combine model-based and data-driven models in order to get the best of both worlds [6].

Many prognostics methods can be established for determining the RUL of a component using information about the state of degradation of the component. Although AI methods are currently prevailing in state-of-the-art literature, we decide to have a model-based approach. The reason for this is that AI models are essentially black-box models with little transparency and high computational times whereas a model-based approach is better at representing the process of creating a model and follow the steps taken because all statistical and mathematical relations are known. This facilitates greater support for decision making and allows for more support of adopting the model by other users [8]. Similarity based approaches have been used in [9]. Weibull distribution prognostics is performed in [10]. Wiener process modelling in combination with Monte Carlo simulation has been done by the authors of [11] and [12]. (Non) linear regression methods can also be used, this has been demonstrated in [13]. Another frequently used methods are Kalman Filtering or Particle Filtering (PF) [14]. In [15] it has been demonstrated that Polynomial Chaos Expansions (PCE) can also be used for RUL determination in prognostics in the case of degrading batteries. Polynomial chaos expansion (PCE) is a non-sampling method to determine the evolution of uncertainty of a certain system with probabilistic uncertainty in the system input parameters. In literature, PCE is often used for models which require very high computational effort to run. Instead, the model can be represented by a set of simpler equations, polynomials, which are faster to evaluate than the original system. In [15] it is demonstrated that PCE has a significant reduction in computational time while preserving the same order of accuracy compared to Monte Carlo methods. In [16] it is stated that the synergy of applying PCE and principal component analysis (PCA) is researched scarcely but might lead to a significant decrease in computational effort while preserving most of the information of a data set.

Regarding maintenance planning, different models have been developed in literature. Threshold-based maintenance strategies have been developed in [17] and [18]. Here, a maintenance action is planned or performed if a component reaches a certain threshold such as a probability of failure. [19] plan maintenance if a certain degradation threshold is reached. Genetic algorithms are also used for stochastic maintenance optimisation and dynamic opportunistic maintenance planning [20, 21]. Another frequently used method for maintenance scheduling is to use a linear programming formulation. In [22] the authors use a linear programming formulation to determine the optimal maintenance schedule for aircraft cooling units. Finally, a Markov decision process (MDP) is also frequently used to plan maintenance. The authors of [23] develop a MDP model to find the optimal policy for aircraft engines based on degradation (dependent on the exhaust gas temperature) and the life limit part. Another MDP is formulated in [24], where a set of missions and maintenance opportunities is defined at first and the optimal sequence of these is determined.

In this paper, we will develop a model-based polynomial chaos expansion (PCE) RUL prognostic model which also allows for uncertainty quantification. We will use PCA to construct an engine health indicator in a similar fashion as was done in [11]. Next to this, we will split our maintenance optimisation model in two phases. Phase 1 is optimisation on the component level using a MDP, which will give optimal replacement times for engines. Phase two involves a

linear programming model which uses the optimal replacement times of individual engines to create a feasible schedule based on hangar availability. A case study is set up which implements all models and where a rolling horizon approach will be used. The performance of the model will be evaluated by comparing the case study with three other maintenance strategies: run-to-failure, periodic and perfect maintenance.

The main contributions of this paper can be summarised as follows:

- A new model-based prognostic approach is developed for aircraft engine maintenance, which involves a combination of PCA and PCE and allows for uncertainty quantification over time.
- We propose a new predictive maintenance framework which uses a model-based prognostic model in combination with a MDP for optimal maintenance planning for a pool of aircraft engines and demonstrate the effectiveness by comparing it to other maintenance strategies.

The paper is structured as follows. In section 2 we develop a health indicator and the model-based prognostic model for aircraft engines using the arbitrary PCE method. Then, in section 3 we provide the results of the prognostic model and do a benchmark with models found in literature using the same data set. In section 4 we present the maintenance optimisation model using a MDP on the component level and a linear programming formulation on the system level. In section 5 we introduce the case study involving the optimisation for a pool of engines operating in a fleet of aircraft. The results of the case study are compared with other maintenance strategies in section 6. Finally, a conclusion and recommendations are given in section 7.

2 RUL prognostics for aircraft engines

In this section we provide a method to develop a health indicator (HI) based on the available sensor measurements. Then, the health indicator is used to develop the prognostic model using a polynomial chaos expansion approach for aircraft engines.

2.1 Description of the sensor measurements for aircraft engines

The data set that we will use for this research is provided by NASA's Prognostics Centre of Excellence. NASA provides an aircraft turbofan engine degradation data set which consists of a flight cycle (FC) time series containing engine data. For each FC, a random snapshot during cruise is taken for 21 sensor measurements which correspond to engine temperature and pressure at different locations, engine fan and core speed etc. Three different operational settings are defined, resulting in either 1 or 6 different operational modes. A data set consists of a training data set containing run-to-failure data for training of the model and a test set to test the developed model. Figure 1 shows an example of the progression of sensor measurements of all engines in the PHM Challenge data set for operational mode 1.

2.2 Constructing a health indicator using principal component analysis (PCA)

We construct a health indicator using principal component analysis (PCA), which relies on the correlation between sensor measurements. PCA is a commonly used method to reduce the dimensionality of a data set while the variance of

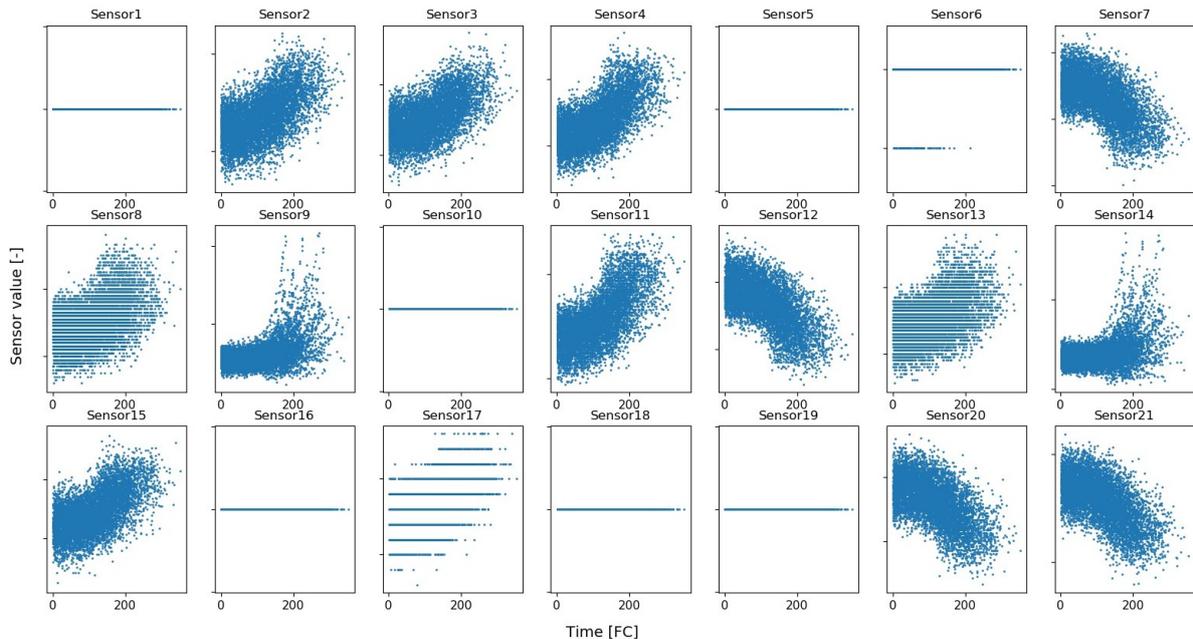


Figure 1: Sensor measurements of all engines in operational mode 1 using the PHM challenge data set.

Table 1: PCoE turbofan engine degradation data sets

Data set	Fault Modes	Operational Modes	Training Units	Testing Units
FD001	1	1	100	100
FD002	1	6	260	259
FD003	2	1	100	100
FD004	2	6	249	248
PHM Challenge	1	6	218	218

the data set is maintained. PCA has been used in, for instance, [11], [10] to determine a health indicator for aircraft engines and we will follow their approach as well.

We obtain the engine health indicator using the PHM challenge, FD001 and FD002 data sets. As there are several operational modes in the PHM Challenge and FD002 data sets, the data is split per operational mode. This results in 6 different sub data sets, one for each operational mode. The FD001 data set contains only 1 operational mode. Figure 1 shows an example of sensor measurements for the PHM challenge data set in operational mode 1.

Sensor measurement selection

Following [9, 11, 13], we select sensors sensors 2, 3, 4, 7, 11, 12 and 15, which are shown to have a monotonically increasing or decreasing trend and which have shown to be the optimal sensors for RUL estimation. These sensors are selected for both the PHM challenge, FD001 and FD002 data sets.

Space definition for PCA

Let N denote the total number of aircraft engines N in the considered data set (PHM Challenge, FD001, FD002 training data sets). Let K denote the number of operational models in a data set, with $K = 6$ for PHM challenge and FD002, and $K = 1$ for FD001. Throughout a flight cycle, the engine is in one of the K modes. Let $z_{i,t}^k$ denote a data vector consisting of 7 sensor measurements of engine $i \in \{1, 2, \dots, N\}$ in operational mode $k \in \{1, \dots, K\}$ at time $t > 0$. Let z_{i,T_i}^k

denote the vector of sensor measurements at the moment of failure of engine i , $i \in \{1, 2, \dots, N\}$ at time T_i and $|z_{i,T_i}^k| = 7$. Data vectors $\{z_{i,T_i}^k\}$, $i \in \{1, 2, \dots, N\}$ are grouped according to their operational mode. This results in K groups of engines failing in each of the K modes. PCA is applied for each of these K groups. Let N_k , $N_k \leq N$, denote the total number of engines failing in mode k . Let Z^k denote the matrix of sensor measurements at the moment of failure for all engines failing in mode k and $|Z^k| = N_k \times 7$, i.e.,

$$Z^k = \begin{bmatrix} z_{j,T_j}^k \end{bmatrix}, \quad (1)$$

where j , $j \in \{1, 2, \dots, N\}$, is an engine in mode k failing at time T_j . For this matrix, the covariance matrix is determined as:

$$Cov = \left(Z^k \right)^T Z^k \quad (2)$$

The eigen vectors and eigen values are determined from the covariance matrix. The eigen values are sorted in descending order and the eigen vectors are sorted accordingly. The contribution of an eigen vector, a principal component, is calculated by taking its eigen value and dividing this by the sum of all eigen values. The explained variance for all operational modes can then be found in Table 2. The first two eigen vectors capture 98 - 99% of the variance in the data set. Thus, for our analysis we consider only the first two eigen vectors (principal components). Each principal component has size 7, e.g. $|PC1| = |PC2| = 1 \times 7$.

Table 2: Explained variance of the principal components as a percentage of the total variance - PHM Challenge data set.

	Mode1	Mode2	Mode3	Mode4	Mode5	Mode6
PC1	60.8	59.0	79.7	72.6	61.5	54.4
PC2	38.0	40.1	19.1	26.7	37.8	44.6
PC3	0.7	0.4	0.8	0.3	0.3	0.6
PC4-7	0.5	0.5	1.2	0.4	0.4	0.4

We next project the matrices Z^k for $k \in \{1, \dots, K\}$ onto the selected 2 principal components by applying the dot

product between z_{j,T_j}^k for each $k \in \{1, \dots, K\}$ and for each $j \in \{1, \dots, N\}$, i.e.,

$$x_{j,T_j}^k, y_{j,T_j}^k = \langle [PC1_k, PC2_k]^T, z_{j,T_j}^k \rangle, \quad (3)$$

where x_{j,T_j}^k and y_{j,T_j}^k are the x- and y-coordinate of data point z_{j,T_j}^k in the obtained 2-dimensional principal component space. Considering all points z_{j,T_j}^k , we obtain K clusters of failure points in the principal component space. This can be viewed in Figure 2. Each point in the principal component space contains all failure points of that specific operational mode.

Finally, the failure centre of an operational mode k , $k \in \{1, 2, \dots, K\}$ is defined as the weighted average of all points x_{j,T_j}^k and y_{j,T_j}^k , $j \in \{1, 2, \dots, N\}$:

$$(\bar{x}^k, \bar{y}^k) = \left(\frac{1}{N_k} \sum_j x_{j,T_j}^k, \frac{1}{N_k} \sum_j y_{j,T_j}^k \right). \quad (4)$$

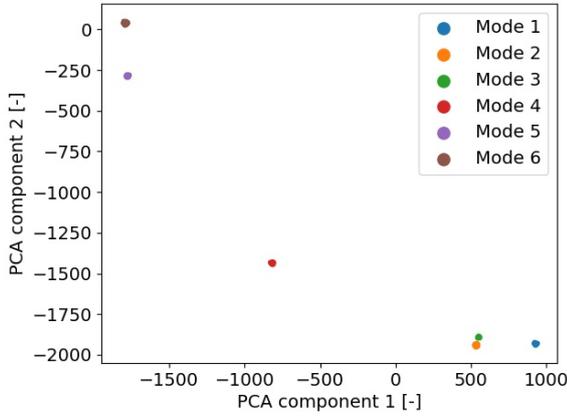


Figure 2: Principal component space with the projection of all failure points using the PHM challenge data set.

Health indicator as a distance metric

Using the K clusters (failure centers), we construct a health indicator for $t < T_i$. A data point $z_{i,t}^k$, $i \in \{1, 2, \dots, N\}$, $k \in \{1, 2, \dots, K\}$ is projected into the obtained principal component space (see Figure 2) by applying the dot product between the first two principal components (PC) of the operational mode k ($[PC1, PC2]$ = 2×7) and the data point $z_{i,t}^k$ ($|z_{i,t}^k| = 1 \times 7$) as follows:

$$x_{i,t}^k, y_{i,t}^k = \langle [PC1_k, PC2_k]^T, z_{i,t}^k \rangle, \quad (5)$$

where $x_{i,t}^k$ and $y_{i,t}^k$ are the x- and y-coordinate of data point $z_{i,t}^k$ in the 2-dimensional principal component space.

A health indicator for engine i , $i \in \{1, \dots, N\}$ at time t in operational mode k , $k \in \{1, 2, \dots, K\}$ is then constructed as the distance $d_{i,t}^k$ between the failure centre (\bar{x}^k, \bar{y}^k) of mode k and the projection of the data point $(x_{i,t}^k, y_{i,t}^k)$ in the principal component space, i.e. [11]:

$$d_{i,t}^k = \frac{\sqrt{(x_{i,t}^k - \bar{x}^k)^2 + (y_{i,t}^k - \bar{y}^k)^2}}{D_k}, \quad (6)$$

with

$$D_k = \sqrt{\frac{1}{N_k - 1} \sum_i \left((x_{i,T_i}^k - \bar{x}^k)^2 + (y_{i,T_i}^k - \bar{y}^k)^2 \right)} \quad (7)$$

the dispersion of failure mode k .

A visualisation of a failure centre, failure points and projected can be found in Figure 3.

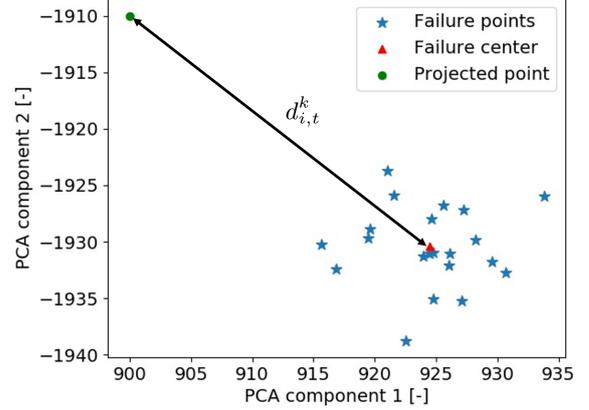


Figure 3: Engine failure points, failure centre and a projected data point in the principal component space - PHM challenge data set, the failure centre corresponds to operational Mode 1.

For an engine i , $i \in \{1, \dots, N\}$, at each time $0 < t < T_i$, a health indicator is now given by:

$$\{d_{i,t}^{k(t)}\}, t \in \{1, \dots, T_i\} \quad (8)$$

where $k(t)$ refers to the operational mode k in which an engine can be in at time t .

Figure 4 shows an example of a health indicator. The distance $d_{i,t}^k(t)$ gradually becomes smaller as time progresses, indicating a decreasing trend for the health indicator.

The procedure outlined in the previous section applied for engines from the training data sets. The health indicator for engines from the testing data set is constructed in a similar way, but the difference being that we do not have to identify the failure centres. Instead, we use the found principal components and failure centres from the training data set and apply Equation 5, Equation 6 and Equation 8 for engines in the testing data set. This results in a health indicator of an engine until the time that the last set of sensor measurements has become available.

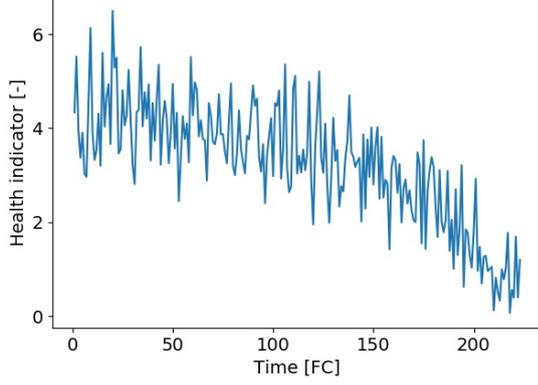
2.3 RUL prognostics using Polynomial Chaos Expansions

In this section we propose a prognostic model to determine the remaining useful life of aircraft engines using the arbitrary polynomial chaos expansion (aPCE) method. The aPCE relies on the fact that an orthogonal polynomial can be constructed from the statistical moments of the underlying distribution of an uncertain input parameter [25, 26].

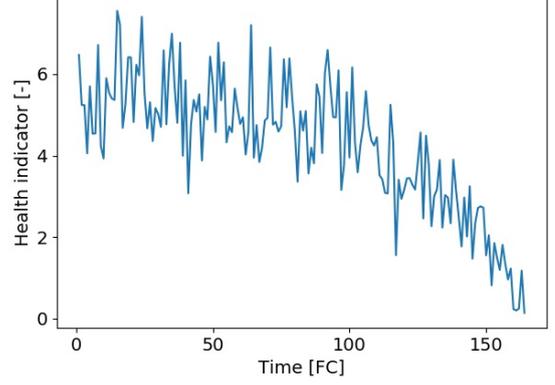
Based on the health indicator $\{d_{i,t}^{k(t)}\}$, $t \in \{1, \dots, T_i\}$, $k \in \{1, \dots, K\}$, for an engine i , we fit a model:

$$H^m(t) = f(t, \boldsymbol{\xi}), \quad (9)$$

with $|\boldsymbol{\xi}| = n$, with $\mathbb{D}(\xi_j)$, $j \in \{1, \dots, n\}$ the distribution of parameter ξ_j and $\mathbf{D} = \prod_{j=1}^n \mathbb{D}(\xi_j)$ the distributions for all ξ_j , $j \in \{1, \dots, n\}$. The support of $\boldsymbol{\xi}$ is $\boldsymbol{\Gamma} = \prod_{j=1}^n \Gamma_j$.



(a) Health indicator of engine 1.



(b) Health indicator of engine 2.

Figure 4: Health indicator progression over time of engines 1 and 2 from the PHM challenge training data set.

2.3.1 RUL prognostics using the arbitrary Polynomial Chaos Expansion (aPCE)

We consider the following methodology for aPCE. A theoretical overview of the arbitrary PCE method is outlined in [25, 26], which we adapt for our prognostic model.

One-dimensional aPCE

Let $f(t, \xi)$ be a stochastic model with one model input ξ . Here, ξ follows a distribution $\mathbb{D}(\xi)$. Let Y be the output of model $f(t, \xi)$. The model can then be expanded as follows:

$$Y = f(t, \xi) \approx \sum_{i=1}^{\tilde{d}} f_i(t) P^i(\xi). \quad (10)$$

Here, \tilde{d} is the order of expansion, $f_i(t)$ are the time dependent coefficients of the expansion and $P^i(\xi)$ is the orthogonal (or orthonormal) polynomial basis with respect to the support Γ of ξ . The difference between aPCE and other PCE methods is that for the aPCE method the support can have any arbitrary form [25]. The coefficients are obtained using the probabilistic collocation method [26]. The mean and variance of the model output Y can be directly evaluated from the expansion coefficients, i.e.,:

$$\text{Mean}[Y] = f_1(t), \quad \text{Variance}[Y] = \sum_{i=2}^{d+1} (f_i(t))^2.$$

Multi-dimensional aPCE

Our prognostic model consists of multiple model input parameters and therefore, we need to construct the multi-dimensional aPCE from the one-dimensional aPCE.

The polynomial basis is constructed by tensorising the 1D orthonormal basis in the space Γ [15, 25, 27]. The multi-dimensional polynomial expansion of order \tilde{d} with n different random variables (surrogate model) is then defined as follows:

$$f(t, \xi) \approx H^{PCE}(t, \xi_1, \xi_2, \dots, \xi_n) = \sum_{i=1}^M f_i(t) \Phi_i(\xi_1, \xi_2, \dots, \xi_n). \quad (11)$$

This surrogate model $H^{PCE}(t, \xi_1, \xi_2, \dots, \xi_n)$ can be seen as the model response surface for the health indicator model $f(t, \xi)$ as defined in Equation 9. Here, M is the number of terms of the expansion, which depends on the number

of input parameters n and the order of expansion \tilde{d} and is defined as follows:

$$M = \frac{(n + \tilde{d})!}{n! \tilde{d}!}. \quad (12)$$

In Equation 11, $\Phi_i, i \in \{1, \dots, M\}$ denotes the function of the multivariate orthogonal polynomial basis P for the input parameters $\xi = \xi_1, \xi_2, \dots, \xi_n$. This basis is constructed by multiplying the univariate polynomials from the one-dimensional case:

$$\Phi_i(\xi_1, \xi_2, \dots, \xi_n) = \prod_{j=1}^n P_j^{(\alpha_j^i)}(\xi_1, \xi_2, \dots, \xi_n) \quad (13)$$

$$\sum_{j=1}^n \alpha_j^i \leq M.$$

Here, $\{\alpha_j^i, i \in \{1, \dots, M\}, j \in \{1, \dots, n\}\}$ is an $M \times n$ matrix of the polynomial degrees of parameter ξ_j in the i th expansion term [25]. Finally, the mean and variance of the multi-dimensional aPCE expansion are as follows:

$$\text{Mean}[H^{PCE}(t, \xi)] = f_1(t),$$

$$\text{Variance}[H^{PCE}(t, \xi)] = \sum_{i=2}^M (f_i(t))^2.$$

Moment-based analysis for aPCE

The first step for the aPCE is to construct the orthogonal basis of the input parameters ξ . For this, we define the polynomial $P^{(\tilde{k})}(\xi_j), j \in \{1, \dots, n\}$ of a degree \tilde{k} :

$$P^{(\tilde{k})}(\xi_j) = \sum_{i=0}^{\tilde{k}} p_i^{(\tilde{k})} \xi_j^i, \quad \tilde{k} \in \{0, \dots, \tilde{d}\}, \quad (14)$$

where $p_i^{(\tilde{k})}$ are the orthogonal polynomial coefficients. These polynomials form an orthogonal basis for the arbitrary distribution $\mathbb{D}(\xi_j)$ [25]. For this, we determine the moments of the input parameters ξ_j up until order $(2\tilde{d} - 1)$, which have to be finite. The definition for the \tilde{k}^{th} statistical moment of parameter ξ_j is:

$$\mu_{\tilde{k}} = \int_{\xi_j} \xi_j^{\tilde{k}} d\Gamma(\xi_j). \quad (15)$$

We then use these moments to construct a set of linear equations relating the moments of an input ξ_j and its corresponding orthogonal polynomial coefficients $p_i^{\tilde{k}}$ for order of expansion \tilde{k} . This set of linear equations is as follows [25]:

$$\begin{bmatrix} \mu_0 & \mu_1 & \cdots & \mu_{\tilde{k}} \\ \mu_1 & \mu_2 & \cdots & \mu_{\tilde{k}+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{\tilde{k}-1} & \mu_{\tilde{k}} & \cdots & \mu_{2\tilde{k}-1} \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} p_0^{(\tilde{k})} \\ p_1^{(\tilde{k})} \\ \vdots \\ p_{\tilde{k}-1}^{(\tilde{k})} \\ p_{\tilde{k}}^{(\tilde{k})} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad (16)$$

where $p_j^{(\tilde{k})}$, $j \in \{0, 1, \dots, \tilde{k}\}$, is the j^{th} coefficient of a polynomial of degree \tilde{k} . This set of equations is constructed by applying the orthogonality condition of polynomial $P^{(\tilde{k})}$ of degree \tilde{k} and polynomial $P^{(\tilde{l})}$ of degree \tilde{l} , i.e.,:

$$\int_{\xi_j} P^{(\tilde{k})}(\xi_j) P^{(\tilde{l})}(\xi_j) d\Gamma(\xi_j) = 0 \quad \forall \tilde{k} \neq \tilde{l}, \quad \forall \tilde{k}, \tilde{l} = 0, \dots, \tilde{d} \quad (17)$$

We also ensure that the \tilde{k}^{th} coefficient of all polynomials (, i.e., the leading coefficient) is equal to 1, i.e. [25]:

$$p_{\tilde{k}}^{(\tilde{k})} = 1 \quad \forall \tilde{k}. \quad (18)$$

The system of equations in Equation 16 can be solved if and only if the left hand side of Equation 16 is not singular [25]. The solution of the system of equations in Equation 16 consists of the coefficients of the orthogonal polynomial $P^{(\tilde{k})}(\xi_j)$ of degree \tilde{k} for one input parameter ξ_j , $j \in \{1, \dots, n\}$. Solving for all n input parameters and for all polynomials in Equation 13, we obtain the functions $\Phi_i(\xi_1, \xi_2, \dots, \xi_n)$. Lastly, these polynomials are normalised, obtaining an orthonormal polynomial as follows:

$$\hat{P}^{(\tilde{k})}(\xi_j) = \frac{P^{(\tilde{k})}(\xi_j)}{\|P^{(\tilde{k})}(\xi_j)\|}, \quad (19)$$

where $\|P^{(\tilde{k})}(\xi_j)\|$ is the normalising constant of polynomial $P^{(\tilde{k})}(\xi_j)$, i.e. [25]:

$$\|P^{(\tilde{k})}(\xi_j)\|^2 = \int_{\xi_j} [P^{(\tilde{k})}(\xi_j)]^2 d\Gamma(\xi_j). \quad (20)$$

Determining the polynomial coefficients

We now determine the coefficients of the entire polynomial chaos expansion $f_i(t)$ (see Equation 11). For this, the probabilistic collocation method (PCM) is used [26, 28].

First, we determine the roots of the constructed orthonormal polynomials of order of expansion d , $P^{(\tilde{d})}(\xi_j)$ for $j \in \{1, \dots, n\}$, which form the available collocation points where we evaluate the health indicator model $f(t, \boldsymbol{\xi})$ (see Equation 9). A collocation point is defined as a unique combination of n points, with each of these points being one of the \tilde{d} roots of the orthonormal polynomials $P^{(\tilde{d})}$. We next determine an optimal set of M collocation points using PCM [26, 28] because the number of unique combinations of collocation points \tilde{d}^n is larger than the required number of M collocation points (see Equation 12).

Evaluating each of the M collocation points in each of the M expansion terms (see Equation 11) separately gives us the time-independent polynomial coefficients $g(\boldsymbol{\xi})$ of size $M \times M$. Evaluating the obtained M collocation points in the model

$f(t, \boldsymbol{\xi})$ (see Equation 9) gives us the output y_m , $|y_m| = M \times t$. Then, the time-dependent coefficients $f_i(t)$ are obtained by applying the dot product between the time independent coefficients $g(\boldsymbol{\xi})$, the inverse of these coefficients $g(\boldsymbol{\xi})$ and the output of the model y_m , i.e.,

$$f_i(t) = \langle \langle g(\boldsymbol{\xi}), g(\boldsymbol{\xi})^{-1} \rangle, y_m \rangle \quad (21)$$

Having obtained $f_i(t)$ (see Equation 21) and Φ_i (see Equation 13), we can now construct a surrogate model $H^{PCE}(t, \boldsymbol{\xi})$ using PCE for the health indicator (see Equation 11).

Bayesian updating

Using the training data set, we have obtained a surrogate model $H^{PCE}(t, \xi_1, \dots, \xi_n)$ (see Equation 11). For the engines in the test data set, we will now calibrate this surrogate model using Bayesian updating [26], i.e., we sample from the distribution $\mathbb{D}(\boldsymbol{\xi})$ of the input parameters $\boldsymbol{\xi}$ of the surrogate model and obtain a posterior distribution $\mathbb{D}(H^{PCE,P}(t, \boldsymbol{\xi}))$ of the output $H^{PCE}(t, \boldsymbol{\xi})$ of this surrogate model. In this way, we have an engine-specific distribution of the surrogate model $H^{PCE}(t, \xi_1, \dots, \xi_n)$.

Formally, let us consider the measurements $\mathbf{y} = [d_{i,1}^{k(1)}, d_{i,2}^{k(2)}, \dots, d_{i,t}^{k(t)}]$ available up to time t for an engine i and for $k \in \{1, \dots, K\}$. With these measurements, we calibrate the surrogate model $H^{PCE}(t, \xi_1, \dots, \xi_n)$ and obtain a corresponding posterior probability distribution ($\mathbb{D}(H^{PCE,P}(t, \boldsymbol{\xi}))$).

Bayes theorem, fundamental for Bayesian updating, is as follows:

$$H^{PCE,P}(t, \boldsymbol{\xi}) = h(\boldsymbol{\xi}|\mathbf{y}) = \frac{h(\mathbf{y}|\boldsymbol{\xi})h(\boldsymbol{\xi})}{h(\mathbf{y})}, \quad (22)$$

where $h(\boldsymbol{\xi})$ is the joint prior PDF of the input parameters $\boldsymbol{\xi}$ of the model $f(t, \boldsymbol{\xi})$, $h(\mathbf{y})$ is a normalisation constant and is the prior probability of \mathbf{y} , $h(\mathbf{y}|\boldsymbol{\xi})$ is the conditional probability density function of the measurements $\mathbf{y} = [d_{i,1}^{k(1)}, d_{i,2}^{k(2)}, \dots, d_{i,t}^{k(t)}]$, given model parameters $\boldsymbol{\xi}$, and $h(\boldsymbol{\xi}|\mathbf{y})$ is the conditional PDF of model parameters $\boldsymbol{\xi}$ given the measurements \mathbf{y} , i.e. the posterior PDF of the surrogate model $H^{PCE,P}(t, \boldsymbol{\xi})$, [26].

It is commonly assumed that the measurement errors are normally distributed and independent and therefore, the likelihood function $h(\mathbf{y}|\boldsymbol{\xi})$ can be formulated as [26]:

$$h(\mathbf{y} | \boldsymbol{\xi}) \propto \exp \left[-0.5(\mathbf{y} - H^{PCE}(t, \boldsymbol{\xi}))^T \mathbf{R}_v^{-1} (\mathbf{y} - H^{PCE}(t, \boldsymbol{\xi})) \right], \quad (23)$$

where \mathbf{R}_v is the diagonal variance matrix of the measurement errors. It can be seen that the likelihood function $h(\mathbf{y}|\boldsymbol{\xi})$ relates the output of the surrogate model to the measurements \mathbf{y} .

We use a Bootstrap filter to perform Bayesian updating (see Equation 22). A sample $\boldsymbol{\xi}^{(i)}$ drawn from the input distribution $\mathbb{D}(\boldsymbol{\xi})$, is assigned a weight w_i , $i > 0$, as follows:

$$w_i = \frac{h(\mathbf{y}|\boldsymbol{\xi}^{(i)})}{\max(h(\mathbf{y}|\boldsymbol{\xi}^{(i)}))}, \quad (24)$$

where $\max(h(\mathbf{y}|\boldsymbol{\xi}^{(i)}))$ is the maximum value across all considered measurement errors (see Equation 23). Samples are selected based on these weights, i.e., a higher weight has a higher probability of being sampled for the posterior distribution of $\mathbb{D}(H^{PCE,P}(t, \boldsymbol{\xi}))$. In this way, a posterior

PDF is obtained, which is the distribution of the arbitrary PCE model, denoted as $\mathbb{D}(H^{PCE,P}(t, \boldsymbol{\xi}))$. The progression of the health indicator is then defined as the mean of the posterior PDF, which we denote by $\bar{H}^{PCE,P}(t, \boldsymbol{\xi})$.

RUL estimation

Having obtained a health indicator $\bar{H}^{PCE,P}(t, \boldsymbol{\xi})$, $t > 0$ for an engine i in the test data set, we first define a failure threshold D_f . We say that an engine i fails as soon as the health indicator $\bar{H}^{PCE,P}(t, \boldsymbol{\xi})$, $t > 0$ reaches threshold D_f . The threshold D_f is obtained based on the analysis of the failure times of all engines in the training set, i.e.,

$$D_f = \arg \min_{D \in D_s} \{(T_i - t \cdot \mathbf{1}_{\min_t\{f(t, \boldsymbol{\xi})=D\}})^2, i \in \{1, \dots, N\}\}, \quad (25)$$

where T_i is the actual failure time of engine i , with engine $i \in \{1, \dots, N\}$ in the training data set, D_s a set of thresholds that we consider for investigation and $\mathbf{1}_{(\cdot)}$ an indicator function.

Now, for an engine i in the test data set with health indicator $\bar{H}^{PCE,P}(t, \boldsymbol{\xi})$, $t > 0$, the RUL at time t is estimated as follows:

$$\bar{H}^{PCE,P}(t + x, \boldsymbol{\xi}) \leq D_f | d_{i,1}^{k(1)}, d_{i,2}^{k(2)}, \dots, d_{i,t}^{k(t)}, \quad (26)$$

where $[d_{i,1}^{k(1)}, d_{i,2}^{k(2)}, \dots, d_{i,t}^{k(t)}]$ are the measurements of engine i up to time t .

3 Results: RUL prognostics for aircraft engines

In this section, we provide the results of the developed RUL prognostic model.

3.1 Performance criteria

We use four criteria to assess the performance of the RUL prognostic model.

PHM challenge scoring function [29]

The Score for N engines in a testing data set is defined as:

$$\text{Score} = \sum_{i=1}^N S_i, \quad S_i = \begin{cases} e^{-\epsilon_i/13} - 1, & \epsilon_i \leq 0 \\ e^{\epsilon_i/10} - 1, & \epsilon_i > 0 \end{cases}, i = 1, \dots, N, \quad (27)$$

where the error ϵ_i is the difference between the predicted RUL_i and the actual RUL_i^a of engine i , i.e.,:

$$\epsilon_i = RUL_i - RUL_i^a \quad (28)$$

For each engine i the errors ϵ_i are penalised exponentially. The lower the score, the better the performance of the RUL prognostic.

Root Mean Squared Error (RMSE)

The RMSE is defined as:

$$RMSE = \sqrt{\sum_{i=1}^N \frac{\epsilon_i^2}{N}}, \quad (29)$$

with N the total number of engines in a training data set and ϵ_i is the prediction error as defined in Equation 28.

Confidence Interval for the Predicted RUL

A 95% confidence interval for the predicted RUL_i of engine i is obtained as follows:

$$(RUL_i \pm 1.96 \cdot \frac{\sigma_i^p}{\sqrt{n_i^p}}). \quad (30)$$

Here, σ_i^p is the standard deviation of the predicted RUL_i and n_i^p the sample size of the posterior health indicator distribution after Bayesian updating (see section 2.3.1).

Continuous Ranked Probability Score

The continuous ranked probability score (CRPS) is a metric evaluating the obtained distribution of the RUL prediction. CRPS is defined as follows:

$$\text{CRPS}(F(RUL_i), RUL_i^a) = \int_{-\infty}^{\infty} (F(z) - \mathbf{1}_{z \geq RUL_i^a})^2 dz. \quad (31)$$

Here, $F(RUL_i)$ is the cumulative distribution of the RUL prediction and $\mathbf{1}$ is the Heaviside step function. A CRPS value of 0 (most desirable) is obtained if the predicted distribution of RUL consists of a correct point prediction with 0 variance.

3.2 Health indicator model selection

In section 2 we introduced aPCE with a general model $H^m(t) = f(t, \boldsymbol{\xi})$. In this section we specify the model $f(t, \boldsymbol{\xi})$.

Existing studies on RUL prognostics for NASA data sets have proposed model-based prognostics using various underlying degradation patterns. In [11] a Wiener process and an exponential function are used to model the engine health degradation and the authors of [13] use a polynomial of the second order to model the engine health degradation.

We use a third order polynomial model for our aPCE. Compared with an exponential model, a third order polynomial model allows for a higher maximum order of expansion of the aPCE. Also, compared with a second order polynomial as in [13], a third order polynomial fits best the sensor measurements $\{d_{i,t}^{k(t)}\}$, $t \in \{1, \dots, T_i\}$, $k \in \{1, \dots, K\}$, $i \in \{1, \dots, N\}$ and results in an improvement of 11% in terms of the PHM scoring function.

We will model $f(t, \boldsymbol{\xi})$ with $n = 3$ uncertain input parameters, i.e., $|\boldsymbol{\xi}| = 1 \times 3$, as follows:

$$H^m(t) = \xi_1 - \xi_2 t^3 - \xi_3 t. \quad (32)$$

Here, ξ_1 is due to the fact that the measurements in the NASA data set are available after some initial degradation of the engines (i.e., non-zero initial degradation), the negative coefficients $-\xi_2$ and $-\xi_3$ indicate a decreasing health of the engines. We have decided to not include a second-order parameter, because this does not improve the model performance in terms of the PHM scoring function (within 1% difference), while computational time is increased by 5%.

To further increase the diversity of the available measurements, we make use of Bayesian calibration [30]. Through Bayesian calibration, Gaussian noise $N(0, \sigma^2)$ is added to a sample of $\mathbb{D}(\boldsymbol{\xi})$, with σ^2 being 0.01%–0.1% of the variance of the original input parameters $\boldsymbol{\xi}$ [30]. For an input parameter ξ_j , $j \in \{1, \dots, n\}$, we set σ^2 to be 0.1% of the variance of an input distribution $\mathbb{D}(\xi_j)$.

Specifically, for each sample from a distribution $\mathbb{D}(\xi_j)$, 5 extra data points are added by adding Gaussian noise from a $N(0, \sigma^2)$ distribution:

$$\xi_j + N(0, 0.1 \text{Var}[\mathbb{D}(\xi_j)]). \quad (33)$$

As a result, in case of the FD002 data set the $\mathbb{D}(\xi_j)$ distribution will consist of 1560 samples instead of 260 samples. The reason to choose 5 points is a trade off between a reduction in speed and improvement of the performance of the model. If more noise is added, the computational time increases while the performance is not increasing significantly. This approach results in an improvement of 23% in terms of the scoring function.

3.3 RUL prognostics results

In this section, the RUL prognostic model results are provided.

3.3.1 PHM challenge data set

The RUL prediction results of the first two engines of the PHM challenge testing data set are given in Figure 5. Using the training data set, an optimal value of the failure threshold $D_f = 0.83$ is obtained (see Section 2.3.1). The aPCE model ($\overline{H}^{PCE,P}(t, \xi)$) and the corresponding standard deviation are shown in this figure, as well as a visualisation of how the RUL estimation is obtained.

Table 3 shows the performance of the aPCE method in comparison with existing studies on RUL estimation using the PHM Challenge data set. Table 3 also shows whether the studies used a model-based (MB) or an artificial intelligence (AI) approach to estimate RUL and whether uncertainty quantification (UQ) is performed, i.e., whether the distribution of the estimated RUL is obtained. The obtained results are ranked 17th compared to the top 20 scores of the actual challenge in 2008 [31]. We note that among the best published results, our approach is the only one performing uncertainty quantification.

Table 3: Comparison between the arbitrary PCE method and published results for the PHM challenge data set.

Source	PHM Score	MB or AI	UQ
Peel (2008) [32]	984	AI	No
Coble and Hines (2008) [13]	2500	MB	No
Wang et al. (2008) [9]	5600	MB	Yes
Riad et al. (2010) [33]	1540	AI	No
Hu et al. (2012) [34]	1349	AI	No
Wang et al. (2012) [35]	1139	MB	No
Le Son et al. (2013) [11]	5500	MB	No
aPCE method	1288	MB	Yes

3.3.2 FD001 & FD002 data set

Using our approach, Table 4 shows that a score of 550 and a RMSE of 16.7 is obtained for the RUL prognostics of the engines in the FD001 test data set (for which $D_f = 0.87$). The RMSE is in the middle range compared to other existing studies, while the score is in the lower range. This is due to the fact that 4 out of 100 engines account for 50% of the Score. For these 4 engines, the model is not able to detect

the trend correctly leading to a relatively large error. This will be elaborated later in Section 3.4.

The score obtained for the FD002 testing data set is 8400 with a RMSE of 25.7 (for which $D_f = 0.83$), which is considerably larger than for the PHM challenge testing data set. The reason for this is that this is a more complicated data set and the number of engines in the test data set is higher. The results in terms of the PHM scoring function and RMSE can be found in Table 4, along with result of other researches found in literature. It can be seen that the arbitrary PCE method has a score and RMSE in the mid-range, which can compete with a number of publications, but there are also better performing publications. A reason for this might be that the arbitrary PCE method is a model-based method, whereas the best performing methods are artificial intelligence methods, which are generally able to capture more underlying relations between variables (at the cost of being a black-box model). On the other hand, these methods do not allow for uncertainty quantification. Also, as we have developed a model-based approach, the computational time is low (roughly 5 minutes to train the model and determine the RUL for all engines) and computational times of AI methods tend to be very high. Unfortunately, computational times are often not included in the publications and therefore, a comparison of computational times cannot take place.

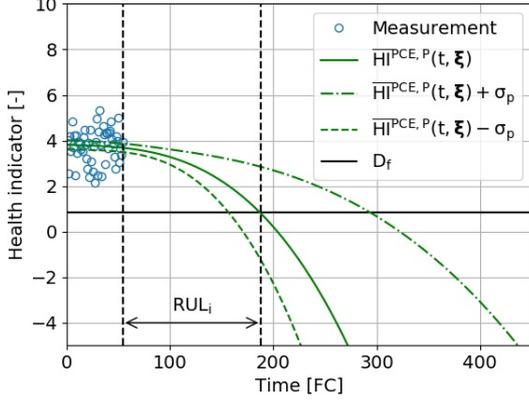
Table 4: Comparison between the arbitrary PCE method and published results for the FD002 data set.

Source	FD001 Score	FD001 RMSE	FD002 Score	FD002 RMSE	MB or AI	UQ
Ramasso (2014) [36]	216	13.3	2800	22.9	MB	No
Zhang et al. (2016) (1) [37]	480	17.9	70000	29.6	AI	No
Zhang et al. (2016) (2) [37]	474	15.7	87000	29.1	AI	No
Zhang et al. (2016) (3) [37]	334	15	5600	25.1	AI	No
Babu and Zhao (2016) [38]	1287	18.8	13570	30.3	AI	No
Bektas et al. (2017) [39]	-	18.2	-	23.6	MB	Yes
Li et al. (2018) [40]	274	12.6	12900	22.4	AI	No
Lim et al. (2018) [41]	-	14.8	-	25.5	AI	No
Ellefsen et al. (2019) [42]	231	12.6	3370	22.7	AI	No
aPCE method	550	16.7	8400	25.7	MB	Yes

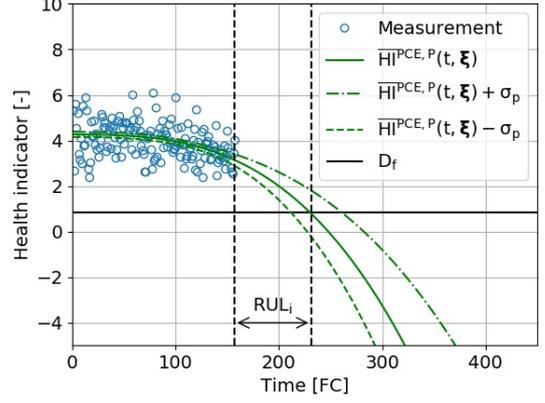
Analysis of engine specific RUL estimations - FD002

In this section we analyse in more depth the RUL estimations for the engines in the FD002 data set. The principles discussed here also hold for the FD001 data set. However, FD002 data set has more operational modes, making it a more sophisticated data set.

Table 5 shows the RUL estimates for the first 12 engines of the FD002 testing data set. Engines 2, 3 and 12 have an absolute error of 30 FC or more, while for engine 11 the RUL is predicted exactly. It can be seen, that for engines with a high RUL_i^a , the CRPS is higher. For example, engines 3, 4 and 6 have a RUL_i^a of 106, 110 and 126, respectively. Also, they have relatively high CRPS values compared to the other engines (22.57, 17.11 and 19.73, respectively). This indicates that when we have an engine with a high RUL_i^a , the model has more difficulty in obtaining an accurate distribution, which is indicated by a high CRPS. It can also be seen



(a) RUL prediction is performed 54 FC after the first time the engine is used. Estimated $RUL=188-54=134$ FC. Engine 1 of PHM challenge testing data set.



(b) RUL prediction is performed 157 FC after the first time the engine is used. Estimated $RUL=231-157=74$ FC. Engine 2 of PHM challenge testing data set.

Figure 5: RUL prediction of engines 1 and 2 of the PHM challenge testing data set.

that engine 2 has a relatively large error and a corresponding score of 98, which is significantly higher. Engines with an absolute error larger than 30 FC dominate the total score of 8400.

Table 4 shows that we obtain a score of 8400 using our aPCE approach. To understand which engines out of the total 259 engines in the FD002 testing data set have the highest contribution to this score, a breakdown of the scores per engine is performed. A histogram of the RUL prediction errors for all 259 testing engines is given in Figure 6. It can be seen that the majority of the prediction errors ($> 70\%$) has an absolute prediction error of less than 20 cycles. The areas identified in Figure 6 consist of 11 engines indicating predictions with an absolute error larger than 60, which account for 60% of the total score.

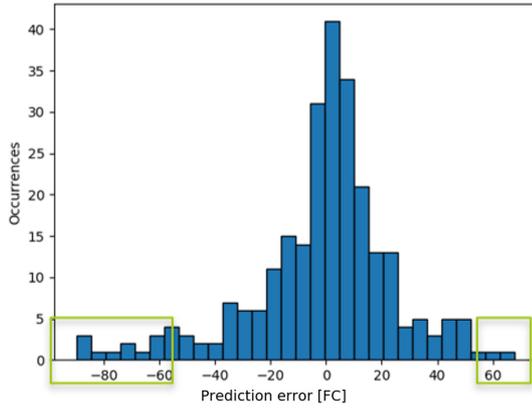


Figure 6: Histogram of the RUL prediction errors of the testing data set of the FD002 data set. The green boxes indicate the prediction errors corresponding to 60% of the PHM scoring function.

It has been identified that the engines with an absolute prediction error larger than 60 cycles is caused by engines which have only a few degradation measurements being available. More specifically, the engines with a prediction error smaller than -60 cycles (the early predictions) are also the engines with the highest actual RUL of all engines in the data set. It appears that the model has difficulty with handling engines with a relatively high actual RUL (RUL_i^a) and few measure-

ments being available. The reason for this is that during the Bayesian updating step of the model, many samples are being sampled as many samples fit the few available measurements. As more measurements are available, the sample diversity narrows down and the prediction of the model improves.

A box plot of RUL estimates for various engines is given in Figure 13. Here, the engines have been sorted in ascending order based on their predicted RUL and 22 engines out of the 260 engines have been randomly selected for visualisation. It can be seen that the boxes are small if the RUL of an engine is low and starts increasing if the RUL is higher. This illustrates that when more measurements are available (and thus the RUL is smaller), the uncertainty associated with the predicted RUL decreases. Furthermore, it can be seen that outliers are mostly on the upper bound of the prediction for engines with a high RUL. The reason for this is that the model performs best if an initial level of degradation is observed for an engine. If the model does not observe an initial degrading trend, samples leading to a too large RUL prediction are sampled more frequently because their degrading pattern also matches the few available measurements with no clear degradation initiation.

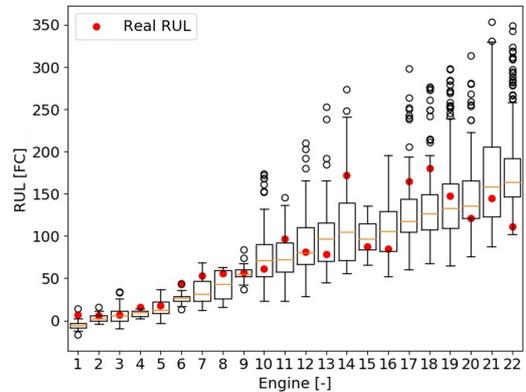


Figure 7: Box plot of RUL predictions for 22 randomly selected engines of the FD002 testing data set. The engine numbers do not represent the engine IDs.

Table 5: Results of the aPCE prognostic model applied to the first 12 aircraft engines of the FD002 testing data set.

Engine ID [-]	RUL _i [FC]	RUL _i ^a [FC]	95% C.I. low [FC]	95% C.I. high [FC]	Std. dev σ_p [FC]	Error ϵ_i [FC]	Score S_i [-]	CRPS [-]
Engine 1	24	18	16	31	11	6	0.82	4.73
Engine 2	125	79	121	129	30	46	98.48	25.25
Engine 3	76	106	67	84	19	-30	9.05	22.57
Engine 4	90	110	83	96	29	-20	3.66	17.11
Engine 5	22	15	18	25	9	7	1.01	5.02
Engine 6	130	155	126	134	32	-25	5.84	19.73
Engine 7	8	6	6	10	6	2	0.22	2.24
Engine 8	76	90	66	86	22	-14	1.94	10.31
Engine 9	8	11	6	11	6	-3	0.26	1.71
Engine 10	91	79	86	96	27	12	2.32	7.02
Engine 11	6	6	4	8	5	0	0.00	1.39
Engine 12	106	73	101	111	29	33	26.11	15.62

3.4 RUL estimation over time - FD002

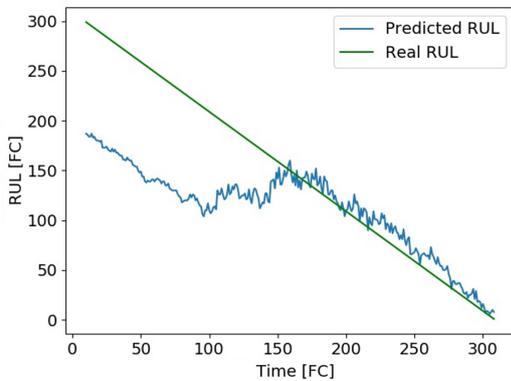
In this section we further investigate whether the RUL estimation improves as more measurements become available as time progresses (i.e., as the RUL decreases). In other words, we investigate whether the large errors in Figure 6 correspond to engines for which only a few measurements are available and have a large engine lifetime. For this, we require run-to-failure data, as we want to investigate how the error of an engine evolves over time until it has failed. The testing data set engines have been truncated after a certain time and therefore this data set is not feasible for this analysis. Therefore, we will make use of the FD002 training data set.

To train the aPCE model, 200 engines of the FD002 training data set are used. From the remaining 60 engines, six engines with the largest engine lifetime are selected to see how the prediction evolves over time. For demonstration purposes, the evaluation of the RUL will take place at 1/3 of the engine lifetime and at 2/3 of the engine lifetime for these six engines. The results can be viewed in Table 6. Table 6 shows that the model has difficulty estimating the RUL of an engine if it has a long lifetime and only a few degradation measurements are available. The predicted RUL at 1/3 of the engine lifetime is high, but when more measurements are becoming available, the prediction error converges to the actual RUL. At 2/3 of the engine lifetime, the prediction of the RUL is significantly better and will be of more value for predictive maintenance planning.

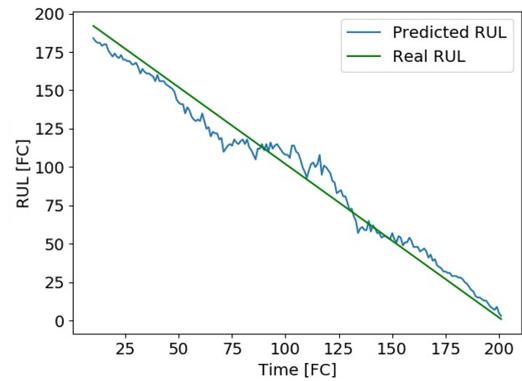
Table 6: Predictions of the RUL in flight cycles at two points in time for the engines with the longest engine lifetime of the selected 60 engines in the FD002 training data set of data set.

Engine ID [-]	At 1/3 of engine lifetime			At 2/3 of engine lifetime			Engine lifetime [FC]
	RUL _i [FC]	RUL _i ^a [FC]	Error ϵ_i [FC]	RUL _i [FC]	RUL _i ^a [FC]	Error ϵ_i [FC]	
203	94	178	-84	78	90	-12	270
228	95	174	-79	93	87	6	261
240	121	176	-55	72	89	-17	267
251	136	176	-40	94	88	6	264
255	87	226	-139	120	113	7	339
257	120	206	-86	98	103	-5	309

To see how the prediction converges over time, the RUL is determined for each point in time until failure. This is done for engine #257 with a long engine lifetime (309 FC) and for engine #249, which has a relatively shorter engine lifetime (202 FC). The progression of the predictions can be seen in Figure 8. From this figure, it becomes more evident that engines with a long engine lifetime require more time before a meaningful prediction can be made. However, it also becomes evident that eventually the prediction becomes accurate enough for maintenance planning, as the prediction converges to the actual RUL. Therefore, for maintenance planning it should be taken into account that optimal maintenance dates might be subject to change when RUL

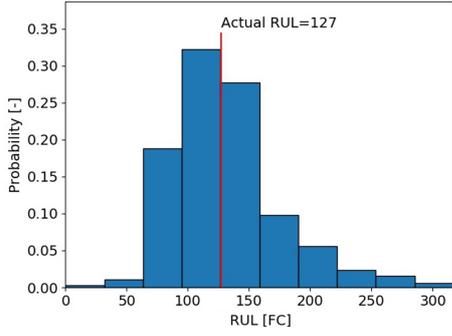


(a) RUL prediction over time of engine #257 of the FD002 training data set.

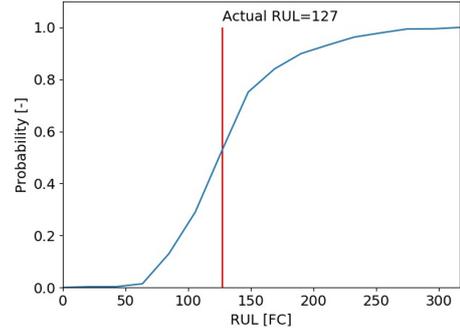


(b) RUL prediction over time of engine #249 of the FD002 training data set.

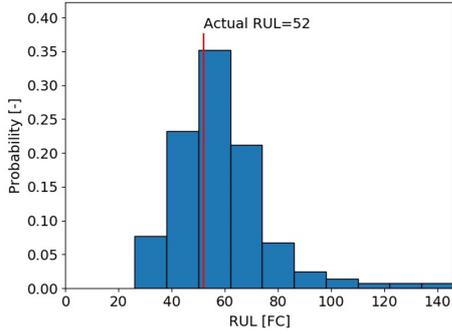
Figure 8: RUL prediction over time of engine #257 with a long engine life time and engine #249 with a shorter engine lifetime. The FD002 training data set has been used to construct the predictions. 200 engines are used for training, and these engines are used for testing.



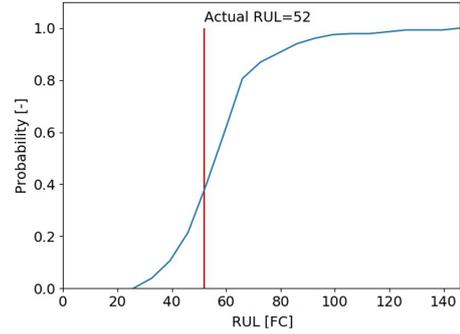
(a) Estimated PDF of the RUL of engine 249 of the FD002 training data set, obtained after 75 cycles.



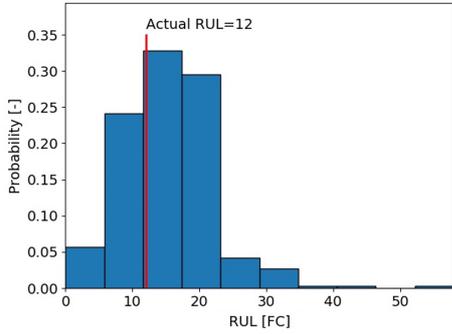
(b) Estimated CDF of the RUL of engine 249 of the FD002 training data set, obtained after 75 cycles.



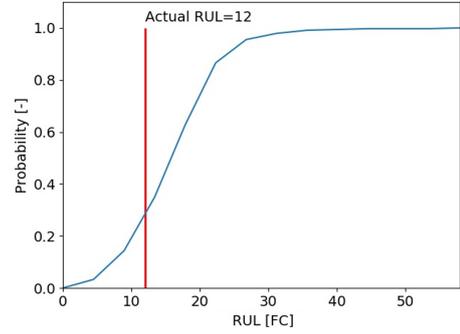
(c) Estimated PDF of the RUL of engine 249 of the FD002 training data set, obtained after 150 cycles.



(d) Estimated CDF of the RUL of engine 249 of the FD002 training data set, obtained after 150 cycles.



(e) Estimated PDF of the RUL of engine 249 of the FD002 training data set, obtained after 190 cycles.



(f) Estimated CDF of the RUL of engine 249 of the FD002 training data set, obtained after 190 cycles.

Figure 9: PDFs and CDFs of the RUL estimation for engine 249 of the FD002 training data set at various instances of time.

predictions are relatively high. On the other hand, if RUL predictions are relatively low, there is a higher certainty that the planned maintenance date resulting from the optimisation model is close to the real optimal maintenance date. This holds for both engines with a long lifetime as well as engines with a shorter lifetime. This will be tested for the engines which will be used for maintenance planning in section 4.2.

Finally, regarding uncertainty quantification, the PDF and CDF of the RUL estimation at various time instances for engine# 249 of the FD002 training data set can be viewed in Figure 9. The time instances at which the PDF and CDF are provided are after 75 cycles, 150 cycles and 190 cycles. It can be seen that the estimated PDF and CDF capture the actual RUL accurately and that the variation of the distribution

decreases for increasing time, leading to an improvement of the prediction precision.

4 Maintenance optimisation model for aircraft engines

In this section the maintenance optimisation model is introduced. The objective of the maintenance optimisation model is to use the previously developed RUL prognostic model to find optimal times to perform maintenance for aircraft engines. We formulate the maintenance of an individual engine as a Markov Decision Process (MDP) that integrates the RUL prognostic model introduced in Section 2.3. In [43] a Markov Decision Process (MDP) is proposed for maintenance opti-

misation of railways, taking into account RUL prognostics. For our research, we assume that the results of the prognostic model are true, i.e., we assume that the estimated PDF of the RUL is trustworthy, so we can formulate a fully observable MDP. When considering multiple engines, together with constraints on the availability of the hangar for maintenance (system level), we propose, a linear programming model (system level optimisation).

4.1 Single component maintenance optimisation

In this section we are interested in finding an optimal action a^* at an optimal time t^* with associated costs C^* [43]:

$$(a^*, t^*) = \arg \min_{a \in A, t \in T} C_m(a, t) + C_r(a, t), \quad (34)$$

$$C^* = \min_{a \in A, t \in T} C_m(a, t) + C_r(a, t), \quad (35)$$

where A is the set of possible maintenance actions and the set T consists of the possible times when maintenance can be planned. Furthermore, $C_m(a, t)$ accounts for the maintenance costs if maintenance is performed at time t and $C_r(a, t)$ are costs associated to the risk associated with wasting the useful life of an engine and the risk of having a component failure (unscheduled replacement). We formulate a Markov Decision Process (MDP) to solve this problem and find an optimal time for maintenance for a component. The Markov Decision Process is formulated as the tuple : $\mathcal{M} = \langle S, A, P, R, \gamma \rangle$, where S is the state space, A is the action space, P contains the state transition probabilities for going from state $s \in S$ to $s' \in S$ when taking some action $a \in A$, R is the reward received when taking an action in a certain state and γ is a discount factor which is used to discount future rewards of the MDP. We consider a discrete-time system with time steps of Δt FC to find the optimal action a given the component is in state s at time t , i.e. to find the optimal time t^* to perform maintenance for this component.

State Space

We consider the state space $S = [1 \ 2 \ \dots \ N_s]^T$ for an engine i , where state 1 indicates that this engine is new, the final state N_s indicates that the engine has failed. The intermediate states $2, \dots, N_s - 1$ indicate an increasing degradation of the health of the engine. We consider $N_s - 1$ levels of degradation where degradation level $j \in \{1, 2, \dots, N_s - 1\}$ is defined based on the health indicator $\overline{H}_i^{PCE,P}(t)$ at time t for engine i and failure threshold D_f (see section 2.2) as follows:

Engine i in state $j \in \{2, \dots, N_s - 1\}$ at time t if :

$$\begin{aligned} D_f + \frac{\overline{H}_i^{PCE,P}(t_0) - D_f}{N_s} \cdot (N_s - j) &< \overline{H}_i^{PCE,P}(t) \\ &\leq D_f + \frac{\overline{H}_i^{PCE,P}(t_0) - D_f}{N_s} \cdot (N_s - j - 1). \end{aligned} \quad (36)$$

For state 1 and N_s respectively, we have:

Engine i in state $j = 1$ at time t if :

$$D_f + \frac{\overline{H}_i^{PCE,P}(t_0) - D_f}{N_s} \cdot (N_s) < \overline{H}_i^{PCE,P}(t). \quad (37)$$

Engine i in state $j = N_s$ at time t if :

$$\overline{H}_i^{PCE,P}(t) \leq D_f + \frac{\overline{H}_i^{PCE,P}(t_0) - D_f}{N_s} \cdot (N_s - j). \quad (38)$$

An overview of the state space definition is given in Figure 10, where the total number of states N_s is equal to 5. The time that a new state is entered based on the engine HI is provided on the x-axis.

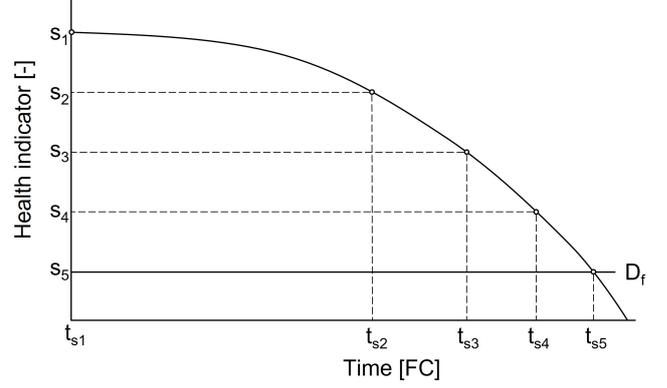


Figure 10: Overview of state definition with respect to time and the health indicator. Here, $N_s = 5$. State 1 is entered at $t=0$.

Action Space

We define the following set of actions in the action space:

$$A = \{\text{Do Nothing (DN), Do Maintenance (DM)}\}. \quad (39)$$

Transition Probabilities

The state transition probability $p(s'_{t+\Delta t} = j | s_t)$ is derived from the probability densities of the RUL prognostics obtained in section 3. Here, $\mathbb{D}(H_i^{PCE,P}(t + \Delta t, \xi))$ is the distribution of the health indicator of engine i at time $t + \Delta t$ resulting from the aPCE prognostic model (see section 2.3). We obtain this estimate at time t :

$$\text{PDF of HI at time } t + \Delta t : \mathbb{D}(H_i^{PCE,P}(t + \Delta t, \xi)). \quad (40)$$

The probability of going from one state s_t at time t to another state s' at time Δt , given that action "Do Nothing" is taken is as follows:

$$\begin{aligned} p(s'_{t+\Delta t} = j | s_t, a = \text{DN}) &= P(H_i^{PCE,P}(t + \Delta t) \\ &< D_f + \frac{\overline{H}_i^{PCE,P}(t_0) - D_f}{N_s} \cdot (N_s - j)) - P(H_i^{PCE,P}(t + \Delta t) \\ &< D_f + \frac{\overline{H}_i^{PCE,P}(t_0) - D_f}{N_s} \cdot (N_s - j - 1)). \end{aligned} \quad (41)$$

Here, s_t is the currently considered state at time t and the next state j can be any state ranging from the current state until the final state N_s . This is the probability that the health indicator of an engine i corresponds to a new state j if one of the conditions provided in Equation 36, Equation 37 or Equation 38 is met for a new state $s'_{t+\Delta t} = j$ at time $t + \Delta t$.

The probability that the engine fails at time $t + \Delta t$, given that the engine is in state s_t at time t and the action "Do Nothing" is taken, is defined as:

$$p(s'_{t+\Delta t} = N_s | s_t, a = \text{DN}) = P(H_i^{PCE,P}(t + \Delta t) < D_f). \quad (42)$$

When an engine i is in state s_t at time t based on the estimated health indicator $\overline{H}_i^{PCE,P}(t)$ at time t , and action "Do Maintenance" is taken, the transition probability of going from state s_t to state $s'_{t^*} = 1$ is as follows:

$$p(s'_{t^*} = 1 \mid s_t, a = \text{DM}) = 1 - p(s'_{t^*} = N_s \mid s_t, a = \text{DM}) \quad (43)$$

This probability is defined as 1 minus the probability of failure before the planned maintenance time.

Lastly, the probability that an engine i fails at time t^* , given that the engine is in state s_t at current time t and action "Do Maintenance" is taken, is defined as:

$$p(s'_{t^*} = N_s \mid s_t, a = \text{DM}) = P(H_i^{PCE,P}(t^*) < D_f), \quad (44)$$

where $P(H_i^{PCE,P}(t^*))$ is the health PDF of the engine health indicator at the time of maintenance t^* .

Finally, as an example, when determining the RUL prognostic for engine #249 from the FD002 training data set, the aPCE model estimates the engine health and corresponding state distributions as shown in Figure 11. The difference between the engine health PDF and the state PDF is due to the fact that we assume that we can either stay in a state or move

forward to the next state, but we can not move backwards, as stated earlier in this section. More specifically, if the distribution $\mathbb{D}(H_i^{PCE,P}(t + \Delta t, \xi))$ has a non-zero probability of moving to a previous state, we assume that this probability is equal to staying in the current state at time t .

Reward Function

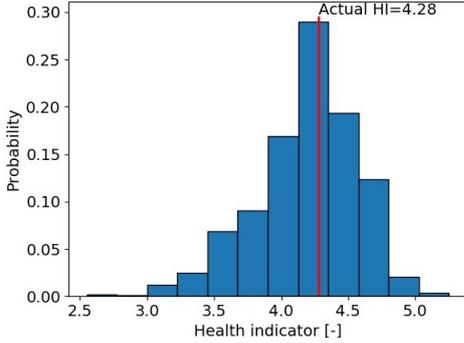
A reward is received depending on the action taken. First, if the action taken at time t is "Do Nothing", a reward of $-C_{r,dn}$ is obtained. These costs capture the risk of having the engine fail at time $t + \Delta t$. If a failure occurs at time $t + \Delta t$, a large negative penalty of $-\alpha_{fail}$ is incurred.

Finally, if the action taken at some time t is "Do Maintenance", then the reward incurred is:

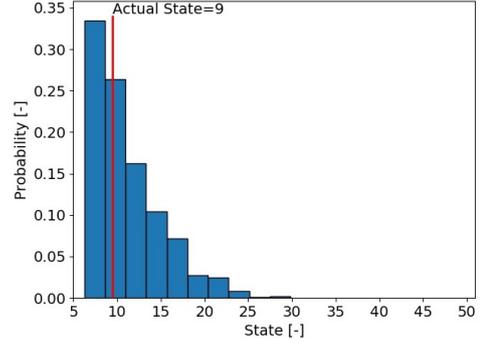
$$R(a = \text{DM}, s_t, s'_{t+\Delta t}) = -C = -(C_m + C_{r,p}), \quad (45)$$

where C_m is the cost of performing preventive maintenance, and $C_{r,p}$ is the cost of wasting the useful life of the engine.

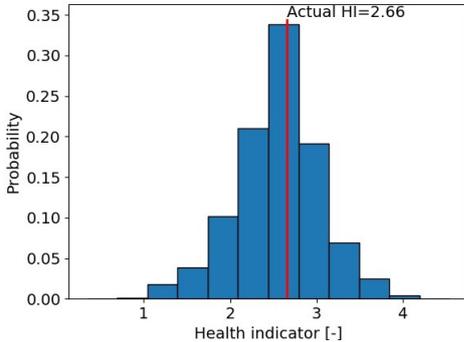
We summarise the reward function as follows:



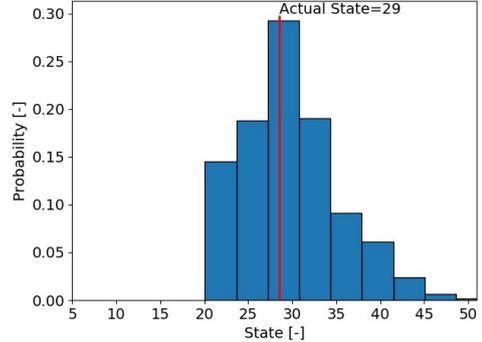
(a) Health indicator PDF ($\mathbb{D}(H_i^{PCE,P}(t + \Delta t, \xi))$) obtained after $t=100$ FC and estimated for $t=120$ FC. The engine health indicator at $t = 100$ is: $\overline{H}_i^{PCE,P}(t = 100) = 4.60$, the corresponding state is 6. The actual HI at $t=120$ FC is 4.28, indicated in the figure.



(b) State PDF obtained after $t=100$ FC and estimated for $t=120$ FC. The engine health indicator at $t = 100$ is: $\overline{H}_i^{PCE,P}(t = 100) = 4.60$, the corresponding state is 6. The actual state at $t=120$ FC is 9, indicated in the figure.



(c) Health indicator PDF ($\mathbb{D}(H_i^{PCE,P}(t + \Delta t, \xi))$) obtained after $t=150$ FC and estimated for $t=170$ FC. The engine health indicator at $t = 150$ is: $\overline{H}_i^{PCE,P}(t = 150) = 3.38$, the corresponding state is 20. The actual HI at $t=170$ FC is 2.66, indicated in the figure.



(d) State PDF obtained after $t=150$ FC and estimated for $t=170$ FC. The engine health indicator at $t = 150$ is: $\overline{H}_i^{PCE,P}(t = 150) = 3.38$, the corresponding state is 20. The actual state at $t=170$ FC is 28, indicated in the figure.

Figure 11: Example of the health indicator PDF ($\mathbb{D}(H_i^{PCE,P}(t + \Delta t, \xi))$) and corresponding state PDF obtained at two moments in time after $\Delta t = 20$ FC for engine #249 of the FD002 training data set. Here, $N_s = 50$.

$$R(a, s_t, s'_{t+\Delta t}) = \begin{cases} -C & \text{If } a = \text{DM and } s'_{t+\Delta t} \neq N_s \\ -\alpha_{fail} & \text{If } s'_{t+\Delta t} = N_s \\ -C_{r,dn} & \text{If } a = \text{DN and } s'_{t+\Delta t} \neq N_s \end{cases}. \quad (46)$$

The costs of performing planned maintenance C_m are constant and account for costs for personnel, preparing maintenance and executing maintenance. The costs related to the risk $C_{r,p}$ are engine and time specific. The formulation of the costs associated with the risk of planning maintenance are as follows:

$$C_{r,p} = P\left(H_i^{PCE,P}(t^*) > D_f | s_t\right) C_{wp}((RUL_{i,t} + t) - t^*)^2, \quad (47)$$

where the first term is the conditional probability that the health indicator of engine i is larger than the defined failure threshold D_f at the potential maintenance date t^* given that the current health of the engine corresponds with state s_t , which means that the useful life of the engine might be wasted. Also, C_{wp} is the penalty cost for wasting one life cycle of the engine. These costs are multiplied by the difference between the predicted time of failure $RUL_{i,t} + t$ (as defined in section 2.3) and the potential time of maintenance t^* . This cost penalises the fact that the maintenance action cannot be performed at the time when the RUL prognostic indicates that the engine is expected to fail. The larger the difference between $RUL_{i,t} + t$ and t^* , the higher the penalty (squared term).

We consider a negative reward $-C_{r,dn}$ when taking action "Do Nothing" at some time t . This corresponds with the risk that the engine fails at time $t + \Delta t$. These costs are formulated as:

$$C_{r,dn} = P\left(H_i^{PCE,P}(t + \Delta t) < D_f | s_t\right) \alpha_{fail}, \quad (48)$$

where the first term is the conditional probability that a failure occurs at the next time instant $t + \Delta t$ given the current state s_t . The penalty α_{fail} is incurred when an engine fails (see also Equation 46).

Solution method

The MDP is solved using the Bellman equation and the value iteration algorithm which is solved using dynamic programming. The Bellman equation is [44]:

$$V(s_t) = \max_{a \in A} \left\{ R(s_t, a) + \gamma \sum_{s'_t} P(s'_t | s_t, a) V(s'_t) \right\}. \quad (49)$$

The solution to this equation is an optimal maintenance policy for an engine, i.e., it specifies in which state corresponding to time t should maintenance be performed and in which state corresponding to time t should maintenance not be performed for each state $s \in S$. The value iteration algorithm is a widely used algorithm to solve MDPs but it is also relatively inefficient algorithm for large problems. However, the current problem size is not limiting an adequate computational effort (1 to 2 seconds to determine an optimal maintenance time for an engine) and therefore, we use this algorithm.

4.2 Single Engine maintenance optimisation - Results

We use the FD002 training data set because run-to-failure data is required to determine the optimal time of maintenance at various points in time. We use 77% (200 engines) of the data set for training the aPCE model and we use the remaining 23% (60 engines) to test the maintenance optimisation model. We first briefly present the results of the prognostic model on the selected 60 engines for testing after which we give the results of the maintenance optimisation model.

Results- RUL prognostics using aPCE

In Figure 12, the RUL progression over time of the selected $N = 60$ engines is given after applying the aPCE prognostic model described in section 2.3. Again, it can be clearly seen that the prediction performance increases over time and that engines which have a long time before failure experience a larger error (predicted RUL is smaller than the actual RUL). It can also be seen, that all engines eventually converge to the actual RUL.

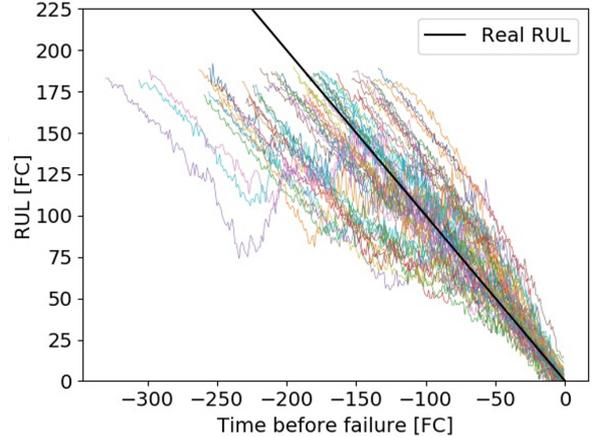


Figure 12: RUL progression over time for all 60 selected engines of the FD002 training data set

Also, the progression of the RMSE over time can be seen in Table 7. We also include the mean average error (MAE) $MAE = \sum_{i=1}^N \epsilon_i^2 / N$, with prediction error ϵ_i of engine i as defined in section 3.1. The progression of the RMSE and MAE corresponds with what we see in Figure 12 and earlier statements made about the prediction performance improving when more measurements are available (and the time before failure is smaller).

Table 7: RMSE progression over time for all 60 selected engines of the FD002 training data set

Time before failure [FC]	10	25	50	100	150	200
RMSE [FC]	5.8	10.7	17.7	25.3	33.3	60.4
MAE [FC]	4.4	7.9	13.2	19.7	26.1	56.6

Finally, a box plot for various instances for engine #249 in time can be found in Figure 13. This also shows how the uncertainty over time progresses and confirms that the prediction is accurate at all times and increases in precision over time.

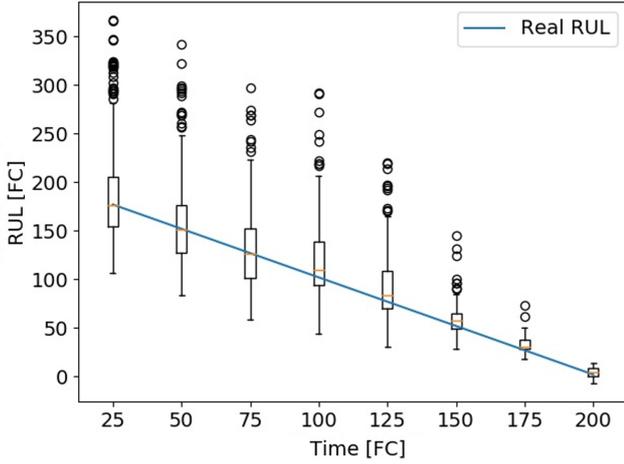


Figure 13: Box plot over time for engine #249 of the FD002 training data set.

Results of the maintenance optimisation model

The parameters of the MDP model are stated in Table 8. Furthermore, we assume that $C_m = 50$ MU, $C_{wp} = 5$ MU and $\alpha_{fail} = 500$, which are cost parameters derived from [17, 18, 43] where similar cost parameters are assumed.

Table 8: MDP parameters - maintenance optimisation component level.

Parameter	Value	Unit	Description
α_{fail}	500	-	Failure penalty
γ	0.99	-	Discount factor MDP
$ S $	50	-	Number of states
Δt	1	FC	MDP time step

The model is executed for several engines for varying times to determine the optimal time of maintenance. In Table 9 the optimal MDP time t^* at which maintenance should be performed is given for engines #211, #212, #213, #214 and #249 at different RUL prediction times. Measurements of the engine health indicator are available up until this time. The optimal state for which the action is "Do Maintenance" is indicated, as well as the aPCE prognostic health indicator $\bar{H}^{PCE,P}(t^*, \xi)$ at the optimal MDP time t^* . The aPCE RUL prognostic estimation (RUL_i^a) (see section 2.3) and the actual RUL (RUL_i^a) are given. Finally, two errors are provided. The first is the error of the optimal MDP time of maintenance t^* , defined as:

$$\epsilon_i^m = t^* - T_i, \quad (50)$$

where T_i is the actual failure time (or actual optimal time to perform maintenance). The second error ϵ_i is the difference between the predicted RUL and the actual RUL of an engine i , as was defined in section 3.1. Let us consider the first entry of the results of engine #211 evaluated after 53 FC (measurements up until 53 FC are present). Engine #211 has an actual life time (or failure time) T_i of 214 FC. It can be seen, that the optimal time to do maintenance t^* is after 181 FC, corresponding to state 46, while the estimated engine health indicator at that time is 1.15. After 53 FC, the aPCE RUL prognostic model has a RUL estimation of $RUL_i^a = 137$, while the actual RUL is equal to $RUL_i^a = 161$.

For engines #211, #212 and #214 it can be seen that when an engine is relatively young, both errors ϵ_i and ϵ_i^m are large and but when they are nearing their end of life time the errors

are significantly smaller. This behaviour is similar to the behaviour described in section 3.3.2, which showed that RUL predictions become more accurate if more measurements are available and the time until failure is small. On the other hand, engine #213 has a consistent prediction accuracy at all time instances when looking at ϵ_i^m . For the error ϵ_i of engine #249, it can be seen that it is also consistently accurate over time. It can also be seen, that for all engines at all times $\epsilon_i^m < \epsilon_i$. This is due to the fact that the MDP model tries to find an optimal time of maintenance by minimising costs, which are minimal if a failure is avoided and waste is minimised. Because the model experiences uncertainty, it is favoured to bring the optimal time of maintenance forward. This can also be derived by looking at the values of $\bar{H}^{PCE,P}(t^*, \xi)$, which are higher than the failure threshold ($D_f = 0.83$).

Regarding the states, it can be seen that the optimal state is not constant for various times and they differ per engine. The reason for this is that state boundaries as defined in section 4.1 are engine specific and time specific, meaning that they differ per engine but also for a different inspection time. As time progresses, optimal states also change because more measurements are becoming available and the uncertainty is decreased.

It has been demonstrated that the prognostic model and the component level optimisation model have an increased performance if time progresses and therefore, it is convenient to adapt a rolling horizon approach for the case study. This means that the maintenance schedule is updated at a certain interval and fixed for a certain amount of days from this point. This approach facilitates that maintenance dates are not fixed at the beginning of their operational life but rather at the end of their life when more data is available and a more accurate optimal maintenance time is found. However, it should be taken into account that the horizon is not too short, as it is more convenient to have a schedule fixed as soon as possible with as little disruptions as possible. This will be addressed in section 5.

4.3 Multi-engine maintenance optimisation

We have found the component level optimal maintenance times for individual engines. Now these optimal maintenance date times for individual engines are integrated into a system level maintenance. It is assumed that we have a set of E of operational aircraft engines operating at one point in time which each have an optimal maintenance date and should be integrated in a maintenance schedule. The driving constraint of this model is that we have limited hangar availability for engine replacements. The system level optimisation and integration is formulated as a linear programming model.

Linear Programming Formulation

First, we define the sets, decision variables and parameters of the linear programming model. Let us define I as the set containing all considered aircraft engines. Furthermore, we have set T containing all potential days at which maintenance can be planned. Next, we consider the following decision variable $w_{i,t}$:

$$w_{i,t} = \begin{cases} 1, & \text{engine } i \in I \text{ is assigned to day } t \in T \\ 0, & \text{otherwise.} \end{cases} \quad (51)$$

Table 9: Results of the MDP maintenance optimisation model for 5 engines. The model is evaluated at different times (RUL prediction time). The optimal state and time to do maintenance t^* are provided as well as the MDP error ϵ_i^m . Results of the aPCE RUL prognostic model are provided as well.

Engine i [-]	RUL prediction time [FC]	Optimal state when a=DM [-]	$\overline{H}^{PCE,P}(t^*, \xi)$ [-]	Optimal MDP time (t^*) [FC]	aPCE RUL_i [FC]	RUL_i^a [FC]	Error ϵ_i^m [FC]	Error ϵ_i [FC]
Engine $i = 211$ (actual lifetime $T_i = 214$ FC)	53	46	1.15	181	137	161	-33	-24
	73	46	1.16	187	123	141	-27	-18
	93	46	1.15	185	101	121	-29	-20
	113	46	1.16	204	100	101	-10	-1
	133	46	1.16	216	93	81	2	12
	153	46	1.15	215	71	61	1	10
	173	45	1.24	210	48	41	-4	7
	193	45	1.24	212	30	21	-2	9
Engine $i = 212$ (actual lifetime $T_i = 149$ FC)	213	48	0.99	213	5	1	1	4
	37	46	1.20	189	161	112	40	49
	57	45	1.28	186	138	92	37	46
	77	46	1.19	183	115	72	34	43
	97	45	1.27	166	78	52	17	26
Engine $i = 213$ (actual lifetime $T_i = 196$ FC)	117	44	1.38	144	37	32	-5	5
	137	43	1.48	141	14	12	-8	2
	49	46	1.18	193	153	147	-3	6
	69	46	1.19	193	133	127	-3	6
	89	46	1.18	186	106	107	-10	-1
	109	46	1.18	209	109	87	13	22
	129	45	1.27	195	76	67	-1	9
Engine $i = 214$ (actual lifetime $T_i = 146$ FC)	149	46	1.18	220	82	47	24	35
	169	45	1.27	206	48	27	10	21
	189	45	1.26	196	17	7	0	10
	36	46	1.11	173	145	110	27	35
	56	45	1.17	170	123	90	24	33
	76	45	1.17	163	97	70	17	27
Engine $i = 214$ (actual lifetime $T_i = 146$ FC)	96	45	1.17	170	83	50	24	33
	116	44	1.23	146	39	30	0	9
	136	43	1.30	135	9	10	-11	-1
	50	45	1.25	183	142	152	-19	-10
	70	45	1.24	172	111	132	-30	-21
	90	45	1.24	193	113	112	-9	1
	110	45	1.25	191	91	92	-11	-1
	130	45	1.25	193	73	72	-9	1
Engine $i = 249$ (actual lifetime $T_i = 202$ FC)	150	45	1.25	195	55	52	-7	3
	170	45	1.25	196	37	32	-6	5
	190	45	1.25	197	17	12	-5	5

Objective function

We formulate the objective function as follows, where the costs of planning maintenance are minimised:

$$\min C_{lp} = \sum_{i \in I} \sum_{t \in T} w_{i,t} \cdot c_{i,t} \quad (52)$$

The costs are minimal for the optimal maintenance date obtained through the MDP model. We define the costs $c_{i,t}$ associated with planning maintenance in the objective function as follows:

$$c_{i,t} = p_{late}^i(t - t^*)^+ + p_{early}^i(t^* - t)^+ \quad (53)$$

Here, t^* is the originally planned maintenance date found during the component optimisation and t is a potential new date. $(x)^+$ is defined as x if $x > 0$ and 0 otherwise. The first term (p_{late}^i) accounts for costs of rescheduling maintenance to a later time, while the second term (p_{early}^i) accounts for rescheduling maintenance to a time before the originally planned maintenance date:

$$p_{late}^i = P(H_i^{PCE,P}(t) < D_f) \alpha_{late} \quad (54)$$

$$p_{early}^i = P(H_i^{PCE,P}(t) \geq D_f) \alpha_{early}. \quad (55)$$

p_{late}^i is a function for rescheduling to a later date which consists of a penalty for postponing α_{late} multiplied by the probability that failure occurs before the newly considered maintenance date t , determined from the health indicator distribution $\mathbb{D}(H_i^{PCE,P}(t, \xi))$ used in section 4.1 as well. p_{early}^i (see Equation 55) is a function for planning earlier than the original date which consists of a penalty α_{early} multiplied by the probability that failure does not occur before the newly potential maintenance time. Notice that we define $\alpha_{late} > \alpha_{early}$ so that rescheduling to an earlier date is favoured over scheduling to a later date.

Constraints

Finally, we define the following constraints. The first constraint guarantees that each engine $i \in I$ should be scheduled

exactly once:

$$\sum_{t \in T} w_{i,t} = 1 \quad \forall i \in I \quad (56)$$

The second constraint ensures that the hangar availability Q is not exceeded for each $t \in T$ for the entire optimisation horizon and is defined as follows:

$$\sum_{i \in I} w_{i,t} \leq Q \quad \forall t \in T \quad (57)$$

5 Case study

The goal of this case study is to simulate a pool of aircraft engines over a finite time horizon and determine an optimal maintenance schedule by implementing the prognostic model and the maintenance optimisation model. This model will then be compared to alternative maintenance strategies in section 6 in order to quantify the benefits of the developed approach. The FD002 training data set is used for the simulation, as this data set contains run-to-failure data. Similar to section 4.2, we will use 77% (200 engines) of the data set for training and the remaining 23% (60 engines) for the actual case study simulation.

5.1 System description

We initiate a pool of E engines operating at a single point in time. The age of each engine at the beginning of the simulation is drawn from a $U(0, T_i)$ distribution to give variability to the problem, where T_i is the failure time of an engine i . It is assumed that each aircraft is operating W cycles per week. It has been demonstrated that our prognostic model improves if more measurements are available and therefore, we adapt a rolling horizon approach. A new RUL prognostic (see section 2.3) and optimal maintenance date (see section 4.1) for an engine i is obtained every t_{ins} days. If maintenance should be planned within a horizon of P_t days, the proposed maintenance strategy resulting from the MDP optimisation is accepted and maintenance is scheduled. Then, in order to see if there is a maintenance opportunity available at this date, the system level optimisation using linear programming as described in section 4.3 is applied. If maintenance should not be planned within P_t days, the proposed date is rejected and we evaluate again after t_{ins} days. When an aircraft engine is maintained, the old engine is replaced with an engine in the as-good-as-new state. This process is applied until the simulation horizon time T_{hor} is reached or no more engines can be replaced because the data is limited. The parameters for the simulation can be found in Table 10. Here, MU is short for monetary units, the currency that will be used for the case study. As outlined in section 4.2, the cost parameters have been derived from the authors of [17, 18, 43].

Furthermore, the additional parameters of the MDP model have already been given in section 4.1, only now Δt has changed from 1 FC to 1 day. Finally, the parameters which are specific for the optimisation on the system level are given in Table 11.

RUL maintenance strategy cost function

We define a new cost function for the developed RUL-maintenance strategy in order to develop a consistent costs framework for the simulation so a comparison with other strategies using a similar costs framework can be performed.

Table 10: Case study parameters

Parameter	Value	Unit	Description
E	20	-	Number of operational engines at a time
Q	1	-	Hangar availability
W	10	Flights/week	Flights per week per aircraft
Δt	1	day	MDP time step
t_{ins}	14	days	Time between RUL prognostic updates
t_m	1	day	Maintenance duration
T_{hor}	300	days	Simulation horizon
P_t	21	days	Maintenance planning horizon
C_f	250	MU	Corrective maintenance costs
C_m	50	MU	Preventive maintenance costs
C_{wp}	5	MU/cycle	Costs for wasting 1 useful lifecycle
C_{AOG}	75	MU/day	Costs for aircraft on ground

Table 11: Case study parameters - maintenance optimisation system level.

Parameter	Value	Unit	Description
α_{late}	100	-	Penalty for postponing planned day of maintenance
α_{early}	50	-	Penalty for preponing planned day of maintenance

The cost function for the RUL-maintenance strategy is as follows:

$$C_{RUL-M} = \sum_{i=1}^I (C_m + C_{wp} (T_i - t^*)) \cdot \delta(T_i > t^*) + (C_f + C_{AOG} (t_{post,i} - T_i)) \cdot \delta(T_i < t^*). \quad (58)$$

Here, C_m is the cost of performing preventive maintenance, C_f is the cost of performing corrective maintenance and C_{AOG} is the aircraft on ground (AOG) cost which is nonzero if failure occurs and maintenance cannot be performed because of hangar unavailability. C_{wp} is the cost for wasting one cycle of a component because maintenance is planned too early. T_i is the true failure time of engine i and $t_{post,i}$ is the new maintenance date of an engine i in case of an AOG situation. Finally, $\delta(x) = 1$ when x is true and 0 else.

5.2 RUL maintenance strategy results

The schedule of the first 100 days is provided in Table 12. It provides information about the day that maintenance is scheduled and the optimal day at which maintenance should have been scheduled. Furthermore, it is indicated whether a failure occurred or not and how many useful flight cycles have been wasted. Also, a cost breakdown is provided.

It can be seen that using the settings which have been stated in section 5.1, result in a schedule without failures during the first 100 days and that the wasted life cycles are significantly lower than the errors found in section 3.3.2 where we provided the results of the prognostic model. This is due to the fact that the schedule is fixed every 14 days for the next 21 days. It can be seen that engine 203 corresponds to the highest error of 21 FC. For these 15 engines, a RMSE of 11 FC is found, significantly lower than for the FD002 testing

Table 12: Optimised maintenance schedule for a pool of aircraft engines for 100 days.

Engine ID [-]	Planned date [day]	Failed Engine [-]	Initial age engine [FC]	Optimal date [day]	Wasted life [FC]	AOG costs [MU]	Corrective m. costs [MU]	Preventive m. costs [MU]	Wasted life costs [FC]	Total costs [MU]
Engine 213	15	No	174	16	2	0	0	-50	-10	-60
Engine 256	25	No	117	33	12	0	0	-50	-60	-110
Engine 248	29	No	184	36	10	0	0	-50	-50	-100
Engine 231	37	No	82	44	11	0	0	-50	-55	-105
Engine 235	38	No	117	47	13	0	0	-50	-65	-115
Engine 233	42	No	124	46	7	0	0	-50	-35	-85
Engine 229	43	No	113	45	3	0	0	-50	-15	-65
Engine 203	51	No	177	65	21	0	0	-50	-105	-155
Engine 245	73	No	131	86	19	0	0	-50	-95	-145
Engine 212	77	No	34	81	6	0	0	-50	-30	-80
Engine 237	83	No	86	87	6	0	0	-50	-30	-80
Engine 205	90	No	59	98	12	0	0	-50	-60	-110
Engine 225	91	No	24	93	3	0	0	-50	-15	-65
Engine 219	99	No	15	99	1	0	0	-50	-5	-55
Engine 240	100	No	111	109	14	0	0	-50	-70	-120

data set (RMSE=25.7 FC). Also, it can be seen that using the current case study parameters no AOG events are present.

It is noted that during this time frame of 100 days two clashes occur of engines requiring maintenance on the same day. The system level optimisation model is used for planning of these engines. Engine 233 and engine 229 both have their optimal maintenance time on day 43. However, the probability of failure of engine engine 233 is larger at day 44 ($p=0.25$) than the probability of failure of engine 229 ($p=0.11$). Also, as the penalty for postponing is larger than the penalty for bringing forward, the model decides in this case to bring engine 233 forward to day 42. If a clash of only two engines occurs, it is often the case that one of the two engines is brought forward, but if more than 2 engines clash, the system optimisation model is more likely to postpone an engine as well. Another clash occurs between engine 205 and engine 225. Both are scheduled at day 91, but although the optimal day for engine 205 is later than for engine 225, engine 205 is brought forward. This is due to the fact that the prediction for engine 205 experiences a higher uncertainty and a higher probability of failure at day 92 ($p=0.28$ for engine 205 and $p=0.21$ for engine 225).

Next, the results of the entire simulation are analysed. The schedule of the engines for 300 days can be viewed in Figure 14. Here, the predicted optimal maintenance time is shown at which maintenance is planned. Furthermore, the actual optimal maintenance time is shown as well as a reference. Ideally, the predicted optimal date and the actual optimal date are equal, but the predicted optimal time also encounters uncertainty. Therefore, with the current parameters the model is rather conservative as no failures occur and some useful cycles are wasted. This will be analysed in section 6.3 with a parameter sensitivity analysis.

Furthermore, the results of the case study for the RUL maintenance strategy can be found in Table 13. It can be seen that more than half of the costs are associated with wasting useful cycles and the other half is associated with the costs of performing preventive maintenance and no costs are related to failures or aircraft on ground costs. The root mean squared error, defined and used in section 3.3.2, is equal to 11.9 FC. This is significantly lower than the RMSE found when testing the prognostic model on the FD002 testing data set (RMSE=25.7 cycles), due to the fact that more measurements are available during the case study when decisions are made. The current presentation of results does not provide a good insight into how the developed model compares to

existing maintenance strategies. Therefore, in section 6 we will develop three other (existing) maintenance strategies so we can compare our model and see the benefits of our model.

6 Comparison with alternative maintenance strategies

In this subsection, three alternative maintenance strategies are provided which will serve as a means to compare the performance of the developed RUL-maintenance strategy.

6.1 Development of alternative maintenance strategies

Periodic MTTF maintenance

Periodic mean-time-to-failure maintenance (PM-MTTF) is a maintenance strategy where components are replaced at fixed intervals. For this research, we use the mean-time-to-failure (MTTF) derived from historic data to determine the intervals at which replacement should occur, which has also been used in [17]. Their definition is as follows [17]:

$$T_{PM} = \left\lceil \frac{\bar{T}_F}{t_{ins}} \right\rceil \cdot t_{ins} \quad (59)$$

Here, \bar{T}_F is the MTTF derived from historical data (the training data set) and t_{ins} is the RUL prognostic update interval as described in section 5.1. In this case, $\lceil x \rceil$ is defined as the upper integer of x . The costs of periodic maintenance are consistent with the costs for the RUL maintenance strategy as described in section 5.1. The costs of the periodic maintenance strategy for all engines are then as follows:

$$C_{PM} = \sum_{i=1}^I (C_m + C_{wp}(T_i - T_{PM,i})) \cdot \delta(T_i > T_{PM,i}) + (C_f + C_{AOG}(t_{post,i} - T_i)) \cdot \delta(T_i < T_{PM,i}), \quad (60)$$

where $T_{PM,i}$ is the time at which periodic maintenance is planned for an engine i .

Run-to-failure maintenance

Run-to-failure (RTF) maintenance means that only corrective maintenance is performed when the component has failed

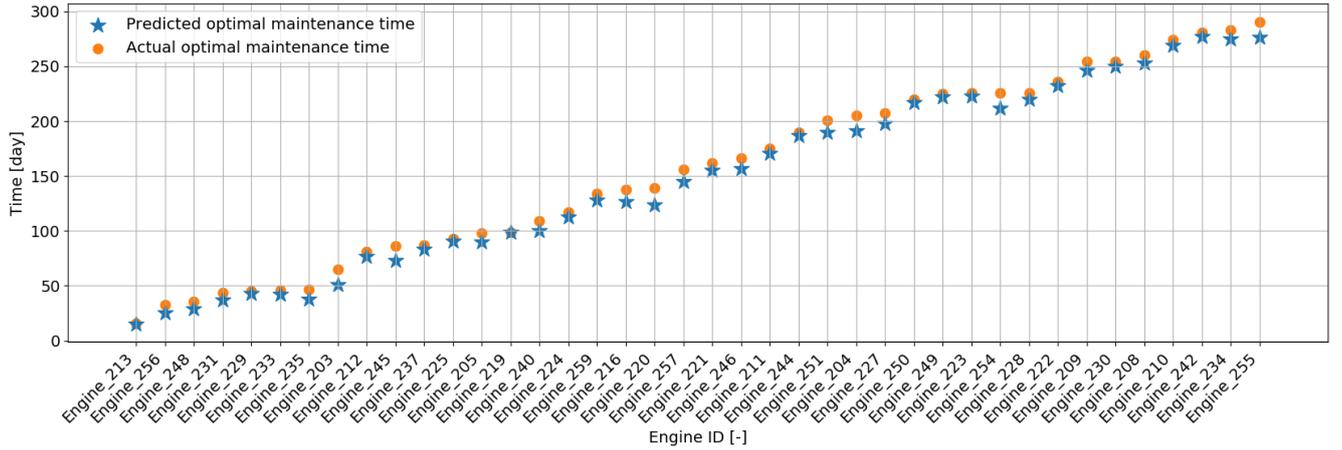


Figure 14: Aircraft engine maintenance schedule after applying the RUL maintenance strategy.

Table 13: RUL maintenance strategy case study results and a simulation time horizon of 300 days.

Scheduled replacements [-]	Unscheduled replacements [-]	Wasted life [FC]	Preventive m. costs [MU]	Corrective m. costs [MU]	Wasted life costs [MU]	AOG costs [MU]	Total costs [MU]
40	0	422	2000	0	2110	0	4110

and maintenance is performed immediately if availability allows for this. The costs associated with this strategy are defined as:

$$C_{RTF} = \sum_{i=1}^I C_f + C_{AOG} (t_{post,i} - T_i). \quad (61)$$

Here, only corrective maintenance costs C_f and aircraft on ground costs C_{AOG} are taken into account.

Perfect RUL maintenance

Finally, perfect RUL maintenance is considered where it is assumed that perfect RUL prognostic information is available and maintenance is performed just before the actual time of failure. Next to preventive maintenance costs, we also consider costs of wasting useful life in the case that several engines require maintenance simultaneously and one needs to be brought forward. The costs are defined as follows:

$$C_{PerfM} = \sum_{i=1}^I C_m + C_{wp} (T_i - t_{pre,i}), \quad (62)$$

where $t_{pre,i}$ is the date that maintenance is being performed in the case that performing maintenance just before T_i is not possible because of hangar unavailability.

6.2 Comparing RUL maintenance with alternative strategies

The results of the other three strategies and the developed RUL maintenance strategy are given in Table 14. Also, an overview of the results is provided in the form of a bar chart in Figure 15. The perfect RUL maintenance strategy serves as a baseline of the ideal world, which we are trying to accomplish. It can be seen that with the current case study parameters, the RUL maintenance strategy outperforms the MTTF-PM strategy and the RTF strategy. The MTTF-PM strategy and the RTF strategy are roughly 2.9 and 2.5 times more costly than the RUL maintenance strategy. Furthermore, the RUL maintenance strategy is roughly two times as expensive as the costs in the ideal world. This is due to the fact that there

is still uncertainty in the predictions, leading to still some wasted useful life (422 FC for 40 engines) which accounts for half of the costs. However, the amount of wasted life of the RUL maintenance strategy is significantly lower than for the MTTF-PM strategy. Waste has been reduced by a factor of 2.5 compared to the MTTF-PM strategy, while the MTTF-PM strategy also has 22 failures (failures have 0 waste). Also, 6 cycles have been wasted for the perfect RUL maintenance strategy because some engines required maintenance at the same day which resulted in engines maintenance dates being brought forward.

It can also be seen that the MTTF-PM strategy is more detrimental than the RTF maintenance strategy. This is due to the fact that approximately half of the engines requires a corrective repair and on top of that, the other half of the engines have many wasted useful life. The reason for this is that the lifetime of engines varies significantly and the MTTF can differ significantly from the actual time-to-failure. This can be viewed in Figure 16 as well, where the schedule of the MTTF-PM strategy is displayed. The engines requiring corrective repair are indicated with black dots here. It can be seen that indeed the wasted life for engines which have not failed is significantly larger than the errors that were observed in Figure 14. This is also supported by the RMSE of 66.8 FC, which is significantly higher than the RMSE of the RUL maintenance strategy (RMSE=11.9 FC). Also, these two strategies are experiencing AOG costs due to the fact that engines fail on the same day. This leads to aircraft for which maintenance needs to be postponed while they cannot fly.

On overall, it can be concluded that using these simulation parameters the RUL maintenance strategy shows benefits in terms of costs and reducing the number of failed engines compared to existing strategies. The model still encounters some uncertainty leading to wasting useful life, however, it is significantly lower than the MTTF-PM strategy. This is in correspondence with results found by the authors of [17], from which the cost functions have been derived.

Table 14: Case study simulation results for a pool of aircraft engines for 4 different maintenance strategies and a simulation time horizon of 300 days.

Strategy	Scheduled replacements [-]	Unscheduled replacements [-]	Wasted life [FC]	Preventive m. costs [MU]	Corrective m. costs [MU]	Wasted life costs [MU]	AOG costs [MU]	Total costs [MU]
RUL maintenance	40	0	422	2000	0	2110	0	4110
MTTF PM maintenance	18	22	1011	900	5500	5055	450	11905
RTF maintenance	0	40	0	0	10000	0	225	10225
Perfect RUL maintenance	40	0	6	2000	0	30	0	2030

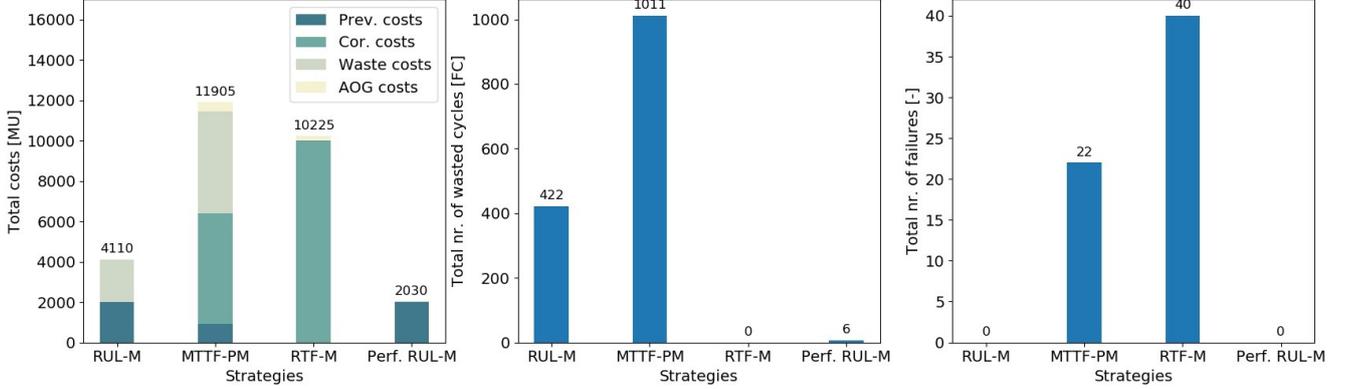


Figure 15: Bar chart of the case study results with a simulation horizon of 300 days. The left figure corresponds to the strategy costs, the middle figure corresponds to the total number of wasted cycles and the right figure corresponds to the number of unscheduled replacements (failures).

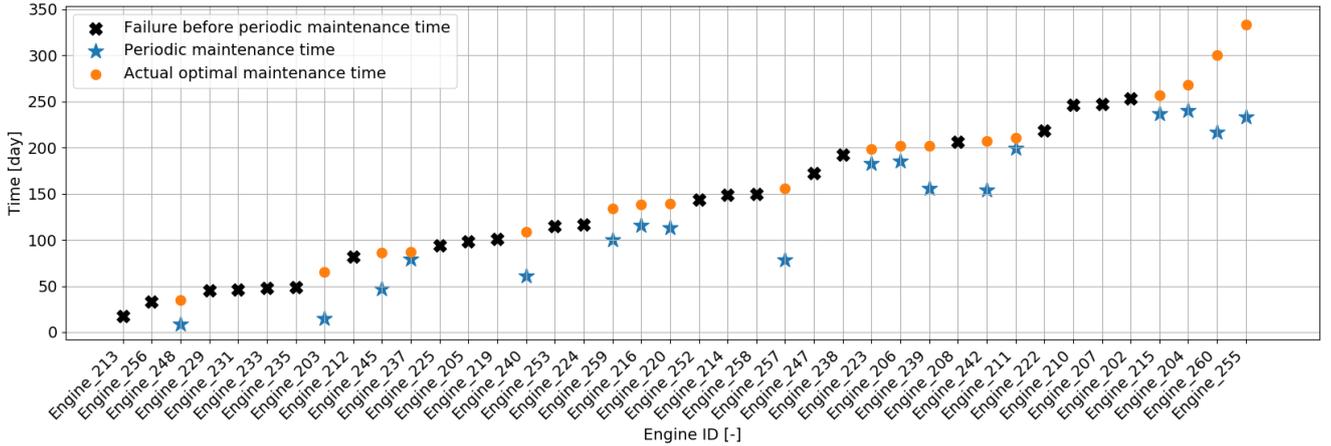


Figure 16: MTTF-PM strategy maintenance schedule with a simulation horizon of 300 days.

6.3 Sensitivity analysis

In this section we will provide a sensitivity analysis. It was stated that the cost parameters were derived from literature and therefore due to unavailability of real data and therefore, it is interesting to see the effect of varying these cost parameters on the performance indicators. The effect of varying model parameters will be performed for all maintenance strategies.

Figure 17 displays the effect on varying α_{fail} , the failure penalty of the MDP. It was presented in section 4.2 that initially this parameter was equal to 500. It can be seen in the figure that the effect of varying this penalty does not have an effect on the other strategies except for our RUL strategy because the parameter is specific for the MDP. From this analysis, it can be seen that there would be actually an optimum in terms of minimisation of costs for α_{fail} being between 400 and 1000. If α_{fail} is increased, it can be seen that the number of wasted cycles increases significantly and the

number of failures converges to 0. Therefore, the model can be easily adjusted to the needs of a potential user by changing α_{fail} , where a high α_{fail} allows for the absence of unexpected failures leading to a safer approach and a lower α_{fail} allows for a reduction in costs while the risk of unexpected failures increases. Although RUL maintenance outperforms PM clearly, RUL maintenance is still more expensive than perfect RUL maintenance. This is due to the fact that our RUL maintenance strategy still experiences uncertainty and the RUL cannot be perfectly predicted. The difference in costs can therefore be attributed to extra waste costs if no failures occur.

In Figure 18 the cost of preventive maintenance C_m is varied. Here it can be seen that the rate at which the total costs are increasing is similar for the RUL maintenance strategy and the perfect RUL maintenance strategy, but the PM strategy has a slower rate. This is due to the fact that many corrective failures occur for PM and as a result, raising preventive costs has a lower share in total costs. Also, it can

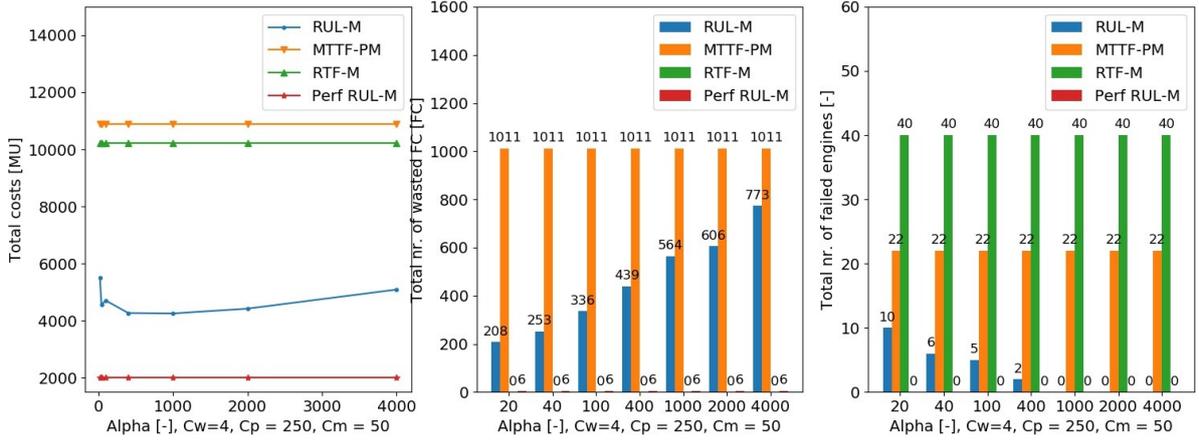


Figure 17: Sensitivity analysis for varying the MDP failure penalty α_{fail} . The left figure corresponds to the strategy costs, the middle figure corresponds to the total number of wasted cycles and the right figure corresponds to the number of unscheduled replacements (failures).

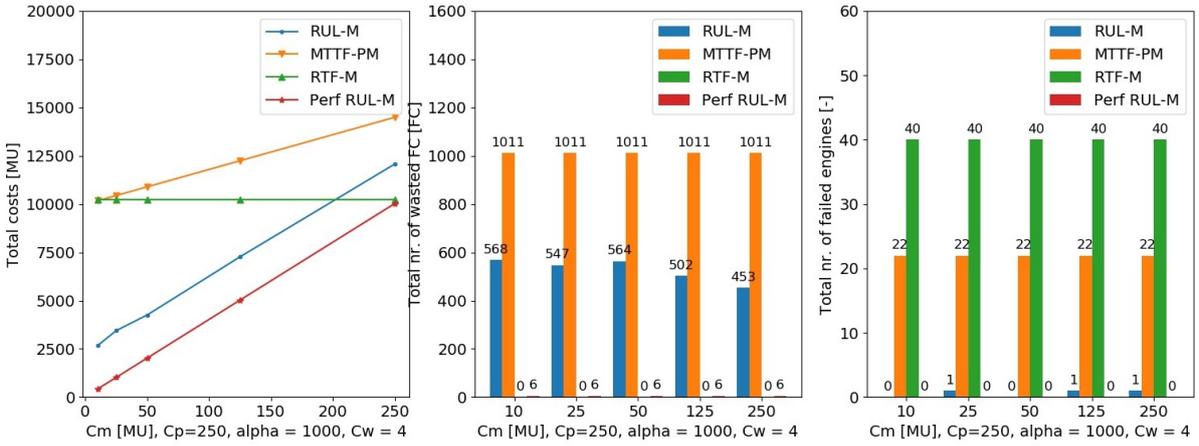


Figure 18: Sensitivity analysis for varying the cost of preventive maintenance C_m . The left figure corresponds to the strategy costs, the middle figure corresponds to the total number of wasted cycles and the right figure corresponds to the number of unscheduled replacements (failures).

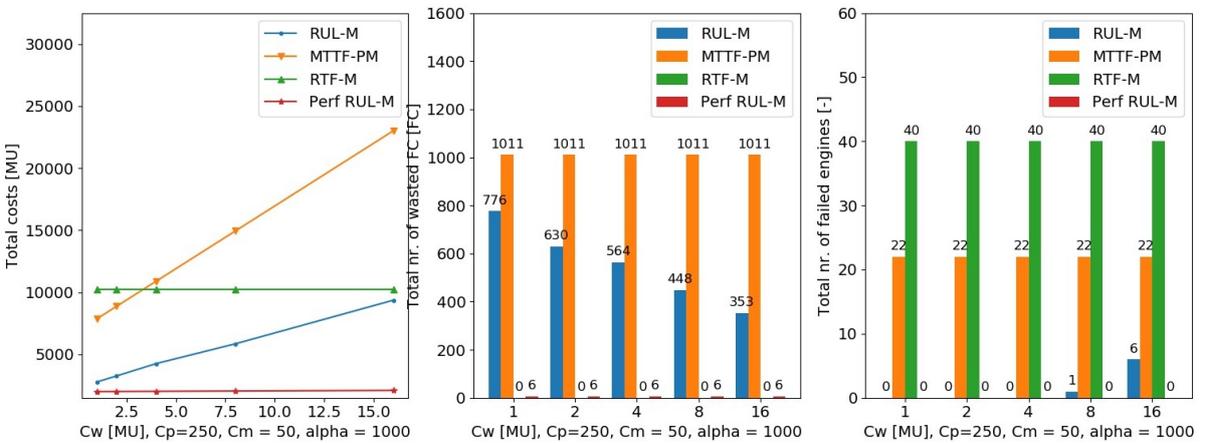


Figure 19: Sensitivity analysis for varying the cost of wasting a useful life cycle C_{wp} . The left figure corresponds to the strategy costs, the middle figure corresponds to the total number of wasted cycles and the right figure corresponds to the number of unscheduled replacements (failures).

be seen that for small values of C_m (≤ 50), the total number of wasted cycles is roughly constant, but a decreasing trend occurs for larger values of C_m (> 50). The reason for this is that when C_m increases and the failure penalty remains constant, the price of taking more risk is reduced. Therefore, maintenance is generally planned later, closer to the perfect time of maintenance at the risk of more failures. For smaller values of C_m it is not worthwhile to plan more conservatively and C_{wp} becomes the most dominant cost parameter, leading to a stagnation. Finally, it can be noted that the costs of the RUL maintenance strategy are higher than for the RTF strategy if C_m is equal to C_p . It would in this case actually be more favourable to let engines fail, but the model is still in favour of planning maintenance and having wasted cycles instead. The reason for this is that the model implicitly assumes that planning maintenance is always more favourable than letting engines fail.

Finally, it can be seen in Figure 19 what the effect is of changing the cost of wasting a useful life cycle C_{wp} . It can be seen that for RUL maintenance the costs are increasing for increasing C_{wp} and that at a certain point, it is more worthwhile to let engines fail because planning on time leads to too many wasted cycles, leading to a higher cost. PM results in many wasted life cycles and therefore the cost rate is higher. The actual cost relation between wasting a useful life cycle and the cost of performing maintenance is unknown. However, the lower this cost would be, the closer the model is to the perfect RUL maintenance strategy, while PM and RTF maintenance still are considerably larger. Also, with the parameters shown in Figure 19 it becomes less favourable to use RUL maintenance if C_{wp} becomes larger than 35% of C_m . Then, this would lead to an unlikely scenario which favours RTF maintenance.

7 Conclusion and Recommendations

In this paper, a novel prognostic model for aircraft engines using polynomial chaos expansions has been developed and integrated into an optimisation model to obtain an aircraft engine maintenance schedule. The objective of this study was to research the possibilities of using a polynomial chaos expansion approach for aircraft engine RUL prognostics and how this approach could be used for uncertainty quantification. Furthermore, it was researched how this approach could be used to develop a maintenance optimisation model resulting in optimal maintenance times for a pool of aircraft engines.

It has been demonstrated that the arbitrary polynomial chaos expansion method is suitable for RUL prognostics. For the data sets consisting of one failure mode and six operational modes, it appeared that our method performs in the mid-range compared to other approaches found in literature. The advantage of our method is that we do not make use of a black-box model and thus our approach is transparent. Also, the aPCE prognostic method allows for uncertainty quantification in a faster way than Monte Carlo sampling.

The aPCE prognostic approach is limited if few historic measurements are available or if an engine has a long lifetime. The reason for both instances is that the model performs best if a decreasing trend has been initiated for the health indicator. The absence of this initiation results in less good estimations. However, it was demonstrated that the model improves over time if more sensor measurements become available. The prediction quality in terms of uncertainty also improved over time, as the variance of the prediction also decreased.

A Markov Decision Process was developed to determine optimal maintenance dates for aircraft engines by minimising costs. The driving parameters were costs corresponding to unexpected failures and costs related to wasting useful engine life. As for the prognostic model, it also holds for the MDP model that optimal maintenance dates are found if more sensor measurements are available.

All models are integrated and completed with a linear programming model to find the optimal maintenance dates for a pool of engines with hangar availability as limiting factor. The case study shows promising results in terms of costs; costs are significantly reduced compared to RTF and MTTF-PM strategies. Furthermore, no failures occurred and the number of wasted cycles reduced by a factor of 2.5 compared to the MTTF-PM strategy. The sensitivity analysis showed that this also holds for different cost combinations and variations of the MDP failure penalty.

It can be concluded that the aPCE prognostic approach is useful for RUL prognostics and uncertainty quantification. Furthermore, it has been demonstrated that this approach can be used to determine an optimal maintenance policy of when to schedule maintenance, which is more cost effective compared to other strategies. Although this model has been developed for aircraft engines, it can also be used in other fields because the model is entirely based on the degrading behaviour of multiple sensor measurements of a system over time.

The model also has some limitations and therefore, some recommendations can be proposed. First of all, the current health indicator should be expanded for multiple failure modes. Also, the data that is used is artificially created data by the CMAPSS simulator. It would be recommended to apply the model to actual aircraft engine turbofan data in order to see the effectiveness with actual data.

Furthermore, the current cost parameters have been assumed and their relative differences have been derived from literature because this data was not available. Therefore, it would be recommended to use real aircraft engine maintenance cost parameters to investigate if the theoretical setup of this research has similar cost improvements compared to the real world.

Finally, the maintenance optimisation model should be extended if it would be integrated into a maintenance schedule of a MRO company. Currently, the pool of aircraft engines are scheduled based on hangar availability but regardless of specific maintenance opportunities, a flight schedule or safety requirements. The LP formulation can easily be expanded to take these constraints into account.

References

- [1] J. Hessburg. *Air Carrier MRO Handbook: Maintenance, Repair, and Overhaul*. An aviation week book. McGraw-Hill, 2001.
- [2] M. Gerdes, D. Scholz, and D. Galar. "Effects of condition-based maintenance on costs caused by unscheduled maintenance of aircraft". In: *Journal of Quality in Maintenance Engineering* 22 (Oct. 2016), pp. 394–417.
- [3] Y. Lei et al. "Machinery health prognostics: A systematic review from data acquisition to RUL prediction". In: *Mechanical Systems and Signal Processing* 104 (2018), pp. 799–834.

- [4] J. P. Sprong, X. Jiang, and H. Polinder. "A deployment of prognostics to optimize aircraft maintenance - A literature review". In: *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM 11.1* (2019).
- [5] K. Medjaher, N. Zerhouni, and J. Baklouti. "Data-driven prognostics based on health indicator construction: Application to PRONOSTIA's data". In: *2013 European Control Conference, ECC 2013* (July 2013), pp. 1451–1456.
- [6] G. Vachtsevanos et al. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. John Wiley & Sons, Ltd, 2006.
- [7] P. Paris and F. Erdogan. "A Critical Analysis of Crack Propagation Laws". In: *Journal of Basic Engineering* 85 (1963), pp. 528–533.
- [8] P. Wen et al. "A generalized remaining useful life prediction method for complex systems based on composite health indicator". In: *Reliability Engineering and System Safety* 205. September 2020 (2021), p. 107241.
- [9] T. Wang et al. "A Similarity-Based Prognostics Approach for Remaining Useful Life Estimation of Engineered Systems". In: *2019 IEEE International Conference on Prognostics and Health Management, ICPHM 2019* (2008).
- [10] R. Li, W. J. Verhagen, and R. Curran. "A comparative study of Data-driven Prognostic Approaches: Stochastic and Statistical Models". In: *2018 IEEE International Conference on Prognostics and Health Management, ICPHM 2018* (June 2018), pp. 1–8.
- [11] K. Le Son et al. "Remaining useful life estimation based on stochastic deterioration models: A comparative study". In: *Reliability Engineering and System Safety* 112 (2013), pp. 165–175.
- [12] Y. W. Hu et al. "Sequential Monte Carlo Method Toward Online RUL Assessment with Applications". In: *Chinese Journal of Mechanical Engineering* (2018).
- [13] J. B. Coble and J. W. Hines. "Prognostic algorithm categorization with PHM challenge application". In: *2008 International Conference on Prognostics and Health Management, PHM 2008* (2008).
- [14] M. Orchard, B. Wu, and G. Vachtsevanos. "A particle filtering framework for failure prognosis". In: *Proceedings of the World Tribology Congress III - 2005* (Jan. 2005), pp. 883–884.
- [15] P. L. Duong and N. Raghavan. "Uncertainty quantification in prognostics: A data driven polynomial chaos approach". In: *2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017* (2017), pp. 135–142.
- [16] J. B. Nagel, J. Rieckermann, and B. Sudret. "Principal component analysis and sparse polynomial chaos expansions for global sensitivity analysis and model calibration : Application to urban drainage simulation". In: *Reliability Engineering and System Safety* 195. January 2019 (2020), p. 106737.
- [17] K. T. Nguyen and K. Medjaher. "A new dynamic predictive maintenance framework using deep learning for failure prognostics". In: *Reliability Engineering and System Safety* 188. March (2019), pp. 251–262.
- [18] L. Zhang and J. Zhang. "A Data-Driven Maintenance Framework under Imperfect Inspections for Deteriorating Systems Using Multitask Learning-Based Status Prognostics". In: *IEEE Access* 9. Ddm (2021), pp. 3616–3629.
- [19] P. Do Van et al. "Remaining useful life (RUL) based maintenance decision making for deteriorating systems". In: *IFAC Proceedings Volumes (IFAC PapersOnline)* 45.31 (2012), pp. 66–72.
- [20] H. C. Vu et al. "Dynamic opportunistic maintenance planning for multi-component redundant systems with various types of opportunities". In: *Reliability Engineering and System Safety* 198. January (2020), p. 106854.
- [21] Q. Feng et al. "An optimization method for condition based maintenance of aircraft fleet considering prognostics uncertainty". In: *The Scientific World Journal* 2014 (2014).
- [22] I. de Pater and M. Mitici. "Predictive maintenance for multi-component systems of repairables with Remaining-Useful-Life prognostics and a limited stock of spare components". In: *Reliability Engineering & System Safety* 214 (May 2021), p. 107761.
- [23] T. Li et al. "An adaptive-order particle filter for remaining useful life prediction of aviation piston pumps". In: *Chinese Journal of Aeronautics* 31.5 (2018), pp. 941–948.
- [24] Y. Hu et al. "Reinforcement learning-driven maintenance strategy: A novel solution for long-term aircraft maintenance decision optimization". In: *Computers and Industrial Engineering* 153. December 2020 (2021), p. 107056.
- [25] S. Oladyshkin and W. Nowak. "Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion". In: *Reliability Engineering and System Safety* 106 (2012), pp. 179–190.
- [26] S. Oladyshkin, H. Class, and W. Nowak. "Bayesian updating via bootstrap filtering combined with data-driven polynomial chaos expansions: Methodology and application to history matching for carbon dioxide storage in geological formations". In: *Computational Geosciences* 17.4 (2013), pp. 671–687.
- [27] A. O'Hagan. "Polynomial Chaos: A Tutorial and Critique from a Statistician's Perspective". In: *Tonyohagan.Co.Uk* (2011), pp. 1–16.
- [28] S. Oladyshkin et al. "An integrative approach to robust design and probabilistic risk assessment for CO2 storage in geological formations". In: *Computational Geosciences* 15 (June 2011), pp. 565–577.
- [29] A. Saxena et al. "Damage propagation modeling for aircraft engine run-to-failure simulation". In: *International Conference on Prognostics and Health Management* (Oct. 2008).
- [30] B. Sanderse et al. "Efficient Bayesian calibration of aerodynamic wind turbine models using surrogate modeling". In: *Wind Energy Science Discussions* 2021 (2021), pp. 1–34.
- [31] A. Saxena and K. Goebel. *PHM08 Challenge Data Set*. NASA Ames Research Center. Moffett Field, CA, 2008.

- [32] L. Peel. “Data Driven Prognostics using a Kalman Filter Ensemble of Neural Network Models”. In: *2008 IEEE International Conference on Prognostics and Health Management* (2008).
- [33] A. e.-d. Riad, H. Elminir, and H. Elattar. “Evaluation of Neural Networks in the Subject of Prognostics As Compared To Linear Regression Model”. In: *International Journal of Engineering & Technology IJET-IJENS* 10 (Dec. 2010), p. 50.
- [34] C. Hu et al. “Ensemble of data-driven prognostic algorithms for robust prediction of remaining useful life”. In: *Reliability Engineering & System Safety* 103 (2012), pp. 120–135.
- [35] P. Wang, B. D. Youn, and C. Hu. “A generic probabilistic framework for structural health prognostics and uncertainty management”. In: *Mechanical Systems and Signal Processing* 28 (2012). Interdisciplinary and Integration Aspects in Structural Health Monitoring, pp. 622–637.
- [36] E. Ramasso. “Investigating Computational Geometry for Failure Prognostics in Presence of Imprecise Health Indicator: Results and Comparisons on C-MAPSS Datasets”. In: *PHM Society European Conference 2 (1)*. July 2014.
- [37] C. Zhang et al. “Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics”. In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10 (2017), pp. 2306–2318.
- [38] G. Sateesh Babu, P. Zhao, and X. Li. “Deep Convolutional Neural Network Based Regression Approach for Estimation of Remaining Useful Life”. In: *Database Systems for Advanced Applications*. Ed. by S. B. Navathe et al. Cham: Springer International Publishing, 2016, pp. 214–228.
- [39] O. Bektas, A. Alfudail, and J. Jones. “Reducing Dimensionality of Multi-regime Data for Failure Prognostics”. In: *Journal of Failure Analysis and Prevention* 17 (Oct. 2017).
- [40] X. Li, Q. Ding, and J. Q. Sun. “Remaining Useful Life Estimation in Prognostics Using Deep Convolution Neural Networks”. In: *Reliability Engineering & System Safety* 172 (Dec. 2017).
- [41] P. Lim, C. K. Goh, and K. C. Tan. “A Novel Time Series-Histogram of Features (TS-HoF) Method for Prognostic Applications”. In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 2.3 (2018), pp. 204–213.
- [42] A. Listou Ellefsen et al. “Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture”. In: *Reliability Engineering & System Safety* 183 (2019), pp. 240–251.
- [43] K. Verbert, B. De Schutter, and R. Babuška. “Timely condition-based maintenance planning for multi-component systems”. In: *Reliability Engineering and System Safety* 159. January 2016 (2017), pp. 310–321.
- [44] R. Bellman. “The theory of dynamic programming”. In: *Bulletin of the American Mathematical Society* 60.6 (1954), pp. 503–515.

II

Literature Study
previously graded under AE4020

1

Abstract

Prognostic methods for aircraft maintenance are becoming more and more popular. Maintenance repair and overhaul companies are shifting their interests from conservative aircraft maintenance strategies (time-based maintenance) towards condition-based maintenance strategies, optimising the useful life of aircraft components and reducing unexpected aircraft-on-ground time. This report provides a thorough literature review on remaining useful life prognostics and how this can be used to optimise aircraft maintenance policies. Using the literature review research gaps are identified.

The literature review starts with an introduction to the current aircraft maintenance strategies; preventive maintenance, corrective maintenance and predictive maintenance. Preventive maintenance is associated with high preventive costs, but low unexpected repair costs. For corrective maintenance, it is the other way around. Predictive maintenance takes the best of both worlds and minimises repair and prevention costs. Prognostics and health management can be used for predictive maintenance, which predicts the remaining useful life of components (the time the component fails). Many different approaches can be considered for determining the remaining useful life. For this review, two approaches are selected and reviewed; particle filtering and polynomial chaos expansions.

First, particle filtering approaches are reviewed. It was seen that particle filtering is a widely used approach in the field of prognostics and has proven to be a useful tool for determining remaining useful life of components. Many different degradation models can be used for particle filtering, but initial information about the initial distribution of parameters is required to be known. Particle filtering has been used for many prognostic applications, among which predicting the remaining useful life of aircraft engines. Also, many different approaches can be taken for particle filtering regarding the sampling and/or resampling algorithm to be used. Polynomial chaos expansions have been used significantly less in the field of prognostics and therefore, other applications in different fields have been reviewed as well. A few papers have been reviewed showing that polynomial chaos expansions can be used for prognostics. However, it has never been used in the field of aircraft maintenance.

Furthermore, a section is dedicated to obtaining the degradation indicator from a data set consisting of implicit sensor measurements of an aircraft engine. Sensor measurements such as the temperature and pressure at different locations in the engine can be used to construct a health indicator which can then be used to predict the remaining useful life of the engine. Different methods are reviewed, such as using principal component analysis to reduce the dimensionality of the data set or using physical equations to model the reduction in engine efficiency.

Finally, the literature review covers different approaches to obtain an optimal maintenance policy which uses the remaining useful life of a component. Stochastic maintenance optimisation and Markov decision process optimisation methods have been reviewed. For aircraft engines, only stochastic optimisation methods using an AI approach to determine the remaining useful life of an implicit multi-sensor aircraft engine degradation data set have been covered in literature. Furthermore, only short-term planning has been considered for this

specific application.

Resulting from the literature review, research gaps have been identified. As was briefly explained above, polynomial chaos expansions have not been used in the field of prognostics for aircraft maintenance yet. Also, no long-term maintenance optimisation has been considered for aircraft engines for a single aircraft or a fleet of aircraft. Therefore, this will be the base of the research to be conducted.

The research aim is subdivided into two parts. The first aim is to develop a stochastic and statistical model-based prognostic model which is able to estimate the remaining useful life of an aircraft engine based on implicit multi-sensor measurement data obtained during the lifetime of that engine as well as quantify the uncertainty of the remaining useful life estimation in the form of a probability density function. The second aim of the research is to develop a maintenance policy optimisation method which is able to implement the result of the prognostic model (the probability density function of the remaining useful life of an aircraft engine) in order to obtain an optimal maintenance policy which reduces long-term maintenance costs compared to other maintenance policies.

Following from this, a main research question is formulated as follows.

Main Research Question

How can RUL prognostics applied to aircraft engines be used to optimise maintenance policies?

This research question will be answered during the research that will be conducted following from this literature review.

2

Introduction

Aircraft maintenance has been necessary since the beginning of the aviation era. Aircraft maintenance is required to ensure that the airworthiness of aircraft is preserved. Historically seen, maintenance repair and overhaul (MRO) companies have had a rather conservative attitude regarding maintenance scheduling during the last century and the current first decade of the current century [15]. During this time frame, components were often replaced at fixed intervals without taking into account the degradation state of the component. This is called a time-based maintenance (TBM) strategy or preventive maintenance (PM) strategy [11]. Currently, MRO companies are continuously trying to optimise their operations and because condition monitoring and modelling techniques have improved significantly in the last decades, the interest of MRO companies is shifting from conservative methods to the investigation of implementing prognostic methods in order to optimise their operations. This is supported by research in academic environments, where the number of publications regarding prognostics and health management have been increasing in the last few years, see Figure 2.1 [25].

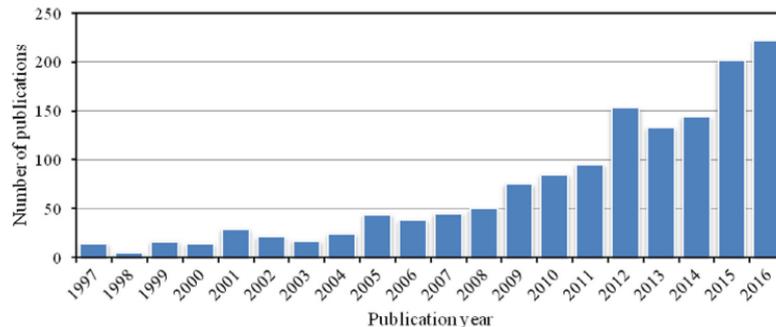


Figure 2.1: Academic publications per year in the field of prognostics[25]

With the increasing availability of big data and operational experience of airlines, MRO companies and aircraft manufacturers, more and more diagnostics and prognostics methods are developed to predict remaining lifetimes of aircraft components. Airbus has indicated that these methods might result in a large decrease in unscheduled ground time due to faults by 2025¹. Using diagnostics and prognostics, aircraft maintenance can be dynamically predicted instead of using periodic maintenance inspections, leading to less aircraft on ground (AOG) time due to unexpected faults, avoiding early replacements leading to less waste and a significant reduction in costs.

Because of this, there is a need in the aviation industry to develop prognostic methods which are able to accurately predict the remaining useful life and the corresponding uncertainty of aircraft components. Furthermore, this result should be used to find an optimal maintenance policy which will minimise cost and unexpected AOG time. For this reason, a literature review is conducted which will review different approaches

¹Retrieved from: <https://www.flightglobal.com/mro/airbus-sees-big-data-delivering-zero-aog-goal-within-10-years/> on 28/04/2021

for prognostics to find the remaining useful life of aircraft engines. Next to this, it is reviewed how sensor measurements of an aircraft engine can be translated to a health indicator of the engine. The review also covers different maintenance optimisation methods. The aim of the literature review is to identify research gaps in existing literature and formulate a research framework in order to address these gaps.

This literature review is structured as follows. In [chapter 3](#) the literature review is presented discussing prognostics, degradation modelling and maintenance optimisation methods. Then, in [chapter 4](#) the research motivation is provided. Here, the research gaps are identified based on the literature review and the aim of the research as well as the research questions are formulated. A Gantt chart is provided in [chapter 5](#), showing the project planning of the research to be conducted. Finally, a conclusion is given in [chapter 6](#).

3

Literature Review

This chapter contains the literature review regarding prognostics and maintenance schedule optimisation. First, literature regarding prognostics will be reviewed in [section 3.1](#). Then, different methods to obtain degradation modelling of multi-sensor systems will be covered in [section 3.2](#). Finally, maintenance optimisation methods are reviewed in [section 3.3](#).

3.1. Prognostics in Aircraft Maintenance

This section introduces the concept of prognostics in a maintenance framework. First, the current situation is discussed in [subsection 3.1.1](#). Then, particle filtering in a prognostic framework is reviewed in [subsection 3.1.2](#) and [subsection 3.1.3](#). In [subsection 3.1.4](#) and [subsection 3.1.5](#), research regarding polynomial chaos expansions is reviewed.

3.1.1. Current Situation

Prognostics is a field in engineering which deals with determining when a certain component of a certain system is no longer able to be used for its intended function. The remaining useful life (RUL) is an important aspect within the field of prognostics; it is the amount of time that a component can still perform its intended function. When the RUL of a component reaches 0, the component can be seen as failed and a reparation or replacement of the component is required in order to operate the system again [53]. Prognostics is of large interest for environmental, operational and economical purposes and therefore, it has been applied in a variety of fields such as the aerospace industry, electronics, civil engineering applications and other industrial machinery applications [24]. Prognostics and health management (PHM) is a term often referred to in literature which describes the discipline which links the remaining useful life of a component to appropriate decision making to maintain the component in time[33]. PHM has been studied extensively by using different techniques such as condition-based maintenance (CBM) and on-condition maintenance (OM). The main maintenance strategies can be identified as preventive maintenance, predictive maintenance (which covers prognostics) and corrective maintenance. The cost and number of failures for these strategies can be seen in [Figure 3.1](#) [52].

It can be seen that preventive maintenance has a high corresponding cost while the number of failures is low, because components are replaced while the RUL might still be relatively large. Corrective maintenance also comes with high cost because this means that the component is used until failure. This results in high costs because unscheduled maintenance must be performed. Predictive maintenance tries to balance these two extremes to reduce costs significantly by maximising the useful life of components and minimising corrective maintenance repairs. It can be seen that predictive maintenance is therefore an interesting strategy to replace the accustomed preventive maintenance strategy in aircraft maintenance.

Within the prognostics framework, three different approaches can be distinguished. These methods are data-driven, model-based and hybrid approaches [53]. Model-based methods depend on the incorporation of a physical model to estimate the RUL. An example of such a model is the physical crack growth model by Paris and Erdogan [40]. Data-driven methods do not rely on any knowledge of the physical behaviour of the com-

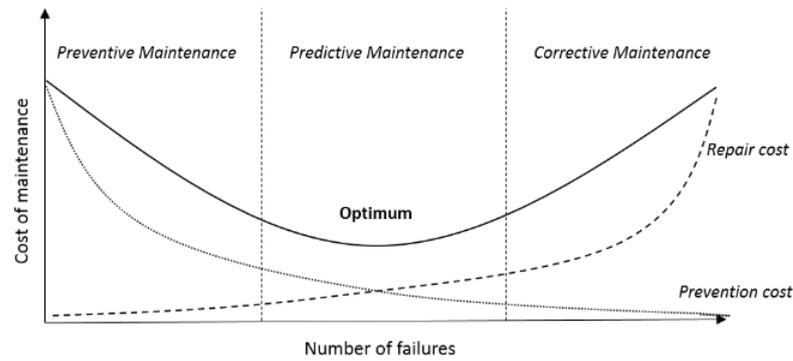


Figure 3.1: Different maintenance strategies with their corresponding relative cost indication and number of failures[52]

ponent which is analysed. They depend largely on measured data. Furthermore, data-driven models can be split into models using artificial intelligence (AI) and models which use statistical and stochastic approaches. Finally, hybrid models combine model-based and data-driven models in order to get the best of both worlds and increase the RUL estimation performance [53].

Many prognostics methods can be established for determining the RUL of a component using information about the state of degradation of the component. Similarity based approaches have been used by [57]. Weibull distribution prognostics is performed by [29]. Wiener process modelling in combination with Monte Carlo simulation has been done by [24] and [16]. (Non) linear regression methods can also be used, this has been demonstrated by [6]. Another frequently used methods are Kalman Filtering or Particle Filtering (PF) [39]. [8] have demonstrated that Polynomial Chaos Expansions (PCE) can also be used for RUL determination in prognostics. Next to the methods named here, there are many more available methods in literature. This literature review will focus on PCE, as little prognostic research has been performed using this method, although it has been applied in different fields in uncertainty analysis. Next to PCE, the main focus will be on Particle Filtering. This method has been widely used in the field of prognostics and might be an interesting and robust approach for the research following from this literature review.

3.1.2. A Brief Revision of Particle Filter Theory

Particle filters, also known as sequential Monte Carlo methods (SMC), consist of a set of algorithms used for solving Bayesian statistical inference problems and filtering problems. It was introduced in 1993 by Gordon et al. [13]. Observations of a certain process, which can be nonlinear, can contain non-Gaussian noise and can be partially observable, are used to generate a set of samples which represent the posterior distribution of the given stochastic process. It is the most widely used method in the field of prognostics [1]. Another method which is often used is a Kalman Filter (KF). A Kalman filter gives the exact probability distribution function of a signalling process which is linear and contains noise from a Gaussian distribution[1]. A brief overview of the principles of PF is given first, after which literature is reviewed using this method in a prognostics framework.

In a prognostics framework, PF is used to estimate the unknown parameters of a certain degradation process of which probability information is available. For this, a state transition function f and a measurement function h are required, see Equation 3.1 and Equation 3.2, respectively [64].

$$x_k = f(x_{k-1}, \theta_k, v_k) \quad (3.1)$$

$$z_k = h(x_k, \omega_k) \quad (3.2)$$

Here, k is the time step index, x_k is the damage state at that time step, θ_k contains the model parameters and z_k is the measurement data. v_k and ω_k are the process and measurement noise. PF uses Bayesian inference to estimate the unknown parameters using observations as a form of the PDF Bayes' theorem is fundamental for Bayesian inference, see Equation 3.3 [3].

$$p(\Theta | \mathbf{z}) \propto L(\mathbf{z} | \Theta)p(\Theta) \quad (3.3)$$

Here, $p(\Theta | \mathbf{z})$ is the posterior PDF of the unknown parameters, $L(\mathbf{z} | \Theta)$ is the likelihood function and $p(\Theta)$ is the prior distribution of the unknown parameters. Each time when a new measurement is available, the posterior distribution from the previous step is used as the prior distribution for the current step. An amount of samples is drawn from the prior distribution. These are multiplied with the likelihood function. The result is given a weight and based on this, the particles are resampled and the posterior distribution is obtained. An overview of this process can be seen in Figure 3.2. When all measurements have been processed, the most up to date estimated model parameters are obtained. These can then be used to predict future degradation levels and the resulting RUL using the model from Equation 3.1 [1].

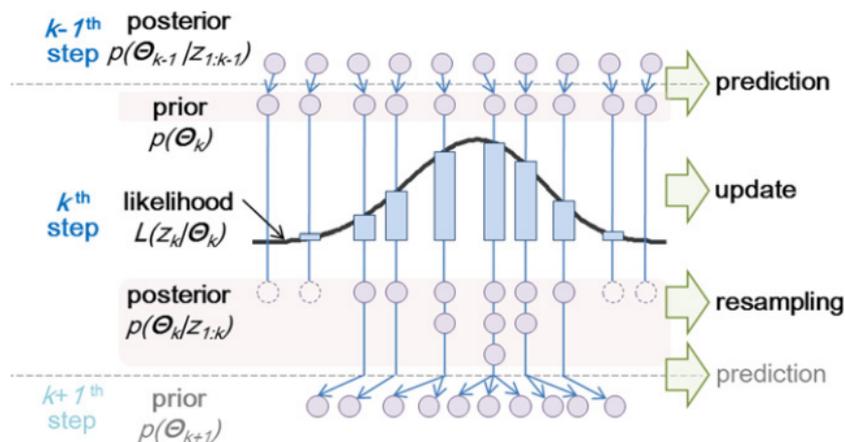


Figure 3.2: Overview of the PF process [1]

3.1.3. Particle Filter Literature Review

A PF approach has often been used in the field of prognostics. An overview of a number of papers researching prognostics and PF can be viewed in Table 3.1. This is only a small grasp of the total number of available papers, however, it gives a wide variety of different applications and degradation models. The remainder of this subsection will elaborate on these papers.

Table 3.1: Literature Review on Particle Filters

Reference	Year	Method	(Re)sampling Method	Degradation Model	Application
Orchard et al. [39]	2005	Particle Filter	ASIS	Assumed crack propagation model	Turbine blade
Zio and Pelsoni [64]	2011	Particle Filter	IS and inverse CDF	Crack SSM	Crack fatigue
Siegel et al. [51]	2011	Particle Filter	ASIS	Exponential degradation	Aircraft engine
Saha and Goebel [46]	2011	Particle Filter	SIR	Empirical charge depletion model	Battery
An et al. [1]	2013	Particle Filter	Inverse CDF	Exponential degradation	Battery
Wang and Gao [56]	2014	Regularized Particle Filter	IS and discard low weights	Exponential/generalized state evolution model	Aircraft engine
Hu et al. [16]	2018	Particle Filter	SIR	Wiener process	Milling machine wear
Li et al. [30]	2018	Adaptive order Particle Filter	ASIS	Historical data	Aviation piston pump
Lei et al. [27]	2019	Particle Filter	Likelihood sampling, fuzzy resampling	Wiener Process Model	Random data set/ Aircraft engine
Gebraeel et al. [28]	2020	Particle Filter	Likelihood sampling, fuzzy resampling	State space degradation with rv degradation rate and Brownian Motion	Random data set/ Aircraft engine
Jiao et al. [18]	2020	Particle Filter	Likelihood sampling, fuzzy resampling	GAP DBN model	Aircraft engine

An et al. [1] provide a tutorial for particle filter and model-based prognostics. A simple example of a battery

degradation model and a crack growth model is presented. The model is combined with measurement data to estimate parameters of the physical models. The paper describes thoroughly how other engineers who are inexperienced with prognostics can use PF for their applications. The paper describes the full process of acquiring measurement data and quantification to RUL prediction using the damage model. It is assumed that the prior information about the distribution of the unknown parameters is known. For the first model, the battery degradation model, 10 measurements have been obtained. Bayesian inference has been applied to estimate the parameters more accurately from the prior distribution. At the start of the algorithm, n samples are drawn from the prior distribution which is assumed to be known. The degradation is predicted for all samples, based on the prior distribution of the parameters. The next step is to update all parameters. This is related to the likelihood of measurement data. It is assumed that the measurement noise is normally distributed and therefore, the likelihood of the measurement can be expressed as the likelihood function for Gaussian distributions. Again, this is performed for all samples. The next step is to resample the samples, i.e. to give high weights to more important samples and low weights to less important samples. For this, An et al. [1] use the inverse CDF method [64]. The cumulative distribution function (CDF) of the former mentioned likelihood function is obtained and numbers from the uniform distribution $U(0,1)$ are drawn n times. Finally, the sample from the $U(0,1)$ distribution is matched with the closest sample in the CDF. This sample is then stored to finally obtain the resampled posterior distribution. When the process is repeated until the final measurement, the most accurate and up to date samples are obtained. These are then used as input to the physical model to predict future degradation values. When the prediction reaches the predefined failure threshold the RUL can be obtained by taking the time difference between the last measurement and the time that the failure threshold has been exceeded. A histogram with the RUL for all n particles is obtained, so the uncertainty and confidence interval can be obtained to evaluate the result. The paper neatly describes the working process of PF in a prognostic framework and gives straightforward examples. With the current information about how PF can be applied in prognostics, other papers can be reviewed and different strategies of different researches can be outlined.

Orchard et al. [39] were one of the earlier researchers who started to research prognostics in combination with particle filtering. They develop a novel approach which employs a state dynamic model and a measurement model to predict the time evolution a crack fault. They also include correction terms in a learning paradigm for improvement of the estimation accuracy. Whereas An et al. [1] use the inverse CDF method to resample particles and give higher importance to frequently occurring particles, Orchard et al. [39] use a method called Auxiliary Sampling Importance Resampling. It is a method derived from the sequential importance sampling method and it has been developed by Pitt and Shephard [42]. Practically speaking the particles are given a weight corresponding with Equation 3.4.

$$w_k = w_{k-1} \frac{p(z_k | x_k) p(x_k | x_{k-1})}{q(x_k | x_{0:k-1}, z_{1:k})} \quad (3.4)$$

Here, $p(z_k | x_k)$ is the likelihood function, $p(x_k | x_{k-1})$ is defined by the model equation (Equation 3.1 and $q(x_k | x_{0:k-1}, z_{1:k})$ is derived from the importance distribution function [2]. The weights are then multiplied with Dirac function in order to come to the posterior distribution [39]. Orchard et al. [39] conclude the paper with a demonstration of their method to a polynomial crack propagation model.

Zio and Peloni [64] propose a similar methodology for estimating the fault prognosis of a crack fault. They employ a state dynamic model and a measurement model to estimate the posterior distribution of the crack fault evolution over time. Again, Bayesian inference is used to estimate the posterior distribution of the system state. Particles are drawn for the PF resampling process. The importance sampling in this paper is performed using Equation 3.5.

$$w_k^i = \frac{p(z_k | x_k^i)}{\sum_{j=1}^N p(z_k | x_k^j)} \quad (3.5)$$

It can be seen from Equation 3.5 that the importance weights are based on the likelihood function of the observations of z . The posterior distribution is then obtained from the weighted samples. Zio and Peloni [64] address the problem of increasing variance of the particle weights over time. Therefore, the importance weight becomes progressively skewed until particles have a negligible weight except for one particle. Poor results are the consequence while computational resources used is still high. Zio and Peloni [64] address this

problem by not taking the obtained weights to the next step, but draw new realisations from the posterior distribution constructed from the obtained weights. The result is that all particles have the same weight again, but the information about the posterior distribution at the current time step is maintained.

An et al. [1] used a model based approach where the unknown parameters are estimated and prognostics is performed by using the model with the estimated parameters. Zio and Piloni [64] do not assume a model with certain unknown parameters, but provide a biased and an unbiased methodology to predict the l -step ahead posterior distribution of the degradation. Essentially the unbiased l -step ahead prediction can be obtained by integrating the model equation (Equation 3.1) and the current state PDF estimate. Swarms of samples until time $k+l$ are drawn and the familiar Monte Carlo PDF approximation method is used in combination with the inverse CDF method to solve the integral. On the other hand, the biased method is based on the maximum likelihood estimation (MLE), where the biased estimator is the probability of the RUL being smaller than the l -step ahead prediction. Both prognostic methods have been applied to a dynamic crack propagation model in the form of a state space model. The unbiased method performs better than the biased method; the median is closer and the confidence intervals are smaller. The results show that the developed methodology is able to predict satisfactory results for a nonlinear dynamic system containing non-Gaussian noise.

Siegel et al. [51] apply PF prognostics in estimating the degradation in aircraft engines. They use a data set from NASA from the 2008 PHM data challenge which contains 100 degradation paths of aircraft engines. The data set consists of 21 measurements obtained during one cycle for the entire lifetime of the engine. So, no explicit form of degradation is present. Since information was provided about where the failure in the engine would occur, the high pressure combustor, the authors of this paper decided to only take into account the sensors corresponding to the high pressure combustor and use these equations relating performance parameters of the high pressure combustor, such as compressor efficiency and compressor pressure ratio. The authors assume an exponential degradation model of the compressor efficiency. A failure threshold is defined as the moment that the compressor efficiency has degraded more than 2% compared to the initial efficiency. The parameters of the exponential function are estimated using the available information of the 100 engines to obtain an initial distribution. Then, they apply the same principles of Bayesian inference as the previously discussed papers. The authors use the same method as Orchard et al. discussed previously to determine the weights of the importance sampling algorithm, as was seen in Equation 3.4 (Auxiliary PF). The posterior density is then updated by multiplying the weight with the Dirac function of the system state.

The results obtained by this method by Siegel et al. [51] are accurate when the prediction is made less than 40 cycles away from the true remaining useful life. However, as only sensor measurements relating to the high pressure compressor have been taken into account for the prognostic model, there is still room for improvement of the estimation accuracy by taking other sensor measurements into account as well. Other methods using this data set have achieved good results for long term (>40 cycles) prediction by using more sophisticated modelling using the sensor measurements [9, 24]. However, a good demonstration about the possibilities to use PF with a rather simple engine degradation model has been provided. Still, the model could be used for accurate short term predictions and maintenance planning.

Goebel and Saha [46] explore the notion of wellness of design of a PF application for determining the RUL of lithium ion batteries. They explain that PF mostly works because particle filters are not subjected to "the curse of dimensionality". This is a common phenomenon for problems with a high state dimension. Exponential growth of computational complexity is the result of the curse of dimensionality. Particle filters avoid this, but in practice, only well designed particle filters are able to escape from the phenomenon. The authors describe different PF model designs subject to different prognostic metric in order to demonstrate how to avoid the curse of dimensionality using PF and how sensitivity analysis may be used to design a good PF prognostic model.

Goebel and Saha [46] explain the theoretical basis behind particle filters escaping the curse of dimensionality. The proposed density function given by the particle samples originate from important regions of the state space that are originating from the integration to obtain the posterior PDF. However, the posterior distribution is not multivariate, non-parametric and unknown beyond a proportionality constant and therefore it is hard to obtain good samples from the posterior distribution [46]. Even when a prognostic problem is considered containing only a low dimensional health vector, extra dimensions to the model are added when

model parameters are added to the state vector in order to track non-stationarity of the model. Therefore, a good choice of the proposal density function must be established [46].

The paper uses the Sampling Importance Resampling (SIR) algorithm to determine the weights of the particles. The equation for the weights can be found in Equation 3.6. Here, $\pi(\mathbf{x}_k^i)$ is equal to the likelihood expression $p(x_k|Z_{k-1})$ and $q(x)$ is the importance distribution which is similar to $\pi(x)$.

$$w_k^i = \frac{\pi(\mathbf{x}_k^i) / q(\mathbf{x}_k^i)}{\sum_{j=1}^N \pi(\mathbf{x}_k^j) / q(\mathbf{x}_k^j)} \quad (3.6)$$

The innovative part of the paper is that model adaptation is proposed including model convergence without incurring the curse of dimensionality. The model is adapted each iteration online, which can come in handy when one model is used for different applications with specific model parameters which need to be estimated in an online fashion. Saha and Goebel introduce parameters of the function f (Equation 3.1) into the state vector. The state equations for the included parameter are obtained using a Gaussian random walk. The proposed PF will converge to the actual parameter value if a suitable starting point is selected. This demonstrates that including an extra dimension still achieves convergence without incurring the curse of dimensionality. The notion of a good proposal density essentially means a good initial estimate of the parameter [46]. Sensitivity analysis can be performed to see which parameter affects the state function most. Instead of a random walk, another method is proposed that takes into account how a change in the parameter value affects the output. Saha and Goebel incorporate the short-term prediction error back into the estimation routine. They demonstrate their online parameter estimation routine to a model-based battery degradation model which successfully is able to estimate parameters using both models. However, they assume that the initial population and prior distribution is known.

Wang and Gao [56] researched a dynamical, non-linear and non-Gaussian system to obtain a real-time degradation prediction method using a regularized particle filter (RPF). RPF overcomes the problem of particle impoverishment. Next to this, they establish a modified system evolution model to model system performance degradation. They start with the establishment of the PF algorithm, which relies on the same principles established as discussed for earlier discussed papers and in subsection 3.1.2. The determination of the particle weights relies on the same principle as the method used by Zio and Peloni [64] described earlier. This depends on the relation between the previous particle weight and the likelihood function. Particles with relatively low weights are discarded to avoid particle degeneracy. The innovative part of the method is the use of the aforementioned RPF method. As particles are drawn from a discrete distribution instead of a continuous distribution, the problem of loss of diversity of particles might occur [56]. Key here, is to change the distribution of the posterior PDF from a discrete to a continuous one. This is performed during the resampling stage with the rescaled kernel structure, relying on a certain specified bandwidth. Furthermore, to account for transient degradation of a system due to a fault which results in a transient change in model parameters and states, Wang and Gao include the prediction error into the state evolution model. If a transient change of output prediction error is obtained between time k and $k+1$, the cost function is compared to a predefined threshold. If the threshold is exceeded, the state transition function (Equation 3.1) is transformed to Equation 3.7 [56]. Here u is the unit step function and γ is related to the cost function.

$$x_{k+1} = x_k + \gamma u + w_k \quad (3.7)$$

Wang and Gao apply the proposed method to the aircraft engine data set from NASA. To model the degradation, they use the same approach as described by Siegel et al. [51] earlier in this section. The method is based on determining the efficiency of the high pressure combustor (HPC) using the sensor measurements. The results of the method are compared to the course of the efficiency of the HPC instead of the actual RUL provided by the verification data set. The results indicate that the transient changes can be tracked within two steps with a low prediction error compared to the actual HPC efficiency degradation [56].

The paper by Hu et al. [16] focuses more on an elaborate application of the developed theory described by the papers above. Hu et al. develop a consistent framework to solve RUL prediction for complex degenerate systems. A state space model is developed. A sequential Monte Carlo algorithm is derived from a Markov chain Monte Carlo algorithm in order to derive the optimal Bayesian estimation. The writers use a hidden Markov Model (HMM) to describe the underlying degradation of a system, with the assumption that the

states can be seen as a first order Markov process. The HMM expression can be seen in Figure 3.3. The PF framework used in this paper follows the earlier explained principles in combination with the "sampling importance resampling" algorithm for the particle weights which was also used by Goebel et al. [46].

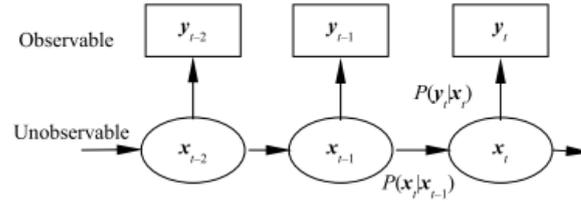


Figure 3.3: Relation between states and observations in a HMM framework

A discrete-time state space model using a Wiener process has been used to model the degradation of cutters in milling machines. The PF algorithm will estimate the unknown Wiener process parameters; the drift constant and the diffusion coefficient of the Brownian motion. The parameters are updated online simultaneously with the system state. The RUL is then predicted using the latest system parameters and state information. The paper provides a clear framework for the application of PF using the developed methods described in other literature.

Li et al. [30] propose a method called adaptive-order particle filtering (AOPF). The key idea behind this approach is to combine a model-based method with a data-driven method, to get the benefit of both worlds. They apply the model to a complex system, an aviation piston pump, for which it is very hard to obtain a physical model. A data-driven model would not suffice, as long term prediction accuracy is poor for their application. Therefore, they propose the AOPF method which combines the two methods. The PF framework by Li et al. rests on the same methods described above, with the use of Auxiliary sampling importance sampling as was outlined by Orchard et al. [39]. The innovative part of the method however, is the adaptive-order PF framework. The application discussed in this paper relies on oil flow data. The oil flow is gradually increasing, which is an indication of degradation. However, the oil flow is not monotonously increasing. Depending on the order of the model and the last observations, the next predicted step might be a decrease or an increase of oil flow, see Figure 3.4. A first order model would predict x_{k+1} , while a second order model would predict x'_{k+1} .

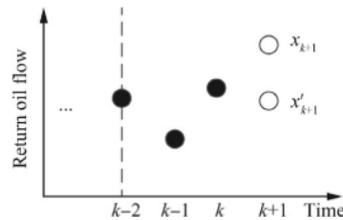


Figure 3.4: Possible predictions of the oil flow depending on the model order [30]

The adaptive-order PF means that the order of the dynamical model is updated each step (order as in how many previous measurement points to take into account). The largest difference with a regular PF is that during the prediction step of a model of order $O_k = p$, the states of particles $\{x_{k-p+1}^i, x_{k-p+2}^i, \dots, x_{k-1}^i\}$ should be stored without resampling the particles [29]. The adaptive order framework determines for each time step the model based on historical data up until the current time. The accumulated errors are obtained for the model and if the errors are smaller for a higher order model, the model is updated. The result is a more accurate short-term prediction. The proposed method is verified using the aviation piston pump. The results show that the proposed method performs better by 42.5% compared to a traditional gray forecasting method. It has been demonstrated that AOPF efficiently fuses the available empirical data with new data points to a model with a new order.

Lei et al. [27] developed a PF method which considers unit to unit variability (UtUV). UtUV is the phenomenon of similar systems of the same population which show different degradation processes because of different operating conditions and health states. A Wiener process model is developed in this paper to describe the different degradation processes of different units. An age and state dependent Wiener process model is designed to describe the UtUV by adding a random parameter to the Wiener drift parameter to account for variation in degradation rates in different units. This random parameter is estimated using maximum likelihood estimation (MLE) for each unit separately, after which the discrete distribution of the parameter for all units can be obtained. The UtUV parameter is updated online by the PF algorithm in combination with a fuzzy resampling algorithm. Next to the UtUV parameter, two other parameters are estimated using MLE. This is performed during offline training of the model, the different complete degradation processes are used for training and estimation of the parameters. The weight of the particles is determined by multiplying the previous weight with the likelihood function and normalising. Then online testing takes place, where the PF algorithm updates the estimated parameters based on the degradation measurements. This will then be used to estimate the RUL using the Wiener Process model.

A fuzzy resampling algorithm is developed for resampling of the particles. The resampling algorithm has similarities with the inverse CDF method described by [1], however, this method is expanded to deal with impoverishment of the particles. Loss of diversity of particles is a problem which arises if impoverishment is not dealt with accordingly. Therefore, after the inverse CDF method is applied, a random noise factor following a Gaussian distribution is added to the particles. This results in diverse particles which hardly consist of duplicates. The method is applied to the degradation of 100 fatigue crack growth units. 90 units are used for training and 10 for testing. The method shows that the fuzzy resampling technique shows very good results as can be seen in Figure 3.5. Without using fuzzy resampling, the UtUV parameter estimation does not converge correctly, because of lack of particle diversity. This problem is omitted by using the fuzzy resampling technique.

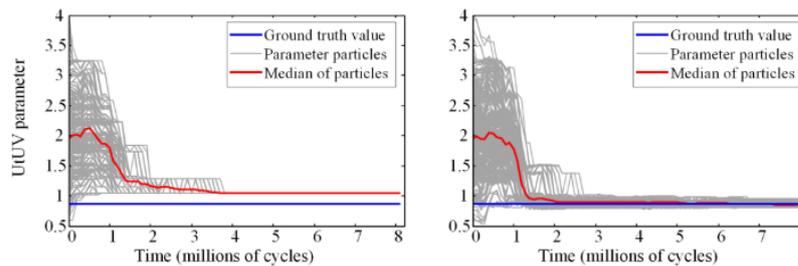


Figure 3.5: Estimation of the UtUV parameter using regular PF (left) and a fuzzy resampling PF (right) [25]

The paper concludes with applying the developed method to an engine degradation set facilitated by NASA, which was also used by [56] and [51]. Instead of using equation for the HPC efficiency, a health indicator is established which fuses the different sensor measurements using a linear weighting model. The method will be elaborated upon in section 3.2. The model shows good results using the fuzzy resampling algorithm and using a health indicator described according to a Wiener process model based on the fusion of different sensor measurements.

Lately, PF is still being used as a prognostic method. Two recent papers will be reviewed which have been published last year. The first one being from Jiao et al. [18]. They develop a novel fault monitoring and RUL prediction framework for multiple fault modes instead of just one, as was done with the previously discussed papers. The method is applied to the same NASA engine data set to verify the method. The major contribution of this paper is that the method is able to predict and identify the type of fault. The paper combines AI and a statistic and stochastic method for RUL prediction. AI, in the sense that a GAP Deep Belief Network (GAP DBN) is used to extract the hidden degradation process of the component. An exponential expression in combination with PF and an automatically selected failure threshold is used for RUL prognostics to model the degradation result obtained from the GAP-DBN model and estimate the RUL.

The GAP metric can be seen as a distance characterisation between different parameters in the Riemann space, ranging from 0 to 1. If the metric is equal to 0, it means that the combination of variables used have

completely similar characteristics. PF is used here to recursively estimate the posterior PDF of the RUL. The initial weights of the particles are set equally (i.e. $1/N$) and the weights are updated by applying the likelihood method, where the maximum likelihood function is multiplied with the previous weight and normalised, as was also the approach from Lei et al. [25]. Resampling is performed by duplicating particles with large weights. This might result in impoverishment as was shown by Lei et al. [25]. The GAP-DBN framework exposes different relations between the extracted features. This gives a clear implication of the fault mode that is occurring. The method is verified using the NASA engine data set and compared to other published methods. It shows very good results compared to other methods while also being able to distinguish the fault modes.

The final paper that will be discussed is a recent paper by Gebraeel et al. [28]. As there is an increasing number of available data of different degrading components in various industries, there is also need for a method which is able to obtain the hidden degradation of such components based on measurement data. Often, the measurement data consists of various measurable characteristics, but the relation between these and the actual degradation is not so obvious. Therefore, Gebraeel et al. provide a multi-sensor fusion model which is able to select the most important sensors and obtain a degradation indicator from these. A Wiener process is used to model the degradation, for which the parameters are estimated using the PF algorithm.

The described method assumes that the input to the model will be a set of sensor measurements for different component degradation trajectories from which no explicit degradation information is known. The state transition function is modelled as a Wiener process and the measurement function is developed as a multivariate measurement function in state space form, relating the state, scale and location parameters, a general expression function, a parameter vector and the noise matrix. The unknown parameters of the state and measurement function are estimated using MLE using the data of the known trajectories for off-line training. During the testing phase, PF is applied to update the parameters and give an estimation of the posterior distribution. The particles are given a weight, which is based on the multivariate measurement function. The particles are resampled using the fuzzy resampling algorithm which was also used by Lei et al. [25]. The median results of the particles of the drift parameter and state estimation are employed as the current estimation. The particle weights combine the results of the measurement function in the state function for degradation modelling, so essentially the particle weight forms the basis of the multi-sensor data fusion model (MSDFM), as it combines the measurements of different signals into one weight value for a certain particle. Furthermore, a sensor selection algorithm is developed. The method is verified using a self developed data set and the 2008 PHM challenge engine data set. The fusion and selection method scores better than no selection and no fusion at all, however, the performance is unknown with respect to other papers also using this data set.

3.1.4. A Brief Revision of Polynomial Chaos Expansion Theory

Polynomial chaos expansion (PCE) is in contrary to PF a non-sampling method to determine the evolution of uncertainty of a certain system with probabilistic uncertainty in the system input parameters. In literature, PCE is often used for models which require very high computational effort to run. Instead, the model can be represented by a set of simpler equations, polynomials, which are faster to evaluate than the original system. This is also called a surrogate model. Polynomial chaos was first introduced by Wiener in 1938 [59]. Wiener developed the PCE method to model Gaussian uncertainties as homogeneous chaos. It involves quantifying uncertainty of stochastic quantities as orthogonal polynomials of random variables [8]. Applying PCE in the field of prognostics is interesting, as other methods such as the earlier described PF or Monte Carlo simulations are computationally demanding for acceptable accuracy. PCE is a considerably faster alternative and because it has been hardly applied in the field of prognostics, it is an attractive approach to investigate [8].

Originally PCE was used for modelling of uncertainties of Gaussian distributions. The expansion is based on Hermite polynomials which are optimal for random variables following a Gaussian distribution. However, often random variables in certain processes cannot be modelling through a Gaussian distribution. Therefore, the PCE method has been expanded to generalised polynomial chaos (gPC). This approach allows to also include uncertain random variables following a uniform, gamma, beta distribution etc. The corresponding polynomial can be found in the Askey scheme [61]. The method can be divided into a non-intrusive and an intrusive method. The intrusive approach requires manipulation of the governing equations of the model that is considered. This might lead to complex and cumbersome solutions, which is why the non-intrusive

approach is gaining popularity [8] as this method does not require this. An example of an intrusive PCE technique is the Galerkin technique which originated from structural mechanics and has been applied in modelling flows [34]. On the other hand, popular non-intrusive methods are the sparse quadrature approach [22] or the probabilistic collocation approach [26]. From the gPC approach a novel approach has been developed called arbitrary polynomial chaos (aPC). This approach is able to process arbitrary distributions as input to a model. The aPC method relies on the fact that an orthogonal polynomial set can be constructed from the statistical moments (mean, variance, etc.) of the underlying distribution [37].

A very brief theoretical overview of the basics of PCE will be provided now. Define a vector $\xi = \{\xi_1, \dots, \xi_n\}$ of independent random parameters with a PDF $\rho = \prod_{i=1}^n \rho_i(\xi_i)$ and the support of ξ as $\Gamma = \prod_{i=1}^n \Gamma_i$. The PCE of order P containing n different random variables is constructed as can be seen in Equation 3.8. The n variate polynomial basis is constructed by tensorising the 1D orthonormal basis in the space Γ [8, 36].

$$y(t, \xi) = f(t, \xi) = \sum_{l=1}^M f_l(t) \Phi_l(\xi) \quad (3.8)$$

Here, $f_l(t)$ are the time dependent coefficients, $\Phi_l(\xi)$ is the polynomial basis and M is dependent on the number of independent variables and the order of expansion. The coefficients are for example obtained using Gaussian quadrature for the gPC non-intrusive method. The mean and variance of the model output can be evaluated from the coefficients. The obtained model can be used for simulations experiencing uncertainty. Another often used application of PCE is sensitivity analysis. An important notation should be made about the input parameters. The input parameters for which PCE is performed should be statistically independent, or their correlation should be removed [38].

3.1.5. Polynomial Chaos Expansion Theory Literature Review

Compared to Particle Filtering approaches, polynomial chaos expansions have hardly been used in the field of prognostics. Duong and Raghavan (2017) [8] are one of the first who address the prognostics problem with the PCE method. They develop an uncertainty quantification method for batteries by using PCE and demonstrate that PCE can be applied in the field of prognostics. However, apart from Duong and Raghavan there has not been performed much research which combines prognostics and PCE and therefore, this literature review explores other fields as well to better understand the possibilities of PCE and how it has been applied in other fields for uncertainty quantification as well. Table 3.2 shows a set of interesting papers which are of value to determine whether PCE can be used as a method of uncertainty quantification in the field of prognostics.

Table 3.2: Literature Review on Polynomial Chaos Expansions

Reference	Year	Method	(Degradation) Model	Application	Intrusive	Coefficient Method
Prabhakar et al. [43]	2010	gPC	Vinh's equations for hypersonic flight dynamics	Flight dynamics uncertainty estimation	Intrusive	Galerkin
Oldayshkin and Nowak [37]	2012	aPC	Exponential decay function	Random system	Both	Galerkin/Collocation
Oldayshkin and Nowak [38]	2013	aPC with Bayesian updating	Darcy's Law	CO2 storage in Geological formations	Non-intrusive	Collocation method
Zhao et al. [63]	2013	gPC with Bayesian updating	Paris' law for crack propagation	Gear	Non-intrusive	Sparse grid collocation
Duong et al. [8]	2017	aPC	Exponential degradation	Battery	Non-intrusive	Gaussian Quadrature
Hawchar et al. [14]	2017	Sparse PCE with PCA	Time variant Von-Mises stress function	Cantilever beam	Non-intrusive	Sparse regression
Casado et al. [5]	2017	aPC	Aircraft motion model	Aircraft trajectory prediction	Non-intrusive	Collocation method
Nagel et al. [34]	2020	Sparse PCE with PCA and Bayesian updating	Water management model	Urban drainage simulation of rain	Non-intrusive	Least angle regression

The PCE literature review will start with aforementioned paper by Duong and Raghavan [8]. Duong and Raghavan tackle the problem of using model-based prognostics with a PCE approach. The motivation to perform research in this area is that PCE has been shown to provide a high accuracy of results and a low computational effort compared to Monte Carlo methods. They use the moment-based polynomial chaos approach (also known as the aPC approach), for the following reasons. First, the moment-based approach avoids the need to assign parametric probability distributions and it can be used with arbitrary distributions. The distributions can be specified as a histogram or as raw data. Finally, sensitivity analysis is straightforward and can easily be used to indicate the dominant parameters of the model.

The moment-based PCE method allows the construction of the orthogonal polynomials and the corresponding quadrature from the moments of the underlying distribution, which is the main idea described by Duong and Raghavan. The k^{th} order raw moment of a random variable can be expressed with Equation 3.9, where the symbols have been elaborated upon in subsection 3.1.4. The raw moments corresponding to the input data are stored in the Hankel matrix. The Hankel matrix is decomposed because it is positive definite and the decomposed matrix is used recursively to construct the underlying distribution of uncertain parameter ξ_i . From the recursive coefficients, the Jacobian matrix is established. The points of quadrature are then the eigen values of the Jacobian matrix, which are then used to solve for the coefficients of the orthogonal polynomials of the expansion.

$$\mu_{ik} = \int_{\Gamma_i} \xi_i^k \rho_i(\xi_i) d\xi_i \quad (3.9)$$

The method is applied to three different examples relying on an exponential degradation function, the first one being a simple exponential function and the second and third are a summation of two exponential functions consisting of different parameters which simulate the degradation of a battery. The examples are verified and compared with Monte Carlo simulations. It has been demonstrated that PCE models the battery degradation uncertainty at different moments in time accurately while only having to run 81 simulations, compared to 5000 Monte Carlo simulations to obtain the same level of accuracy. PDFs of the battery can be obtained for moments in time and the uncertainty propagation can be modelled as such. Also, sensitivity analysis based on the moment-based approach is performed to see which degradation factor is most dominant. It is stressed that the number of uncertain input parameters should be smaller than 6, because otherwise the computational time of the aPC method increases exponentially and a compressed sensing approach should be used to obtain the expansion coefficients. The paper demonstrated that PCE might be a practical approach for uncertainty quantification in the field of prognostics [8].

The moment-based aPC method described by Duong and Raghavan above has been developed by Oladyshkin and Nowak [37]. They state that only moments of probability measures are required for aPC to work and no additional information is required. The research demonstrates that this is indeed the case by providing a thorough theoretical proof of the concept and a more practical example. After the proof of concept, Oladyshkin and Nowak establish 6 properties of moment-based aPC polynomial basis, which can be seen in Table 3.3.

Oladyshkin and Nowak [37] apply their developed concept to the exponential decay function which is a differential equation containing one uncertain input parameter. Two sources of the input parameters will be used: one of raw data and one of a PDF fitted to the raw data. Intrusive and non-intrusive methods are used to obtain the coefficients of both sources, respectively the Galerkin method and the collocation method. From the results it becomes apparent that the method assuming the PDF and not having raw data as input has a slower convergence than the moment-based aPC method. It has been demonstrated that fitting a PDF to the input data leads to errors for higher order expansions, while using raw input data avoids this.

Oladyshkin and Nowak [38] continued with their moment-based PCE approach discussed above by including Bayesian updating via Bootstrap filtering, which is filtering technique similar to particle filtering discussed in subsection 3.1.3. In short, the PCE model is calibrated using history matching and updating each time step a new measurement is obtained. Two main steps can be identified in the approach. The first step consists of obtaining a mathematically optimal response surface via the aPC technique which can be seen as the surrogate model. The second step consists of matching the model to observation data through the bootstrap filter. The method is demonstrated on the storage of carbon dioxide storage in geological formations.

Table 3.3: Properties of the polynomial basis of the moment-based aPC method[37]

Property 1	The orthonormal basis can be constructed without any hierarchical conditions or recurrence relations
Property 2	Existence of the moments μ_0, \dots, μ_{2d} is the necessary and sufficient condition for constructing an orthonormal basis $\{\Psi^{(0)}, \dots, \Psi^{(d)}\}$ to degree d , together with the condition that the number of supports points of x is greater than d if x is a discrete variable or is represented by a data set.
Property 3	The orthonormal polynomial basis for arbitrary probability measures is based on the corresponding moments only, and does not require the knowledge (or even existence) of a probability density
Property 4	All the zeros of the orthogonal polynomials are real, simple and located in the interior of the interval of orthogonality [1]. This property is useful for numerical integration, especially for bounded distributions.
Property 5	As particular cases, the Hermite, Laguerre, Jacoby polynomials, etc. from the Askey scheme and the polynomials for log-normal variables by Ernst et al. [14] can be reconstructed within a multiplicative constant.
Property 6	All distributions that share the same moments up to order $2d$ will also share the same basis, and thus will lead to identical results in an expansion up to order d .

When the surrogate model is obtained using moment-based aPC, Bayesian updating can be performed using Bayes' theorem, relating the likelihood function, the prior PDF of the model parameters and the prior probability of the measurement function. Oladyshkin and Nowak assume that measurement errors are independent and follow a Gaussian distribution. As was also done with PF, a number of samples from the prior PDF is drawn. These are multiplied with the corresponding weight which is determined from the likelihood function. Samples are resampled using the inverse CDF method which was also used in by An et al. [1] for PF. The sample with the highest weight is taken, and added as a new collocation point after which it is evaluated again and coefficients are determined again using the collocation method. This leads to an improvement of the solution by taking into account new measurements as well.

The authors demonstrate the method using a carbon dioxide leakage model in geological formations using Darcy's law, with three uncertain parameters of which some information about the prior distributions is known. The authors state that the approach works best when the PDF of the parameters is known, but if the prior guess is inaccurate and offset against the posterior, iterative Bayesian updating is proposed. This increases the accuracy of the expansion at relatively low computational cost. Oladyshkin and Novak show that a combination of aPC and Bayesian updating for model calibration can be used for any application because no specific properties are required by the forward model [38]. Because of this reason, this paper might provide an interesting approach to apply in the field of prognostics using measurement data of degrading components.

Prabhakar et al. [43] use PCE to model uncertainty in hypersonic flight dynamics. The evolution of uncertainty in state trajectories is modelled for a hypersonic aerial vehicle which will enter the Mars atmosphere and land on the planet. The initial conditions of this vehicle such as L/D coefficient, ballistic coefficient and atmospheric density are the unknown parameters for which the uncertainty must be modelled. The method they use is generalised PCE for stochastic differential equations.

After the gPC framework is developed for differential equations, the authors apply it to a simple first order dynamic system of the form $\dot{x} = Ax$. First, the authors transform the system of stochastic dynamics into deterministic dynamics in higher dimensional state space using the chaos expansion. As the authors deal with the gPC framework, they use Galerkin projection (an intrusive method) to solve for the unknown coefficients of the expansion. The system contains three unknown initial conditions following a Gaussian distribution. It is shown that the gPC method is able to exactly predict the resulting system behaviour if the proper set of orthogonal polynomials is used. Following from this, the method is applied to Vinh's equations describing hypersonic flight dynamics with three unknown parameters which are assumed to be uniformly distributed.

The equations are transformed to deterministic differential equations using the gPC framework. 10% variation in the initial conditions is used to model uncertainty and the distribution is modelled either as a uniform or as a Gaussian distribution. Results show that the uncertainty propagation is as accurate as Monte Carlo simulations but the computational time is reduced by a factor of 60 [43].

Prabhakar et al. propose a different method (gPC) than the methods described earlier (aPC). The reason for this, is that Prabhakar et al. create a surrogate model from a physical model of which the initial conditions are uncertain, but the distributions of these are known. In contrary, the method described by Oladyshkin and Nowak [37, 38] does not assume any known distribution, but the method can model uncertainty from any input data because no specific properties are required by the forward model as was mentioned earlier.

Zhao et al. [63] propose a stochastic collocation approach based on the gPC framework for gear health prognostics. Next to this, they also include a Bayesian inference approach to integrate condition monitoring data of the gear, as was also done by Oladyshkin and Nowak [38]. However, the difference is that Zhao et al. use the generalised PCE approach instead of the arbitrary approach developed by Oladyshkin and Nowak. The goal of the paper is to model the uncertainty of the RUL prediction for the gear and reduce the uncertainty during inspection times by including condition monitoring data.

The gPC framework is outlined and the method to find the polynomial coefficients is discussed. For this, Zhao et al. use the collocation method on a sparse grid. Important in the collocation method is to find the selection of optimal nodes to evaluate the polynomial in. Gaussian quadrature can be used in the one-dimensional case or tensor products for the multi-dimensional case. However, this is not feasible for high dimensions as the method will be subject to de curse of dimensionality as was also discussed in subsection 3.1.3. Therefore, Zhao et al. propose the sparse grid collocation method which avoids this problem. The key idea behind this method is to sum low-order tensor products established using one-dimensional quadrature. This results in a reduction of nodes compared to high-dimensional tensor products.

The method is applied to a crack propagation model described by Paris' law, with uncertainties in the measurement error, model error and input parameters (two material random parameters). The input parameters are samples from a normal distribution which are initially estimated through historical data for the entire gear population and updated each inspection time using Bayesian updating, leading to more accurate parameter estimations and less uncertainty in these. The effect of increasing the polynomial degree and collocation points is investigated and the method is compared with Monte Carlo simulation. The gPC method is more than 100 times faster than MC approach with an increase in accuracy. The authors conclude that the developed stochastic collocation gPC method is a good approach for prognostics based on physical models and can be applied in various fields.

Hawchar et al. [14] propose a method which combines principal component analysis (PCA) with PCE for time-variant reliability problems. An instantaneous performance function is established at each time node of a discretised time frame. PCA is applied to these functions to obtain a reduced number of dominant components of these functions. The components are then approximated using PCE to obtain a faster surrogate model. The obtained surrogate model is used with Monte Carlo simulation in order to evaluate the evolution of the probability of failure of a system over time.

The sparse gPC method is used for expansion. The advantage of sparse PCE compared to regular PCE is that a reduced number of terms of expansion is required and less evaluations of the deterministic model are required. To determine the coefficients of the resulting polynomial, the physical model is evaluated to obtain the exact response of the model for a set of samples from a set of input random variables. These are stored in a vector and subtracted from the sparse PCE model. The resulting equation is solved using sparse regression. The method is a non-intrusive method.

The sparse PCE method is applied to the degradation of a cantilever tube structure, which is subject to time variant torques and forces and other deterministic parameters. The degradation is modelled for 5 years, with a time discretization step of 6 weeks, resulting in 41 time nodes and thus 41 instantaneous responses. As this is rather high and makes sparse PCE inhibitive to perform one sparse PCE per performance function along the 41 time nodes, PCA is applied to these 41 functions to represent all 41 response functions by a reduced

number of non-physical components [14]. This results in 14 components for the tube for which the sparse PCE is then performed. Once the model is obtained, MC simulation is performed to simulate several instances and develop the probability of failure within the given 5 years. Hawchart et al. have demonstrated that PCE can also be applied in cases where time-dependency is introduced explicitly and through stochastic processes.

Polynomial chaos expansions have also been applied in predicting the uncertainty of aircraft trajectories. This has been performed by Casado et al. [5]. Casado et al. apply arbitrary PCE in order to obtain a fast uncertainty quantification model. Many uncertainties in aircraft trajectory are identified of which three are chosen to consider for the model: flight technical uncertainties, earth surface model and the gravitational model. A non-intrusive approach is formulated to solve the coefficients for the multivariate aPC model. For this, the probabilistic collocation approach is used, which was also used by Oladyshkin and Nowak [37, 38]. In total, 10 sources of uncertainty with varying distributions are used for the analysis. Because this number is high compared to number of uncertain parameters modelled in previously discussed papers, the number of polynomial expansion factors is rapidly increasing for higher order expansions. A third order expansion results in being only 10 times faster than Monte Carlo simulation, while Zhao et al. [63] and Prabhakar et al. [43] reached even 60 to 100 times faster computational performance for higher order expansions. However, the method still provides a fast solution with good accuracy for relatively low order of expansion.

The final paper listed in Table 3.2 by Nagel et al. [34] also combines PCA with polynomial chaos expansions as was also performed by Hawchar et al. [14]. The goal of the paper is to identify unknown parameters and quantify measurement and prediction errors for an urban drainage system and applying a sensitivity analysis for these parameters. The original model for the simulation is computationally heavy and PCE is used to have a fast uncertainty quantification. Also, Bayesian inference is used to update posterior distributions.

The authors use a water management model applied to an urban area. The model consists of 8 unknown hydrological parameters. The original simulation model is run 2000 times and observations over a 15 hour rain period are obtained for training of the model, which is discretised in 600 time steps. In the case of the authors, this would mean that 600 different surrogate models should be evaluated simultaneously, which is a computational burden for fast analysis. Therefore, they apply PCA to reduce the dimension of the output of the problem. PCA is mainly used to reduce the number of features in a large data set, however, it has been shown that it can also be used in reducing the number of output dimensions. 9 principal components capture 99% of the total variance of the model output. The next step that is applied is to apply PCE to the obtained components. Hawchar et al. [14] applied sparse PCE in combination with PCA, which is also the approach that Nagel et al. are using. However, a different approach is used to obtain the expansion coefficients; the non-intrusive least angle regression method is performed for this. Once the PCE has been computed for each principal component, the vector containing the model outputs is approximated. Global sensitivity analysis is then performed to the surrogate model, which can be efficiently done using Sobol' indices for PCE by post processing the expansion coefficients, which Duong and Raghavan [8] did as well in a prognostics framework as was discussed earlier in this subsection. The performed sensitivity analysis by Nagel et al. allows to identify the most dominant uncertain input parameters. Finally, two models are developed which use Bayesian updating to update the posterior distributions of the 8 uncertain input parameters based on the observed data by means of Markov Chain Monte Carlo simulation. The uncertainty is decreased significantly, however, the model still does not align with the data completely. This is accounted for by the estimation of a discrepancy function and applying it to the posterior distribution. It is concluded that the synergy potential of PCA and PCE has not been studied extensively, but shows promising results as computational time is reduced significantly and most of the information is preserved after applying PCA [34].

3.2. Degradation Modelling of Multi-Sensor Systems

This section contains a literature review on different types of degradation modelling of systems providing sensor measurement data but no explicit information about the degradation of the system. This can be obtained through various methods, which will be discussed in subsection 3.2.2. Before the methods are discussed, the available data set is briefly introduced in subsection 3.2.1.

3.2.1. Introduction to Data

A commonly used data set used for research in the field of prognostics is an engine degradation data set provided by NASA's Prognostics Centre of Excellence (PCoE). The data set was originally created for the PHM 2008 challenge, a competition organised by the PHM society. The data set was created by simulating several engine trajectories from the start of lifetime until the end of lifetime of an engine. The time is measured in flight cycles and for each flight cycle a set of engine sensor measurements is captured. In total, 21 sensor measurements are taken for each flight cycle and three different operational settings are defined, resulting in 6 different operational modes. Furthermore, 4 data sets for training are provided and 4 for testing. The RUL of the test data sets is provided in another data set for verification purposes. 2 data sets contain 1 failure mode and the other 2 contain 2 failure modes, failure of the high pressure combustor and failure of the fan [48].

As there is no explicit sensor measurement which measures the system degradation, a method needs to be defined in order to obtain the underlying system degradation from the sensor measurements. Ultimately, the information of useful sensors should be combined into a degradation indicator which runs from 0, system is in a perfect state, to 1, system has failed. Because of this, there is a need to review literature which tackles problems such as how to model degradation of a system through its sensor measurements and how to construct a degradation indicator using these sensor measurements. The literature review regarding this is provided in [subsection 3.2.2](#).

Furthermore, some papers include the performance of the model by means of the 2008 PHM challenge scoring function as seen in [Equation 3.10](#). Here, d is the difference between the actual RUL and the predicted RUL. It can be seen that late predictions are penalised more. The scores can be used to compare the performance of different approaches.

$$S = \sum_{i=1}^N S_i, \quad S_i = \begin{cases} e^{-d_i/13} - 1, & d_i \leq 0 \\ e^{d_i/10} - 1, & d_i > 0 \end{cases}, \quad i = 1, \dots, N \quad (3.10)$$

3.2.2. Current Solution Methods

The engine degradation data set has been used often since its publication in 2008 for data-driven prognostics where no prior information about the system is known. Often when developers are designing a prognostics model, it is hard to get high quality data of the system they are considering. Companies are hesitant with making data of their systems open source and therefore, the artificial degradation data set from NASA offers a great alternative for developing new methods, as the data is open source and it can be used for verification purposes. A review of different methods for degradation modelling is performed and the reviewed papers can be found in [Table 3.4](#). The score refers to the PHM data challenge scoring function seen in [Equation 3.10](#) and "Direct/indirect HI" refers to whether an explicit form of the health indicator is obtained or not.

The review will start with three papers dating from 2008 which result from participation to the competition. Peel [41] describes the winning method of the challenge. Peel uses an ensemble of regression models to model the remaining useful life. This was done by using a neural network using Multi-Layer Perceptron (MLP) and Radial Basis functions (RBF). This results in several regression models which are fused using a Kalman Filter, which can be used for similar applications as Particle Filtering, as was discussed in [subsection 3.1.3](#). The approach is to first explore the data and create a set for training the model and for validating the model. Neuroscale visualisation concludes that 6 clusters are identified, corresponding with 6 operational modes. Peel normalises the data set to obtain a solid scale for all features in the data set. After obtaining the MLP and RBF model, it became clear that the RBF underestimates the RUL and the MLP model overestimates the model. Also, previous predictions are not taken into account for new predictions. Therefore, the two models are ensemble by using a Kalman filter. The mean squared error reduces significantly when using the ensemble and the Kalman filter. Peel develops an AI solution for modelling the RUL, however, for this literature review the focus will be on developing degradation models using statistical and stochastic approaches and therefore, this will be the main focus for the remainder of this section.

Coble and Hines [6] divide prognostic models in three types, the first being the use failure time data. Second, stress-based prognostic models are used which also take into account environmental stresses (e.g. temperature or load). Finally, Coble and Hines define effects-based prognostic models, which take into account the specific usage of the component. Often, this involves generating a degradation measure and a path which

Table 3.4: Literature review on degradation modelling

Reference	Year	Method	Degradation Model	Fault Modes	Operational Modes	Score	Direct/indirect HI
Peel [41]	2008	Ensemble of a MLP and RBF neural network using a Kalman Filter	-	Single	Multiple	984	Indirect
Coble and Hines [6]	2008	General Path Model with Bayesian updating	Average HI based on similar sensor output	Single	Multiple	2500	Direct
Wang et al. [57]	2008	Regression and similarity-based matching	Linear and exponential regression model	Single	Multiple	5600	Direct
Giantomassi et al. [12]	2011	Hidden Markov Model i.c.w. an ANN	-	Single	Multiple	-	Indirect
Siegel et al. [51]	2011	Particle Filter with exponential degradation	Exponentially degrading compressor efficiency	Single	Single	-	Direct
Le Son et al. [24]	2013	HI construction using PCA and model as WPM	Wiener Process Model	Single	Multiple	5520	Direct
Fang et al. [9]	2017	MFPCA and a (log)-location-scale functional regression model	-	Single	Single	-	Indirect
Wen et al. [58]	2020	Genetic Programming for HI	Exponential function	Single	Single	-	Direct
Li et al. [29]	2020	State space model with Particle Filter	Wiener process model	Single	Single	-	Direct

evolves over time until a certain threshold is reached. For each type they develop a model and apply it to the PHM 2008 challenge data set.

Coble and Hines outline one or more models for each previously mentioned prognostic type. The first type using failure data can be modelled using a Weibull distribution to model the failure rate of a component. For stress-based prognostic models, proportional hazard models can be used. The model uses covariates in which environmental information is stored to create a new hazard rate each time step based on the baseline hazard rate. Another option is to use Markov chain models for this type of models. For effects-based prognostics, Markov chain-based models, shock models and general path models are outlined to obtain degradation information from data sets and use this to create a degradation indicator. Bayesian updating is introduced as well for integrating historical data and newly acquired data. The three types are applied to the data set and the failure data and stress-based models perform inadequate. For the third type, two models are developed. First, a set of sensors showing similar behaviour are combined resulting in an average parameter. A second order polynomial is fitted through this parameter and an average failure threshold is determined. The results are better than the first two types, but not phenomenal. Therefore, Coble and Hines finally develop a general path model (a polynomial) with Bayesian updating to update the polynomial coefficient estimations. This model performs 8 times better than models of type 1 and 2, but still performs worse than the method by Peel [41].

Wang et al. [57] develop a similarity-based approach for determining the RUL of aircraft engines. Data of many different degradation trajectories resulting from the run-to-failure data are stored in a library. When the RUL of an engine must be determined, the degradation trajectory of that engine is compared to a degradation trajectory in the library and the most similar trajectory is used to predict the RUL of the engine. Before similarity-based matching is performed, the sensor measurements are fused into a single health indicator, a process called performance assessment. For this, Wang et al. use a simple linear regression model of the form $y = \alpha + \beta^T \cdot x + \epsilon$, where x is the feature vector containing all sensor measurements at a certain time, α and β are the model parameters and ϵ is the model noise. Once the health indicator is obtained, it will be matched to the degradation trajectories stored in the library, for which the health indicator is obtained in a similar fashion. The current engine which is inspected is matched to one of the library degradation trajectories minimising the Euclidean distance between the two trajectories.

Wang et al. apply the model to the PHM 2008 data set. Peel identified that the data set consists of 6 operational modes, which is also identified by Wang et al. Wang et al. manually select sensors of which they think are useful. The selected sensors show a clear trend of degradation. For each of the six operating regimes, a

linear regression model is build. The result of this is a one-dimensional time series which can then be used to build a model which is able to capture the degradation pattern from normal operating conditions to failure. Wang et al. use an exponential regression model for this. The test model is compared to the trained trajectories and the trajectories corresponding with the least distance to the test model are selected based on a threshold. Outliers from the selected trajectories are removed and the average RUL of the remaining trajectories is computed to obtain the final RUL estimation. An overview of this can be seen in Figure 3.6. The results of the approach are not as good as the previously discussed methods, as Peel [41] and Coble and Hines [6] both achieve better results (5 and 2 times better, respectively).

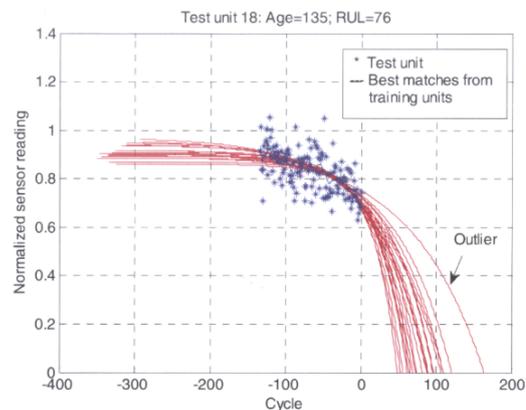


Figure 3.6: Similarity-based matching of degradation trajectories in a library to test data [57]

Giantomassi et al. [12] use a Hidden Markov Model (HMM) for the health estimation and prognostics of jet engines. HMMs are composed by a set of states that map observations in a PDF. Two processes are modelled in a HMM, a directly observable process and a process which is not directly observable but can be deduced from the observable process. Giantomassi et al. use three different algorithms to solve the HMM; the forward backward algorithm, the Viterbi algorithm and the Baum-Welch algorithm. The state transition probability matrix is updated by subtracting the HMM matrix with the new data matrix. The result is used to compute the new transition probability. The algorithm to generate the HMM as a large computational complexity and therefore, Giantomassi et al. make use of an artificial neural network (ANN) to reduce computational complexity. The ANN model is trained to extract a scalar that describes the behaviour of all sensors combined for that flight. Results show that the determination of the RUL estimation differs significantly from the real RUL during the first flights of a jet engine, until the moment that a fault is detected, often around half of the total lifetime, after which the estimation improves significantly and the difference between estimation and actual RUL is minimal. However, no scores have been provided so the method cannot be compared to aforementioned methods.

In subsection 3.1.3 the research by Siegel et al. [51] was reviewed. Siegel et al. used a Particle Filter approach to determine the RUL of jet engines. The model that they used to represent the degradation was briefly discussed here as well and will be elaborated upon further now. Siegel et al. make use of the fact that prior information is given about the fault location, which is in the high pressure combustor for the data set containing 1 fault mode. Using this information, Siegel et al. set up equations relating performance parameters of the high pressure combustor (pressure ratio, speed, flow and efficiency). The sensor measurements are used for these equations for which the compressor efficiency will eventually be the leading parameter.

Siegel et al. assume that the engine parameters degrade in an exponential manner, as this has been done in previous studies as well. As such, the compressor efficiency and non-dimensional compressor flow is assumed to degrade exponentially. Also, Siegel et al. assume that a decrease of 2% efficiency can be set as the failure threshold. The exponential degradation model of the compressor efficiency contains three unknown parameters which are updated using Particle Filtering, which was discussed in subsection 3.1.3. Results show that predictions for a RUL shorter than 40 cycles is quite accurate, but long-term predictions are rather poor. This is due to the fact that compressor degradation only becomes significant towards the end of the engine lifetime and the threshold of 2% is not optimal for all engines. Tang et al. [20] model the failure threshold

as a random variable from which the distribution is obtained from historical data, which showed favourable results. This might improve the method developed by Siegel et al. Only one operating condition is taken into account for this research and therefore, the performance cannot be compared to previously discussed papers.

Another method is proposed by Le Son et al. [24]. Previously discussed papers obtained degradation information using assumptions to model the degradation using compressor efficiency [51], averaging sensor measurements [6], linear and exponential regression [57] or using AI methods [12, 41]. Le Son et al. use a different approach by using a data analysis technique called principal component analysis (PCA). This method was also used in combination with polynomial chaos expansions, as was seen in subsection 3.1.5.

Le Son et al. set the goal to obtain a set of paths consisting of one health indicator. Next to this, the goal is to find a trend of this indicator for accurate RUL predictions. They propose to construct a "failure space" which contains a "failure place". The failure space is constructed using principal components of the data obtained through PCA. PCA is a technique to reduce the dimensionality of a data set by constructing non-physical principal components where the loss of information of the original data set is minimised. It does so, by constructing uncorrelated variables that maximise the variance[21]. Le Son et al. first select useful sensors to reduce the data set before applying PCA. Also, the data set is split into the 6 operational modes and PCA is applied for each mode. From the PCA it follows that 99% of the variance of the data set is stored in the first 2 principal components, which will be used for the construction of the degradation indicator. For each operational mode, the barycenter is located in the failure space, which is the centre of the failure space of an operational mode. The degradation of a unit at a certain time is then defined as the Euclidian distance of the point in the failure space to the barycenter of the failure space. An overview of the failure space and the corresponding barycenter of one operational mode can be seen in Figure 3.7.

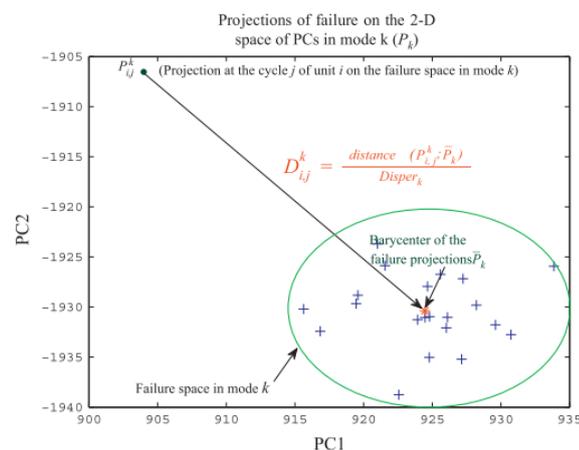


Figure 3.7: Failure space consisting of principal components with an indication of the barycenter of a certain operational mode [24]

Two methods for prognostics are used and compared to obtain the RUL. The first method models the resulting degradation indicator as a Wiener process. The Wiener process parameters are obtained by maximum likelihood estimation based on the training data set. The RUL is then determined by using Monte Carlo simulation. Secondly, a similarity based approach is developed, based on the method by Wang et al. [57]. Here, the degradation indicator is fitted to an exponential function for which the parameters are estimated using the training data set. A library containing a set of degradation trajectories is obtained and for the test data set, the trajectories that fit best are obtained using the similarity-based matching approach introduced by Wang et al. The Wiener process model obtains the best result. Compared to the other methods outline in this section, it compares slightly better than the method from Wang et al., but worse than the methods from Peel and Coble and Hines.

Fang et al. [9] also apply PCA in order to fuse sensor signals to one single degradation indicator. Next to that, they develop an algorithm, also using PCA, to identify which sensors are most informative for RUL prediction. The previously discussed papers either use all sensors or select a number of sensors manually.

Fang et al. start with the sensor selection algorithm which combines functional principal component analysis (FPCA) with a penalised (log)-location-scale functional regression model. Performing PCA to the data set results in FPC scores, which are regressed to the time to failure of the samples of the training data set. The regression model is then applied to determine the most informative sensors. The resulting dominant sensors are then selected for the generation of a health indicator. For this, multivariate functional PCA (MFPCA) is used. The advantage of using MFPCA is that it gives MFPC scores to fused signal features. Again, a (log)-location-scale functional regression model is developed to relate the system time to failure to the obtained fused degradation signal of the sensors. The parameters of the regression model are estimated using maximum likelihood estimation, which uses the MFPC scores. When the RUL is predicted for a new sample, the MFPC scores are updated when new data becomes available. Fang et al. do this by applying adaptive functional regression. Key for this approach is that training systems with a smaller lifetime than the current observation time are not taken into account for recalculating the MFPC scores. The remaining training sets are truncated at the current observation time so the time domain is equal for all signals. MFPC scores are recalculated using FPCA and served as input to the regression model which is then used to give a new estimation of the RUL of the system. The model is applied to the 2008 PHM challenge data set and results show that selecting useful sensors can improve estimation quality significantly. The results are not generalised with the competition score function, so no comparison regarding performance can be made with aforementioned papers.

Wen et al. [58] develop a composite health indicator using a nonlinear data fusion method based on genetic programming. Genetic programming is an optimisation algorithm based on the genetic algorithm. This will be used to fuse the sensor data to a degradation indicator. Genetic programming can be used to automatically select useful sensors. Genetic programming is based on phenomena occurring in the field of biology, such as inheritance and natural selection. Wen et al. use a data set containing 100 degradation trajectories, therefore the population is set at 100. The population propagates several generations until an optimal degradation function containing the most useful sensors is obtained. Each individual is modelled as a tree structure in genetic programming. The tree is split into several terminals and nodes, which include mathematical operators and sensor measurements. Wen et al. use a fitness value for each individual using a function. Individuals with a high score have a higher chance of reproducing. After a few generations, an optimal function is then established, which is reformulated as the health indicator. The next step is to develop a method to model the health indicator and to predict the RUL. An exponential formed degradation model is used for this. The degradation model parameters are estimated using weighted least squares for the training data set. Results of this method show that results regarding the RMSE of RUL prediction is superior compared to other methods. Wen et al. demonstrate that their method is suitable for RUL prediction of complex systems subject to multiple sensor measurements.

Li et al. [28] develop a method to automatically select useful sensors as well. They develop another multi-sensor data fusion model to predict the RUL of multi-sensor systems. The degradation process is modelled as a Wiener process. A multivariate measurement function is used to explicate the sensor measurements. The system state is then estimated by fusing the sensor signals with the use of particle filtering. The paper has already been reviewed in [subsection 3.1.3](#) with an emphasis on the particle filtering approach, but the degradation modelling aspect will be reviewed more thoroughly here.

As was explained in [subsection 3.1.3](#), the degradation process is modelled as a Wiener process with a drift parameter. A multivariate measurement function is established for all sensor measurements at a certain time. This function consists of an expression relating the system state and sensor signals with the corresponding coefficients. The expression should be monotonic, have deterministic model parameters and should have a limit from 0 to 1. An example is an exponential function ($\exp(cx)$) or a polynomial (x^c). The parameters of the state transition function and the multivariate measurement function are obtained using maximum likelihood estimation using training data. It was explained in [subsection 3.1.3](#) that PF is used for RUL prediction. The relation between the state transition function and the measurement function is recorded in the particle weights. Particle weights are updated each time step based on the observations, scale and location parameters, covariance matrix of measurement noise and the function relating measurements and system states. The method does not result in an explicit degradation indicator, but rather a direct approximation of the RUL. Furthermore, Li et al. develop a sensor selection algorithm. First, sensors are individually used for RUL esti-

mations in order to analyse the most dominant sensors. Based on this, a random group of best performing sensors is selected. This is done several times to obtain the best sensor selection. Results show that informative sensor selection leads to good results. The PHM scoring function is not applied, so the method cannot be compared to other methods, however, Li et al. compare their method to the earlier discussed method by Fang et al. [9]. Li et al. have a lower absolute relative error, indicating that the proposed prognostics and sensor selection method is better than the method from Fang et al.

Li et al. [29] have performed a comparative study of several prognostic approaches. Approaches discussed earlier have been replicated and the approaches are all applied to the same data set containing 6 operational modes and 1 fault mode. Scores are defined according to Equation 3.10. The results can be seen in Table 3.5. Approach 1 has been developed by Li et al. themselves, whereas approach 2 is based on the paper by Wang et al. [57] and approaches 3 and 4 are based on the paper by Le Son et al. [24]. It can be seen that approach 3 performs best, which involved PCA for the generation of a health indicator and further modelling using a Wiener process.

Table 3.5: Evaluation of different stochastic and statistical approaches on the 2008 PHM data set [29]

ID	Approach	PHM08 Evaluation	RSE	MSE
1	Weibull Distribution	7249.4	261.8072	1008.0
2	Similarity-based with Regression	231.0338	155.1644	354.0588
3	Wiener Process with PCA	190.2898	133.9813	263.9853
4	Similarity-based with PCA	305.0142	179.8027	475.4265

3.3. Maintenance Optimisation Methods under Uncertainty

This section reviews research performed in the field of maintenance policy optimisation. In section 3.1 and section 3.2 prognostics approaches and degradation modelling approaches have been discussed, which ultimately led to the prediction of the RUL of a component. The prediction with a quantified uncertainty can then be used to optimise a maintenance policy. A division is made between two types of maintenance optimisation; stochastic maintenance methods provided in subsection 3.3.1 and Markov decision process (MDP) optimisation methods in subsection 3.3.2. It has been decided to not include linear programming optimisation methods, because these are often used to optimise maintenance opportunities in flight schedules. The main focus will be to use prognostics results to decide when to schedule maintenance and not how this will fit into a flight schedule.

A selection of reviewed papers which are relevant for the research to be conducted is provided in Table 3.6. Stochastic optimisation and MDP optimisation methods are both provided here, but will be separately reviewed in the following sections. In Table 3.6, "Instant. duration" refers to whether the method assumes that maintenance is performed instantaneous or not. Also, "perfect/imperfect" refers to whether it is assumed whether the repaired component will be in an as-good-as-new state or not.

3.3.1. Stochastic Optimisation Methods

Do Van et al. [7] use the result of RUL prognostics to develop a stochastic maintenance policy for a single degrading system. The system is not continuously monitored, but periodic inspections are performed. The policy considers decision making related to maintenance actions and opportunities as well as logistic support and time between inspections.

The deterioration of the component is modelled as a Gamma process because this behaviour is observed in many physical deterioration processes. The parameters of the Gamma process are estimated using maximum likelihood estimations based on historical data. It is assumed that the degradation can be observed during periodic inspections which are assumed to be instantaneous, perfect and non-destructive. It is also assumed that preventive and corrective repair restore the system to the as good as new state. Different costs of maintenance are defined, an inspection cost and a cost for preventive maintenance which consists of system specific costs, setup-costs and planned shutdown costs. During an inspection, the next maintenance oppor-

Table 3.6: Literature review on maintenance optimisation methods

Authors	Year	Method	RUL Method	Optimisation Type	Optimisation Objective	Instant. duration?	Perfect/imperfect	Time horizon	Application Domain
Rajabi et al. [45]	2006	MDP solved using DP	N/A	MDP	Minimize cost and lost energy	No	N/A	Short	Power generator
Do Van et al. [7]	2012	CBM planning based on rules for RUL and time between inspections	Gamma process	SO	Minimize cost	Yes	Perfect	Long	Random system
Kim et al. [23]	2015	MDP solved using DP	Weibull	MDP	Minimize cost	No	N/A	Short	Water Main System
Cai et al. [4]	2016	CBM planning based on rules for RUL, PoF and availability	Linear growth model	SO	Maximise safety, minimise waste, maximise availability	No	N/A	Short	Aircraft engine
Shi and Zeng [49]	2016	CBM planning based on RUL and grouping opportunities of other components	Weibull	SO	Minimize cost	Yes	Perfect	Long	Random system
Verbert et al. [54]	2017	MDP solved using DP and RL	Assumed PDF for RUL	MDP	Minimize cost	Yes	Perfect	Short	Railway segments
Li et al. [31]	2018	MDP solved using RL (Gauss-Seidel algorithm)	Weibull	MDP	Minimize cost	Yes	Both	Long	Aircraft Engine
Nguyen et al. [35]	2019	CBM planning and ordering based on RUL	LSTM network	SO	Minimize cost	Yes	Perfect	Short	Aircraft Engine
Jin et al. [19]	2020	Semi-MDP solved using a simulation technique	Exponential/Weibull	MDP	Maximize revenue	No	Both	Long	Random system
Zhang and Zhang [62]	2020	CBM planning based on RUL, PoF and time between inspections	SAE-LSTM network	SO	Minimize cost	Yes	Imperfect	Short	Aircraft Engine
Hu et al. [17]	2021	MDP solved using RL (ELM and Q-learning)	N/A	MDP	Maximize mission reward	No	Perfect	Long	Aircraft Component

tunity within a certain amount of time is modelled through a truncated normal distribution. Two strategies are considered, a classical CBM policy and a RUL based policy. The classical policy involves two degradation thresholds, one of failure and one of system functioning, but badly. Failure involves corrective maintenance and unavailability costs. A badly functioning system results in preventive replacement. This threshold is optimised through Monte Carlo simulation. This policy serves for comparison with the more sophisticated RUL based policy. This policy plans maintenance based on the RUL and the next opportunity to perform maintenance. If the RUL is slightly larger than the time between inspections, the model looks for an opportunity to perform maintenance and setup the maintenance. The costs are minimised by optimising the time between inspections. The different costs involved rely on assumptions, but corrective repair is significantly more costly than preventive repair. A simulation study shows that RUL based policy shows significantly better results than the classical CBM policy. The paper shows how a simple decision process based on the RUL of a component leads to an optimal inspection time which minimises costs regarding maintenance performance and maintenance preparation.

Cai et al. [4] develop an optimal policy for aircraft engine shop visits based on the RUL of the engine. They make use of the approach to perform maintenance just-in-time (JIT) to minimise the waste of an aircraft engine. Also, Cai et al. include the effect of shop visit decision making on the service level and risk of failure of the engine. Furthermore, they develop a state space model for the prognostics part, which is based on the engine exhaust temperature. The method is demonstrated on a CFM-56 aircraft engine.

The prognostic model is developed by creating a state space model (SSM) consisting of a measurement equation and a degradation equation. The degradation equation is adopted as a Gaussian linear growth model

containing process noise. The model parameters are assumed to be known at the beginning. The parameters are updated using Bayesian state estimation. An analytical form of the PDF of the RUL k time steps ahead is formulated. The prognostic model has similarities to the method described by Siegel et al. [51] in subsection 3.2.2, as here also a SSM was developed using a single engine parameter. Next, the optimisation criteria are defined. First, the engine must not have a probability of failure larger than a threshold. Secondly, only one engine can be repaired at a time. Therefore, the probability of having two engines to be repaired at the same time should be minimised. A visualisation of the criteria can be seen in Figure 3.8.

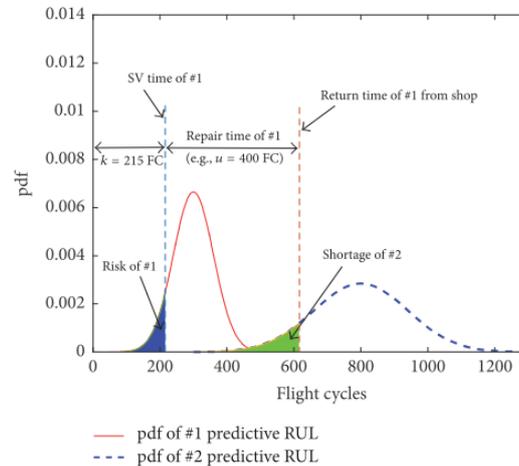


Figure 3.8: PoF and risk of unavailability second engine if first engine is being repaired [4]

The model is applied to a CFM-56 aircraft engine. Also, the repair time of an engine is modelled by a Gaussian distribution. The shop visit decision policy is formulated as follows. If the optimal repair time is more than 150 cycles, the next decision moment will be after 100 cycles. If the optimal repair time is between 100 and 150 cycles, the next decision moment will be after 50 cycles. Between 50 and 100 cycles, the engine must visit the shop in k cycles and below a RUL of 50 cycles the engine must be repaired as soon as possible. Furthermore, the probability of failure requirement must be taken into account and the probability that two engines must be maintained at the same time must be below a threshold. Depending on which criteria is violated first, the engine is maintained after k flight cycles. Although the RUL prediction is simple, it is concluded that it is better than regular linear regression. The results of the maintenance policy show that the policy can easily be implemented by MRO companies. The method by Cai et al. maximises the full life of engines and maximises the availability of engines by avoiding two repaired engines at the same time. In contrary to the method by Do Van et al. [7], no cost optimisation is taken into account.

Shi and Zeng [49] apply a dynamic opportunistic maintenance policy for maintenance decision making. The method is applied to multi-component systems of which the effect of degradation of one component can also influence the degradation of another component. A model is created which optimises the trade-off between maximising the useful life of a component and minimising the setup costs for maintenance, which reduces if several components are grouped together for maintenance at the same time. A dynamic opportunistic maintenance zone is established and the optimisation objective is to minimise long-term maintenance cost.

Shi and Zeng use historic condition monitoring data to predict the RUL of components real-time. Costs are defined for preventive maintenance and the corresponding setup costs. Corrective replacement is given a very high cost compared to preventive maintenance cost. Corrective replacement results in whole system downtime, but a maintenance opportunity for other components to be replaced. Penalty costs are defined for decreasing the RUL of some components if maintenance is performed early due to grouping, which reduces setup costs so this must be optimised. Real-time RUL prediction is performed by a stochastic filter and applying Bayesian theory. Liu et al. [32] also apply stochastic filtering in combination with the optimisation of a maintenance policy. However, they use it to model the influence of sensor degradation as well. When all cost functions are defined, a function to define the optimal OM zone is defined, meaning that components being in this horizon are grouped together for maintenance. An opportunity is defined as the first component that

needs replacement. The method is optimised long-term by dynamically grouping components over and over again until an optimal grouping structure is obtained. The method is applied to a random system with assumed costs and the optimisation algorithm is solved using particle swarm optimisation. Relations between preventive and corrective costs are similar to costs defined by Do Van et al. [7]. Results show that roughly 8% of costs are saved by grouping components for maintenance compared to single-component maintenance, which was presented as well by Do Van et al. [7]. Shi et al. [50] have also developed a maintenance policy which groups different components together. They use a dynamic-priority-based heuristic algorithm to group components, instead of an opportunistic policy. Shi et al. demonstrate as well that grouping might lead to significant cost reduction despite of not using the entire lifetime of a component.

Another stochastic optimisation approach is proposed by Nguyen et al. [35]. Nguyen et al. also use the 2008 PHM challenge data set to develop a prognostic model and use the resulting RUL to plan maintenance decisions. The prognostic model is developed using a long short-term memory network, which is a widely used in deep learning applications. As degradation models using the PHM challenge data set have been already discussed in subsection 3.2.2, the main focus here will be on the maintenance policy optimisation.

Based on the resulting RUL from the prognostic model, several decisions can be made: repair or do nothing and order a spare part or do not order. The decision to order or not to order depends on the corresponding cost. For example, ordering early means that high costs are involved for storing the part for a long amount of time, but ordering too late results in system downtime as the replacement cannot start immediately at the desired replacement time. Replacing or doing nothing also depends on the decision resulting in the lowest cost. Early replacements are for example very expensive, as this results in waste of useful life. Furthermore, the decisions are made at regular inspection times (every 10 cycles) and the optimisation objective is to minimise costs. Three states are defined for the decisions. A RUL larger than 20 cycles means doing nothing and ordering if costs are lower than not ordering (order in time). The spare part must be in stock in the second state between a RUL of 10 and 20. A (perfect) repair is performed for a RUL smaller than 10. The cost rate is defined as the relation between the cost of corrective or preventive replacement and the inspection period and time. The cost rate serves to compare the developed method with periodic maintenance policy and a perfect maintenance policy. Results show that the proposed methodology performs significantly better than periodic maintenance and only slightly worse than the perfect maintenance policy, indicating that the proposed method performs very well. Again, costs regarding maintenance are assumed as no available data regarding this can be found in literature.

A final stochastic maintenance optimisation method is discussed now. Zhang and Zhang [62] use the PHM challenge data set as well to develop a prognostic model based on long short-term memory. The difference with the prognostic method by Nguyen et al. [35] is that Zhang and Zhang also include a stacked autoencoder. Also, whereas Nguyen et al. assume perfect inspections for their maintenance policy, Zhang and Zhang develop a framework using imperfect inspections. Also, the maintenance policy is different, which will be reviewed now.

The state of an engine can be subdivided into three states: good, defective and failure. Good means that the failure threshold has not been reached and the probability of failure is smaller than a threshold. Defective means that the engine is still working, but one of the two thresholds has been crossed. The failed state is clear by its name. Again, a cost rate is defined which will be compared to a periodic and a perfect maintenance policy for performance evaluation. The cost rate consists of a term for downtime cost, inspection cost, corrective cost and preventive cost. No ordering of spare parts is included, as it is assumed that the component can be repaired imperfectly and not replaced. It is assumed that every 25 cycles an inspection is performed and it is assessed whether to perform maintenance or not. If the state is good, nothing is done and evaluation occurs again at the next inspection. When the component is defective, preventive repair is performed. When the component has failed, corrective repair is performed. The method is compared to the aforementioned maintenance policies for different times between inspections, which is optimised. The key difference between this method and the method by Nguyen et al. is that Zhang and Zhang use an imperfect maintenance strategy and include the probability of failure as well. Results are similar, the method is significantly better than periodic maintenance and almost as good as the cost of perfectly predicted maintenance.

Many other papers provide stochastic optimisation for maintenance policies. Other interesting papers vary

the degradation threshold over time in order to locate and identify possible early failures [60]. Another method that is often used in literature to solve stochastic maintenance optimisation problems using genetic programming [10, 55]. However, that will be beyond the scope of this review.

3.3.2. Markov Decision Process Optimisation Methods

The second optimisation method which involves Markov decision process (MDP) optimisation will be reviewed now. Rajabi-Ghahnavie and Fotuhi-Firuzabad [45] provide a simple method which uses a MDP to determine when a power generating unit must be maintained. There are three available power generating units, of which one can be maintained at a time as a limitation due to crew and facility capacity constraints. The goal is as in many optimisation problems to minimise costs by solving the cost equation of the MDP, which is often formulated as $V(s)$ in Equation 3.11. The optimal policy $\pi(s)$ consists of a set of actions which minimises this cost equation. Here, s is the state, s' is the next state, P is the associated transition probability, R is the reward function and γ is the discount rate. It is required to solve Equation 3.11 to obtain the minimum cost. Often, dynamic programming is used for this in literature. Key for this method is to divide the problem in smaller and simpler sub-problems and solve them recursively. A visualisation of a MDP can be seen in Figure 3.9, showing how states, costs, actions and state transition are defined.

$$\begin{aligned} V(s) &:= \sum_{s'} P_{\pi(s)}(s, s') (R_{\pi(s)}(s, s') + \gamma V(s')) \\ \pi(s) &:= \operatorname{argmin}_a \{ \sum_{s'} P(s' | s, a) (R(s' | s, a) + \gamma V(s')) \} \end{aligned} \quad (3.11)$$

The actions that can be taken here are to either do nothing or to perform maintenance. The objective in this case is to minimise costs by minimising the unserved energy and the unserved reserve costs. Results show that the unit generating the most power should be maintained first, because the expected energy consumption is expected to increase later in the year. The other 2 units are maintained immediately after completion. Furthermore, the impact of maintenance duration and energy consumption variation is considered. The method shows simply how MDP processes can be used to solve the pricing problem and determine an optimal policy in which order and when to perform maintenance to minimise costs.

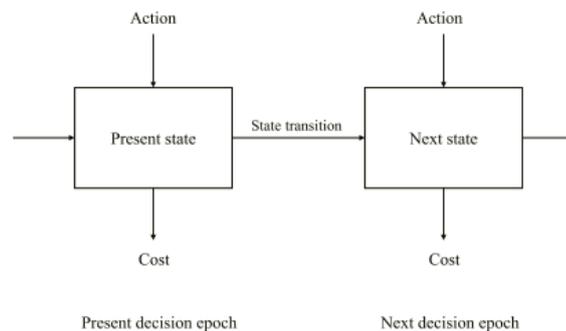


Figure 3.9: Sequential decision problem [44]

Kim et al. [23] also provide a relatively simple demonstration of how to use a MDP to determine an optimal maintenance policy. The method in this case is applied to a water main system. Kim et al. start by outlining the MDP framework and assuming that the water main system degrades following a Weibull distribution. The MDP framework is solved using dynamic programming and the Bellman equations. The method is applied to the water main system of which the system health is divided in 5 states with a decision horizon of 100 years. Costs are assumed as well as no data is available for this. Results are compared to a periodic maintenance policy and a myopic policy which maximises short-term rewards. The problem is subdivided into 5 sub-problems which are solved separately. Each sub-problem corresponds with one of the 5 states as the initial state. Solving the framework results in an optimal time between inspections. Again a simple framework which combines MDP and dynamic programming is provided which illustrates the simple implementation of Markov decision processes.

Verbert et al. [54] develop a more thorough and advanced MDP framework for multi-component systems using condition based maintenance planning. A method is developed which favours timely planning instead

of last minute maintenance planning. The framework is split into two parts. The first part involves determining the best maintenance strategy for an individual component so maintenance decisions can be tuned to specific situations. A MDP is formulated which can be either solved by dynamic programming or reinforcement learning. The second part involves optimising the maintenance policy on a system level by combining the individual maintenance policies. Here, economic dependence and availability of the system is taken into account by combining or spreading individual maintenance activities. Again, the goal is to minimise cost, while the system down time is also minimised and failure should be prevented.

RUL determination using prognostics is beyond the scope of the paper, so it is assumed that the PDF of the RUL is known at the time of an inspection. Inspections are performed at fixed times and perfect repair of components is assumed. Decision making on the component level is done by determining what type of maintenance to perform when. Also, a trade-off is made between early maintenance planning (which is favourable) with a less good prediction quality and late maintenance planning for which prediction quality is better. Three types of cost are defined for individual components; direct maintenance cost, indirect maintenance cost and cost associated with risk due to an action being inadequate or too late. The sum of these should be minimised. The procedure is then at a time t to accept the proposed maintenance strategy or to postpone until the next evaluation time and decide again to plan or postpone (these are the two actions). For this, the MDP framework is formulated consisting of many states depending on cost of the current maintenance strategy, the component health, whether maintenance is planned or not and whether the component has failed or not. Transition probabilities are formulated as well and a reward function is established. A reward is given for planning maintenance. If no maintenance is planned and the system does not fail, no reward is given. If the system fails, a penalty is given. The system is solved for each component and then the optimisation on system level can be performed. Again, the total long-term costs are minimised. These costs consist of negative costs for economies of scale (so not an actual cost), reduction in downtime costs and loss of functionality costs. Verbert et al. optimise on the system level using a brute force technique. The method is applied on a railway network. Individual railway track sections are the components, whereas a triangular city network serves as the system. The action to postpone or to plan maintenance is given an analytical expression (including the RUL prognostics) which is solved each time it is evaluated whether to postpone or to plan maintenance. Two types of defects can occur, so two types of maintenance can be scheduled. The optimal maintenance type and time is determined for each railway track section. Then, the system level is optimised. It is not ideal that some railway sections have maintenance simultaneously, as then the connection between cities might not be available for a certain amount of time. As only 7 sections require maintenance, the solution can be obtained through brute forcing. An optimal maintenance policy for the system is obtained as such. The method shows that MDPs can be used very well in combination with the PDF of the RUL obtained through prognostics. Also, it has been shown that several fault types, maintenance types and ideal maintenance times can be obtained on the component level as well as on the system level.

Li et al. [31] develop a maintenance policy optimisation method formulated as a MDP. The MDP is solved using reinforcement learning, one of the three paradigms in machine learning next to supervised and unsupervised machine learning. The model is applied to the maintenance policy of aircraft engines. The method is compared to traditional maintenance policy optimisation methods, which are included in the model as well and solved synchronously to the reinforcement learning model. The traditional models consist of periodic maintenance, corrective maintenance and condition-based maintenance. In reinforcement learning, an agent determines optimal maintenance policy. The agent is able to respond to the dynamically changing states of the aircraft engine. Depending on the engine state, the aircraft takes an action which results in a certain cost. The objective of the agent is to minimise the cost. Li et al. set up a multidimensional state space consisting of life limit part (LLP) state, engine performance state (depending on the exhaust gas temperature) and random failure state. For each state, different actions can be chosen on each decision epoch (replace, recover, correct or do nothing). Transition probability matrices are defined as well. The LLP probabilities are always 1 depending on the action. Performance state transition probabilities are derived from a Weibull distribution. The random failure probability is related to the performance state. Cost functions are provided based on hypothetical data. The resulting MDP framework is solved using the Gauss-Seidel value iteration algorithm. Li et al. [31] apply the framework to aircraft engines, consisting of 5 states for both the performance and LLP states. The optimal policy minimising the cost depending on the system states can be seen in Figure 3.10.

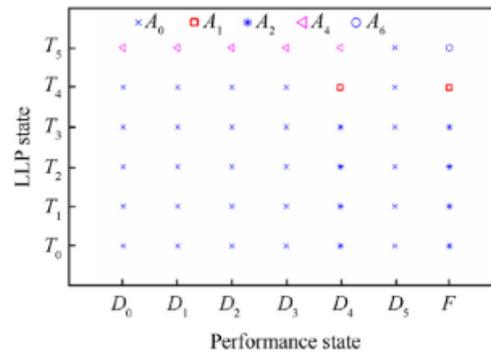


Figure 3.10: Maintenance policy map where A defines the action to be taken [31]

Other experiments were conducted which showed that a robust method is proposed which is able to address the hybrid maintenance strategy by combining LLP maintenance, CBM and CM. As little historical data was available, the reinforcement learning optimisation was performed which showed good performance. Although different components are considered, grouping these to simultaneously perform maintenance is not discussed in this paper, which was done by Verbert et al. [54] and Shi and Zeng [49].

Jin et al. [19] develop an efficient maintenance strategy using a semi-MDP which also includes the deterioration process of a multi-state system. The main difference between a regular MDP and a semi-MDP is that for a regular MDP the time spent in one state is one unit of time, whereas for a semi-MDP this amount of time can vary or even be a random variable [44]. In the framework of maintenance optimisation, this often involves the maintenance duration time. The previously discussed paper by Li et al. [31] for example assumes that maintenance is performed instantly, whereas Jin et al. [19] do not make this assumption.

The semi-MDP framework is set up in a similar fashion as outlined previously in this section and is applied to a small and a large scale problem. The goal of the framework is to maximise the revenue of the system over an infinite horizon. The semi-MDP is not solved using dynamic programming or reinforcement learning, as was seen previously in this subsection, but using a simulation technique because the goal is to apply the framework to problems with many states and possible actions. These type of problems are sensitive to the curse of dimensionality if the number of states and actions increases rapidly. The iterative simulation technique evaluates many policies which results in an optimal policy. Simulation studies show that the method is especially convenient for large-scale maintenance policy problems, for which the type of maintenance, time of maintenance and corresponding revenue are provided.

The final paper that will be reviewed is a paper by Hu et al. [17]. Like Li et al. [31], Hu et al. develop a MDP process and use reinforcement learning to solve for the optimal maintenance policy. The research is motivated by the fact that many maintenance optimisation methods from literature focus on short-term rewards and hardly cover mission rewards on the long-term. This research addresses both short-term rewards and long-term mission rewards. Innovative in this paper is that next to maintenance policy, aircraft missions are modelled as well. An agent is able to choose which mission to perform based on the mission reward and corresponding maintenance cost and time. Furthermore, decisions can be made regarding flying or not, ordering a spare part or not and repairing or not, similar to the options defined by Nguyen et al. [35]. Reinforcement learning is applied to obtain the policy which maximises the reward of the action-value function. The aircraft state depends on the RUL (which is assumed to be known prior to a mission), mission profile and the available spare components. The policy resulting in the optimum of action-value function is obtained using an extreme learning machine algorithm. The algorithm is trained using the Q-learning method, a commonly used reinforcement learning algorithm.

A simulation study is performed to demonstrate the proposed reinforcement learning solution. As hardly any data is available for mission reward and maintenance cost for aircraft, the parameters are simulated through uniform distributions of hypothetical distribution parameters. The result is a schedule for a single aircraft containing information on when to do which mission and when to perform maintenance. Many

papers compare their maintenance policy result to CM, PM and CBM policies as was seen in this section, which Hu et al. also do. The RL policy scores slightly better than the CBM policy due to the fact that storage costs are high for the CBM policy. If data is subject to noise, all policies perform poor except for the RL policy.

4

Research Framework

This chapter covers the research motivation. First, the research gap is identified in [section 4.1](#) following from the literature review from [chapter 3](#). The scientific research gap is used to define the research aim in [section 4.2](#) and based on this, the research questions are formulated in [section 4.3](#). Finally, the research scope will be identified in [section 4.4](#).

4.1. Research Gap

Following from the literature review covered in [chapter 3](#), scientific research gaps can be identified. The identified research gaps will then be used in order to define the research aim, research questions and research scope later in this chapter. The research gap regarding prognostics and degradation modelling will be covered first, after which the research gap regarding maintenance optimisation will be covered.

Two main strategies for RUL prognostics have been covered in the literature review, particle filtering and polynomial chaos expansions. It was seen that particle filtering is a widely used approach in the field of prognostics in combination with a component degradation model. It was also seen that particle filtering has been used for the PHM data challenge several times as well, however, different degradation models and sampling methods were used. Also, different approaches have been used to establish the degradation or health indicator from aircraft engines. Although particle filtering prognostics has been researched several times in combination with the PHM challenge data set, there are still variations of the problem which have not been covered yet by particle filtering. For example, it was seen that Siegel et al. [51] develop a degradation model based on the efficiency degradation calculated using only a few sensor measurements. An exponential degradation function is fitted using historical data. The parameters of the function are updated using particle filtering and particle filtering is used to estimate the RUL. It would be interesting to develop another degradation model not based on solely the engine efficiency which utilises more sensors. Principal component analysis (PCA) in combination with a barycenter for a fault mode to model degradation of an implicit multi-sensor data set was introduced by Le Son et al. [24]. It has not been researched how a degradation indicator obtained through PCA and using an exponential function to model the degradation in the future and using particle filtering to estimate the function parameters can lead to an estimation and PDF of the RUL. This could be an approach which can be taken if it is chosen to use a particle filtering approach. Furthermore, no research has been conducted which combines a statistical and stochastic prognostic approach in combination with a maintenance policy optimisation method, as most research focuses on either developing a RUL prognostics approach or using the result to develop a maintenance policy optimisation approach. Therefore, a completely novel approach would be to research how a model-based statistical and stochastic prognostic approach using particle filtering would result in an optimal maintenance policy which performs better than existing maintenance policies such as periodic maintenance policy or corrective maintenance policy.

Polynomial chaos expansion (PCE) was the second prognostics approach covered in [chapter 3](#). Here, it was seen that PCE has hardly been used in the field of prognostics. Some research has been performed that has demonstrated that PCE can be used to determine the RUL estimation and uncertainty quantification of degrading systems. However, most research that has been conducted using PCE techniques are used for other

applications such as uncertainty quantification in aircraft trajectories or to obtain surrogate models for computationally extensive physical models such as an urban water drainage model. Therefore, a research gap can be identified which is to use PCE for prognostics to the PHM challenge data set, as this approach has not been researched but PCE for RUL determination has been shown to be significantly faster than Monte Carlo methods with significant accuracy. Furthermore, several papers discussed in [subsection 3.1.5](#) discussed the combination of PCA and PCE. Nagel et al. [34] concluded that the synergy of PCE and PCA has not been researched extensively and recommend to research the combination of these two approaches as this leads to significant computational time reduction while preserving most of the information in the used data set. Therefore, it would be interesting to research the combination of PCE and PCA in a prognostics framework, as this has not been done yet. An approach would be to use PCA as introduced by Le Son et al. [24] to obtain the health indicator of a component using an implicit multi-sensor data set. As was explained above for particle filtering, the result could be modelled using an exponential function for which PCE can be used for RUL prognostics and uncertainty quantification. In the previous paragraph it was also outlined that no research has been conducted which combines a stochastic and statistical model-based prognostics approach with a maintenance policy optimisation method, which also applies in the case of using PCE for the prognostic method.

Different maintenance policy optimisation methods have been reviewed in [chapter 3](#), which were split in stochastic optimisation methods and Markov decision process optimisation methods. Two papers were reviewed which covered a stochastic maintenance policy optimisation method using the 2008 PHM challenge data set. For the prognostics part, both used a long short-term memory network, an artificial intelligence method. No stochastic and statistical model-based prognostics approach has been used to determine an optimal maintenance policy and therefore, it would be interesting to research this. Also, both approaches by Nguyen et al. [35] and Zhang and Zhang [62] assume that maintenance is being performed instantaneously and the optimisation horizon is short. Both use a stochastic optimisation approach to determine the optimal short-term maintenance policy. Because of this, a research gap can be identified as no Markov decision process has been used to optimise a maintenance policy for aircraft engines using an implicit multi-sensor data set. Also, the long-term optimisation for single aircraft or a fleet of aircraft has not been covered in literature for both optimisation methods. Next to this, it would be interesting to not assume that maintenance can be performed instantly. This would be especially useful for optimisation using a long-term time frame and if optimisation takes place for a single aircraft or fleet of aircraft.

The research gaps identified in this section will be the basis of the remainder of this chapter. The research gaps will be used to formulate the research aim and to construct corresponding research questions.

4.2. Research Aim

Following from the identified research gaps, the research aim can be defined. The research will make an effort to provide useful information in the field of prognostics and maintenance policy optimisation and help to further develop prognostic methods and provide new insights. The research aim can be subdivided in two parts:

1. Developing a stochastic and statistical model-based prognostic model which is able to estimate the remaining useful life of an aircraft engine based on implicit multi-sensor measurements obtained during the lifetime of that engine, as well as quantify the uncertainty of the remaining useful life estimation in the form of a probability density function.
2. Developing a maintenance policy optimisation method which is able to implement the result of the prognostic model (the probability density function of the remaining useful life of an aircraft engine) in order to obtain an optimal maintenance policy which reduces long-term maintenance costs compared to other maintenance policies.

The aim of the research is defined and now the research questions can be formulated, which will be done in the following section.

4.3. Research Questions

The research questions follow from the research gaps and the research aim defined in the previous sections. For the research to be conducted the following main research question is formulated:

Main Research Question

How can RUL prognostics applied to aircraft engines be used to optimise maintenance policies?

In order to answer the main research question, the following research sub-questions are defined.

1. Considering an implicit multi-sensor engine degradation data set of a certain aircraft engine type, how can the RUL and its corresponding PDF of an aircraft engine of this type be determined using a stochastic and statistical method?
 - (a) What type of model will be used to predict the RUL?
 - (b) How will the underlying degradation of an engine be modelled using available sensor measurements?
2. How can the PDF of the RUL of an aircraft engine which is in a certain degradation state be used to optimise the maintenance policy of an aircraft utilising this aircraft engine?
 - (a) What type of optimisation model will be used to optimise the maintenance policy?
 - (b) What will be the objective of the optimisation policy?
 - (c) What will be the cost of the maintenance policy?
3. How does the proposed prognostic method and maintenance policy compare to current maintenance strategies?
 - (a) How does the prognostic approach compare to other methods?
 - (b) How does the maintenance policy compare to periodic maintenance, corrective maintenance and perfect maintenance strategies in terms of costs, number of unscheduled replacements and waste life?

4.4. Research Scope

The research gaps, research aim and research questions have been formulated in the sections above. Finally, the research scope will be determined in order to set boundaries to the research framework and determine what will be done and what will be left beyond the scope of the research project.

First of all, the available 2008 PHM challenge data set consists of four different data sets. The first two data sets contain only one fault mode. One data set consists of 1 operational mode and the other data set consists of 6 operational modes. The other two data sets also have respectively 1 and 6 operational modes, but two fault modes are present here. For this research, the data set containing one fault mode and six operational modes will be used to develop the initial model. Then, if time allows, the model will be extended to two fault modes.

The output of the prognostic model using this data set will be the probability density function of the remaining useful life. It is assumed that when the RUL of an engine is determined at a certain time, historical data until that point is complete and no missing data needs to be accounted for. The prognostic model will not be evaluated through a sensitivity analysis, this will only be performed for the maintenance optimisation part.

The maintenance policy optimisation will use the output of the prognostic model. The emphasis of the research will be on developing the prognostic method and the maintenance policy optimisation model will serve as a demonstration to show how prognostics can be used to minimise engine waste and minimise maintenance costs. For engine maintenance, it is assumed that there is hangar availability for one aircraft to be maintained at a time. Furthermore, grouping of different aircraft components to perform maintenance simultaneously is not considered. In literature, the optimisation for maintenance often includes cost functions to estimate the cost of maintenance. However, the exact numbers for these functions are often unknown and assumptions are made for these numbers, as was outlined in [section 3.3](#). Therefore, for this research the costs of aircraft will also be assumed based on assumed cost functions in literature. The exact costs will then not be known, but they can be compared to existing maintenance strategies using which can be calculated using the same cost functions. It is also assumed that aircraft maintenance can take place if no other aircraft is undergoing maintenance regardless of the exact flying schedule of the aircraft. In other words, the exact flight schedule of a fleet of aircraft will not be taken into account. The result of the maintenance policy optimisation will be an optimal policy of when to do maintenance and when not to do maintenance.

5

Thesis Planning

5.1. Gantt Chart

The milestones for the thesis can be seen in [Table 5.1](#) and the Gantt chart for the thesis can be seen in [Figure 5.1](#).

Table 5.1: Thesis planning of milestones

Week	Date	Milestone
Week 0	06/05/2021	Kick-off meeting
Week 14	16/09/2021	Midterm
Week 26	10/12/2021	Hand-in thesis draft
Week 27	17/12/2021	Green light meeting
Week 29	07/01/2022	Hand-in thesis final version
Week 31	20/01/2022	Thesis defence
Holidays	Start	End
Summer Holiday	19/07/2021	20/08/2021
Christmas Holiday	27/12/2021	31/12/2021

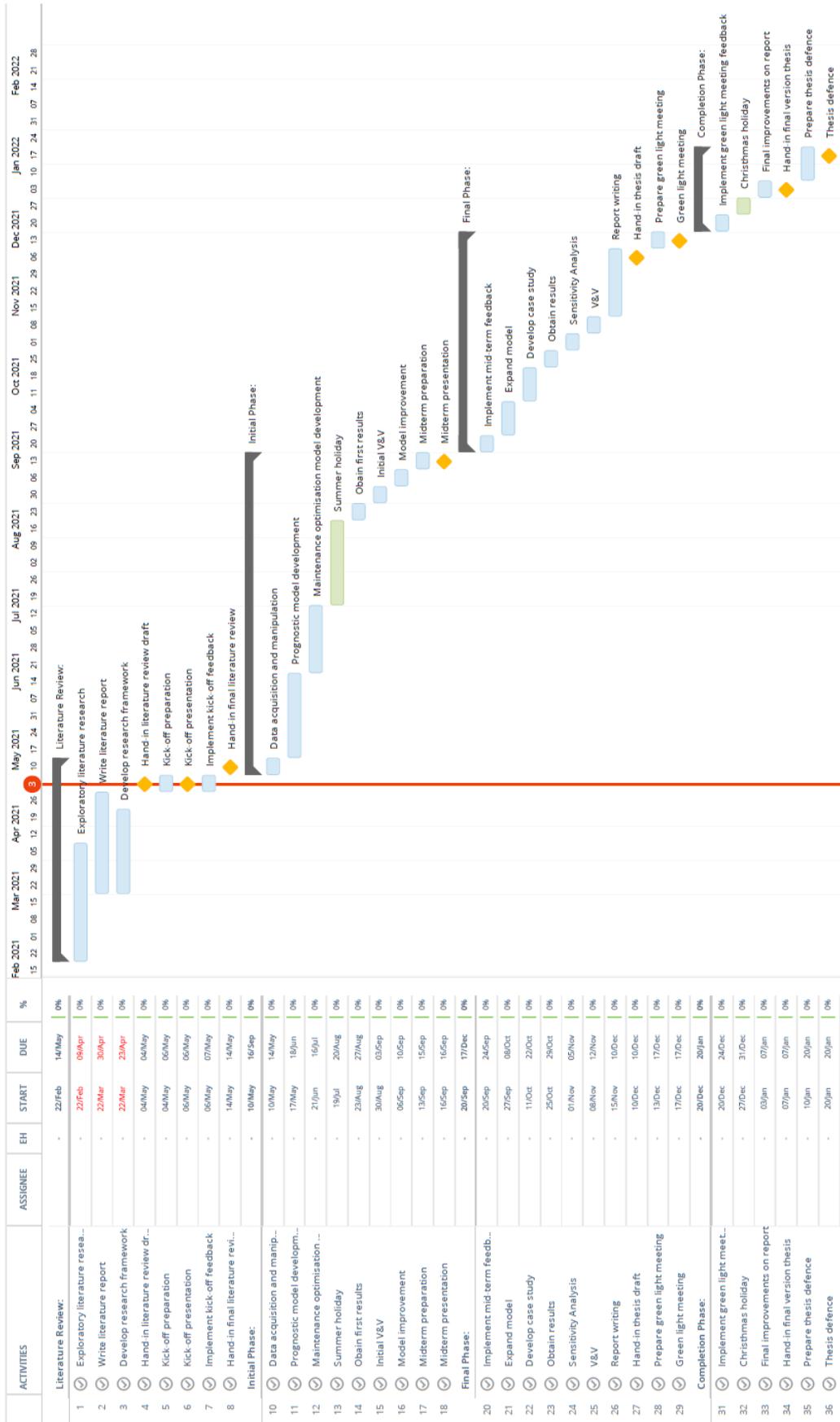


Figure 5.1: Project Gantt chart

6

Conclusion

Prognostics in aircraft maintenance is becoming more popular every year. Maintenance repair and overhaul companies are gradually shifting from conservative maintenance strategies to the implementation of prognostics in order to optimally use aircraft components and reduce aircraft on ground time due to unexpected faults. A literature review was conducted in order to set up a research framework. The literature review considered three major topics; prognostics in aircraft maintenance, degradation modelling of multi-sensor systems and maintenance optimisation methods.

The literature review resulted in the identification of research gaps. Polynomial chaos expansions have been applied in many fields, but are relatively undiscovered in the field of prognostics. A small number of papers apply and demonstrate the feasibility of applying PCE in the field of prognostics, but no extensive problems using multi-sensor data have been solved using PCE, especially not in the aviation maintenance industry. Therefore, it can be concluded that it would be a novel approach to develop an aircraft engine prognostic model using PCEs.

Furthermore, maintenance optimisation methods have been reviewed. The emphasis of the review was on stochastic optimisation and Markov decision process optimisation. Both methods have not been used in combination with a stochastic and statistical model-based prognostic model in order to obtain an optimal maintenance policy. Also, no long-term optimisation strategy for aircraft engine maintenance policy optimisation has been researched for a single aircraft or a fleet of aircraft. Therefore, both methods would be suitable to be researched for long-term aircraft maintenance.

Following from this, a model will be developed which is able to provide a probability density function of the aircraft engine remaining useful life at a certain moment in time by using a stochastic and statistical model-based prognostic model using PCEs. Next, this result is used to develop an aircraft engine maintenance policy optimisation method which provides an optimal maintenance policy which minimises costs compared to other, conservative, maintenance strategies.

III

Supporting work

Additional results: health indicator modelling & RUL prognostics

In this chapter, additional results are presented which are used to create the engine health indicator over time for the different data sets. Also, some additional RUL prognostic results are given for the three data sets discussed in [Part I](#).

1.1. PHM Challenge data set

1.1.1. Sensor measurements split per operational mode

In [Figure 1.1](#) to [Figure 1.6](#) all sensor measurements of all engines split per operational mode are provided for the PHM Challenge training data set.

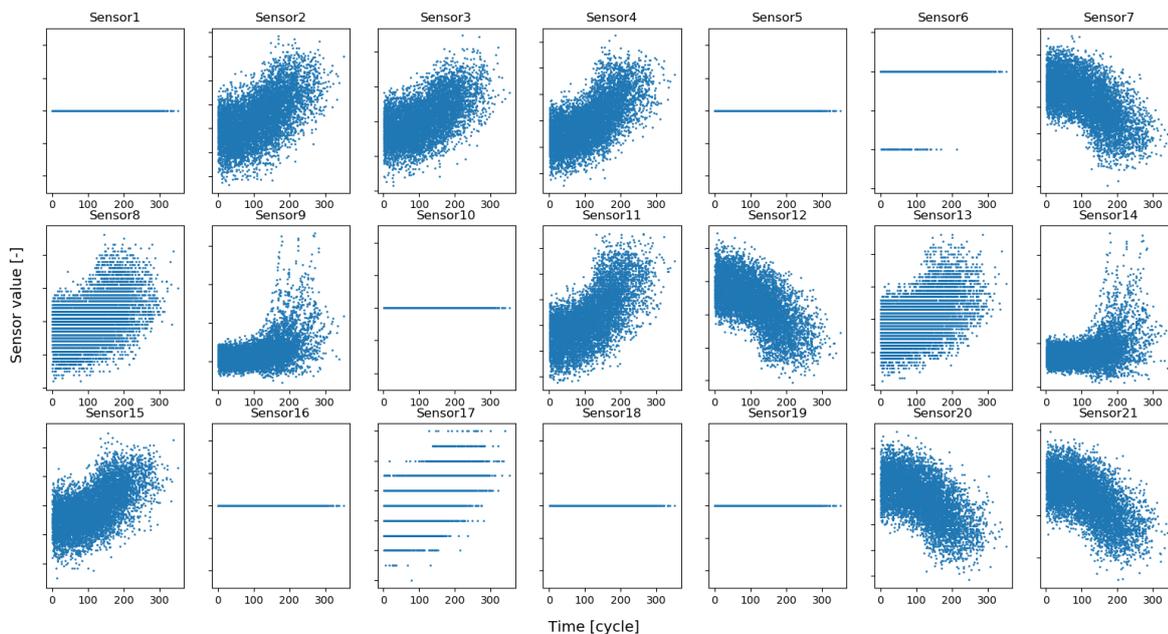


Figure 1.1: Sensor measurements of all sensors - PHM Challenge training data set operational mode 1.

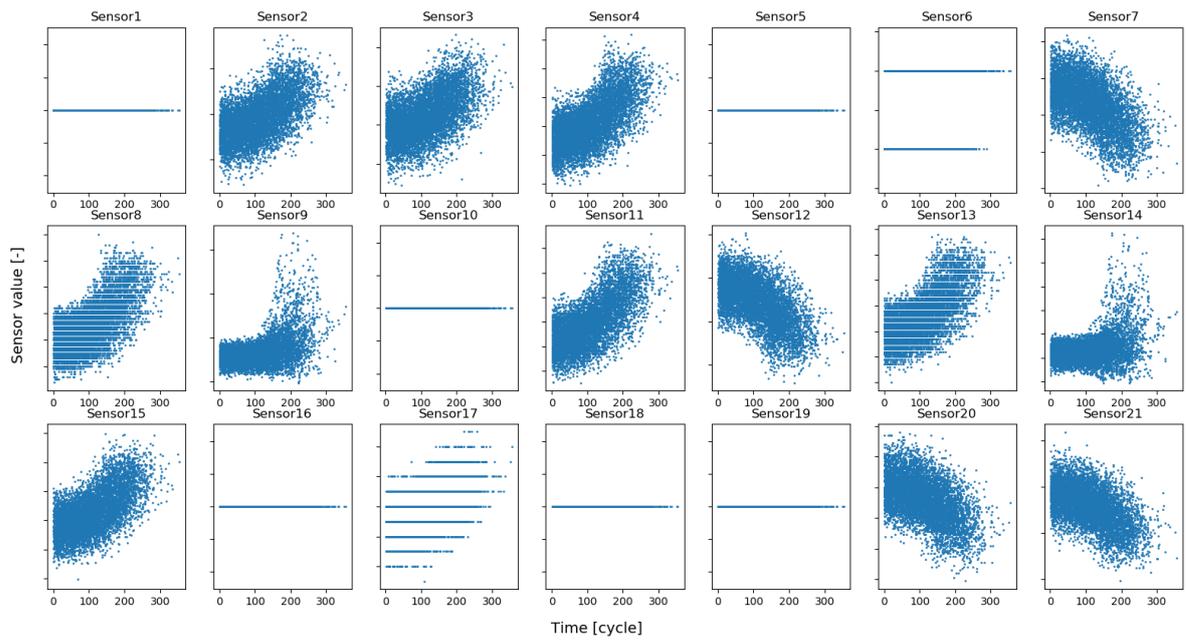


Figure 1.2: Sensor measurements of all sensors - PHM Challenge training data set operational mode 2.

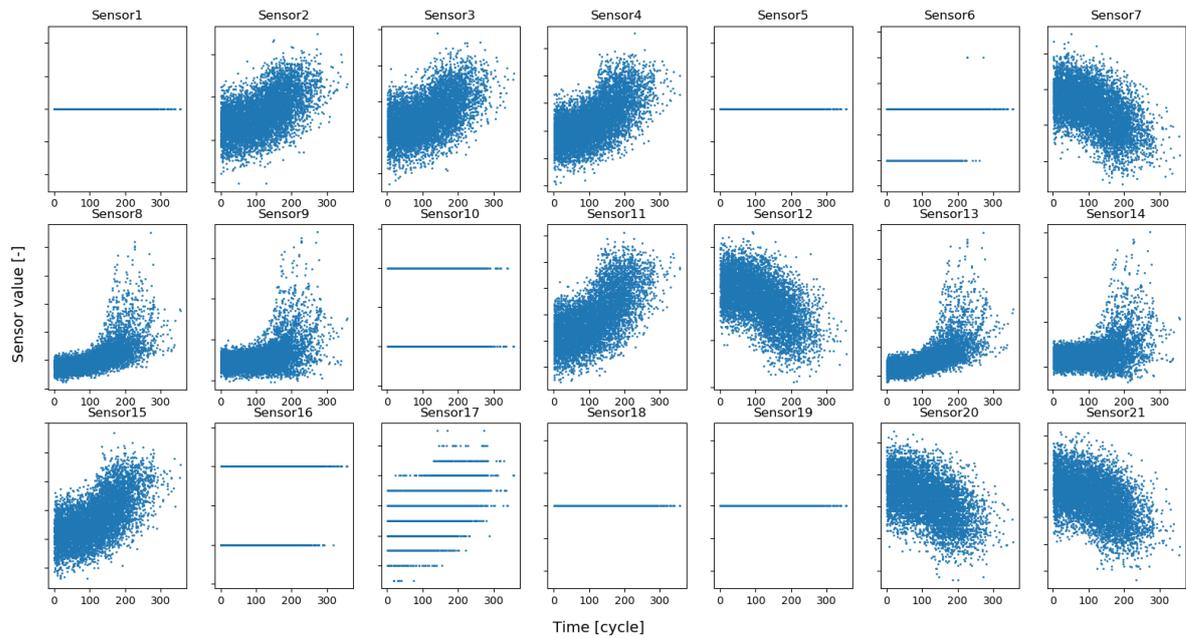


Figure 1.3: Sensor measurements of all sensors - PHM Challenge training data set operational mode 3.

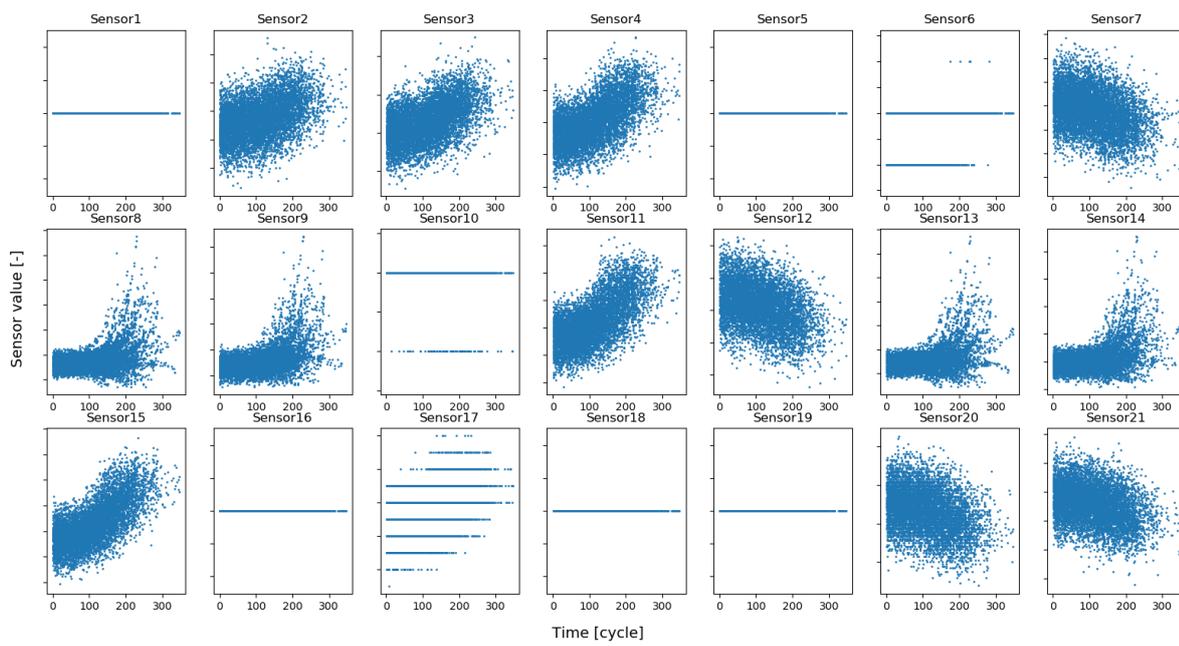


Figure 1.4: Sensor measurements of all sensors - PHM Challenge training data set operational mode 4.

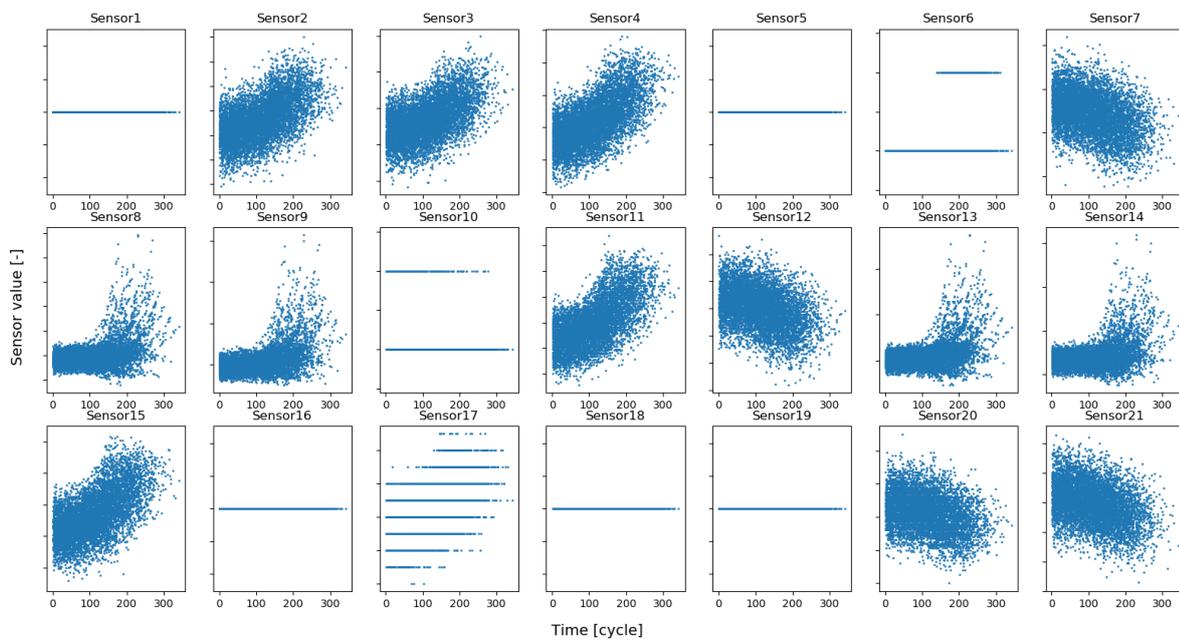


Figure 1.5: Sensor measurements of all sensors - PHM Challenge training data set operational mode 5.

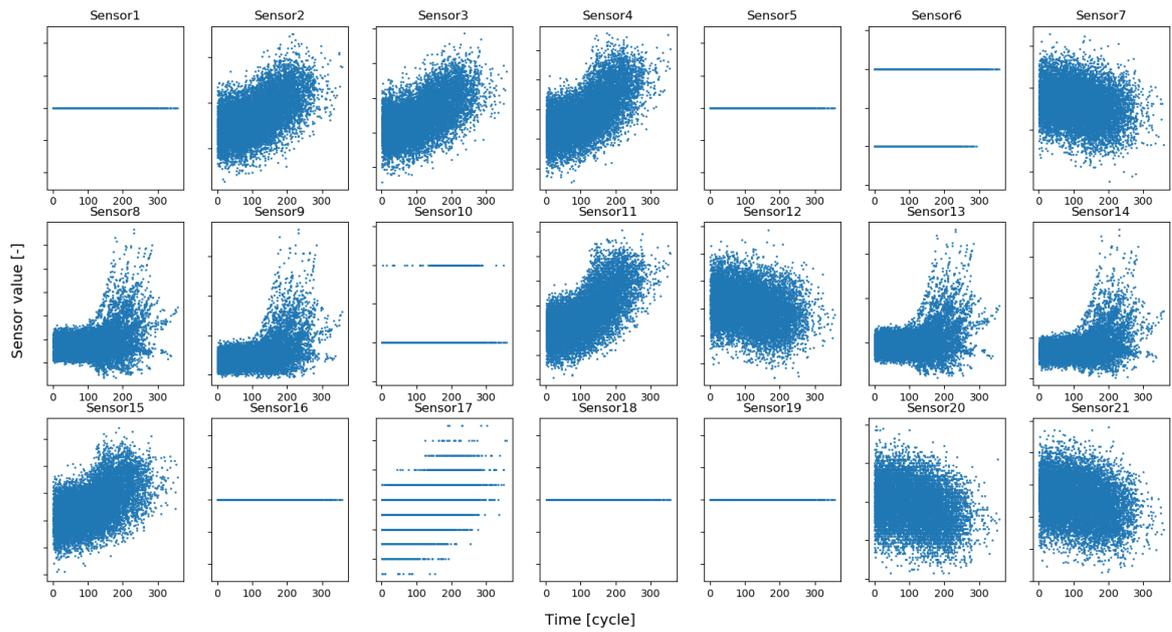


Figure 1.6: Sensor measurements of all sensors - PHM Challenge training data set operational mode 6.

1.1.2. Principal component analysis

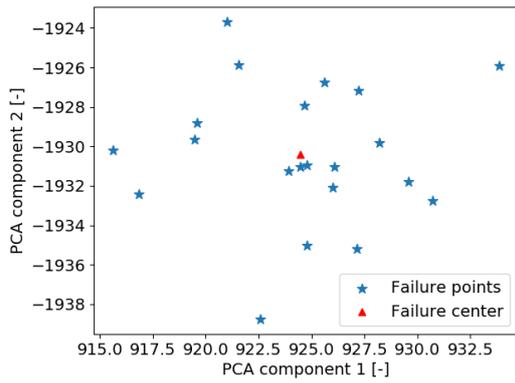
In [Table 1.1](#), the explained variance of the 7 principal components is provided of the PHM Challenge training data set. It can be seen that the first two principal components capture more than 98% of the variance of the data set, see [Part I](#) for the methodology on how to obtain the principal components.

Table 1.1: Explained variance of the principal components of the PHM Challenge training data set.

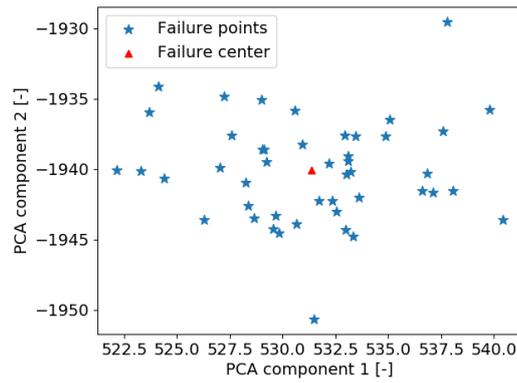
	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
PC1	60.85	58.95	79.65	72.64	61.45	54.41
PC2	38.04	40.07	19.12	26.75	37.85	44.55
PC3	0.66	0.40	0.77	0.28	0.34	0.56
PC4	0.28	0.31	0.39	0.18	0.23	0.25
PC5	0.14	0.24	0.07	0.13	0.12	0.20
PC6	0.02	0.02	0.00	0.02	0.01	0.03
PC7	0.00	0.00	0.00	0.00	0.00	0.00

1.1.3. Failure clusters per operational mode

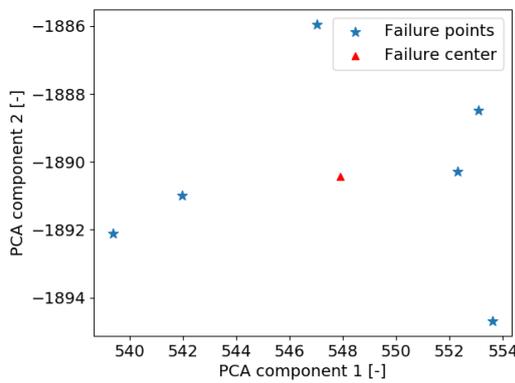
The six failure clusters as described in [Part I](#) for the PHM Challenge training data set can be viewed in [Figure 1.7](#). The overview of all failure clusters in the 2D principal component space can be viewed in [Figure 1.8](#).



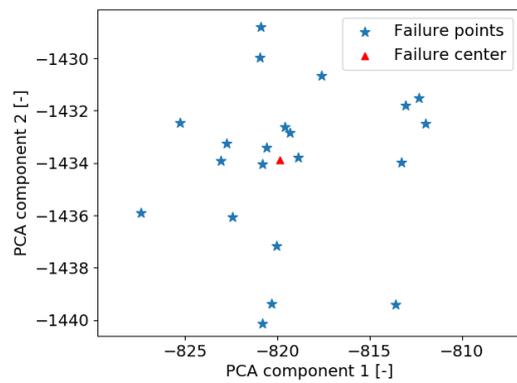
(a) Engines failing in operational mode 1 projected in the 2D principal component space.



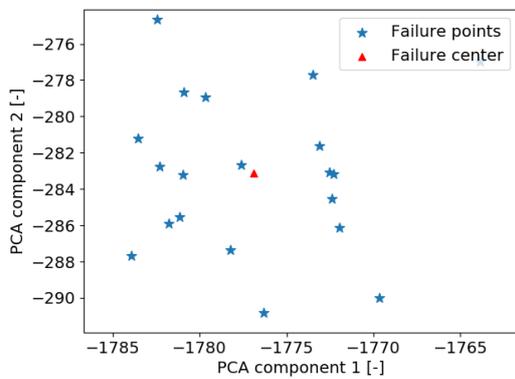
(b) Engines failing in operational mode 2 projected in the 2D principal component space.



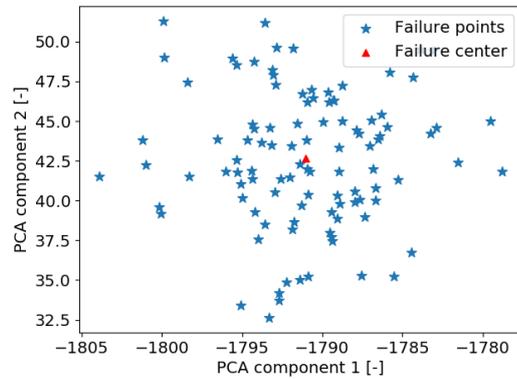
(c) Engines failing in operational mode 3 projected in the 2D principal component space.



(d) Engines failing in operational mode 4 projected in the 2D principal component space.



(e) Engines failing in operational mode 5 projected in the 2D principal component space.



(f) Engines failing in operational mode 6 projected in the 2D principal component space.

Figure 1.7: Engines failing per operational mode projected in the 2D principal component space for the PHM Challenge training data set. 7 sensor measurements are used to develop the 2D space, see Part I.

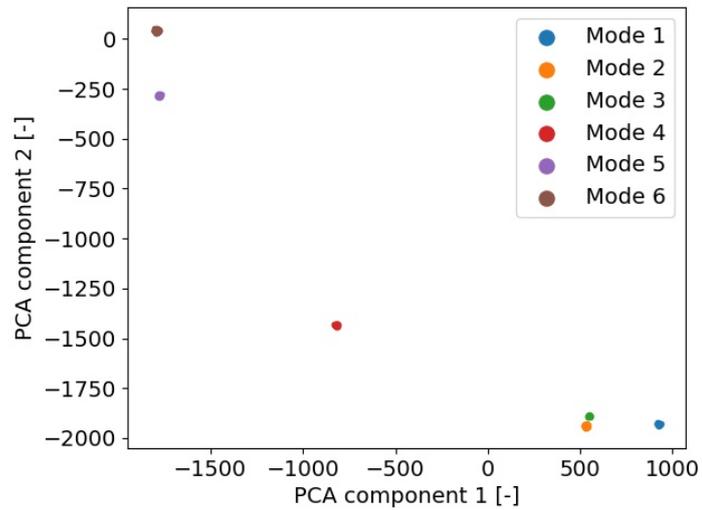
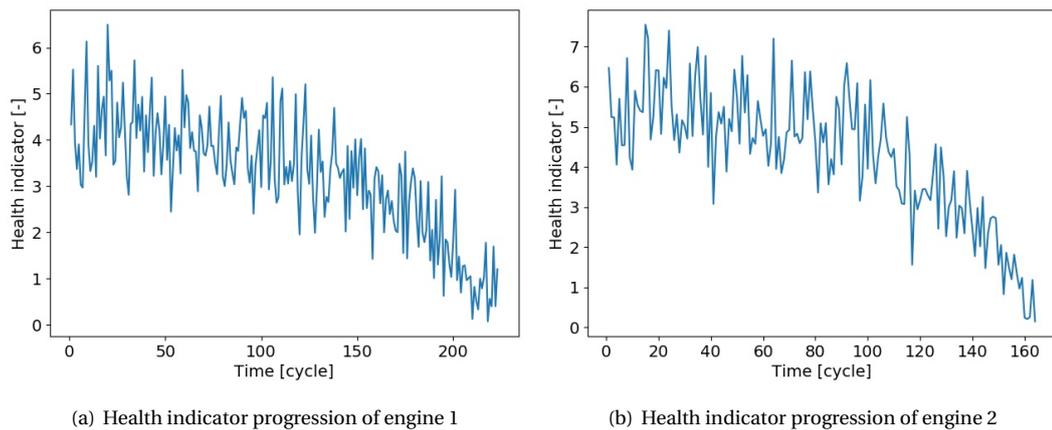


Figure 1.8: Projection of all failure clusters identified in Figure 1.7 in the principal component space.

1.1.4. Health indicator progression example

In Figure 1.9 the health indicator progression of the first 2 engines of the PHM Challenge training data set have been provided. The methodology on how to obtain these is provided in Part I.



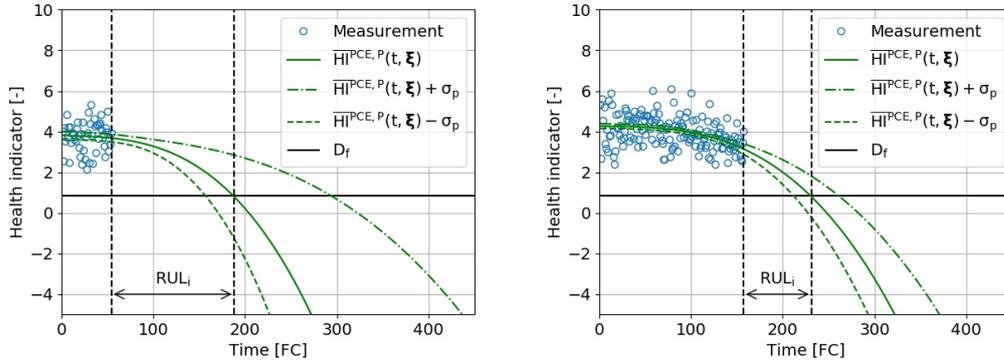
(a) Health indicator progression of engine 1

(b) Health indicator progression of engine 2

Figure 1.9: Constructed health indicator for the first 2 engines of the PHM challenge training data set.

1.1.5. RUL estimation example

In Figure 1.10, two engines of the PHM Challenge testing data set have been provided with their corresponding RUL estimation. The methodology on how to obtain these is provided in Part I.



(a) RUL prediction is performed 54 FC after the first time the engine is used. Estimated RUL=188-54=134 FC. Engine 1 of PHM challenge testing data set.
 (b) RUL prediction is performed 157 FC after the first time of the engine is used. Estimated RUL=231-157=74 FC. Engine 2 of PHM challenge testing data set.

Figure 1.10: RUL estimation for engines 1 and 2 of the PHM Challenge testing data set

1.1.6. Results aPCE model

In Table 1.2 the aPCE RUL prognostic model results are provided for the first 12 engines of the PHM Challenge testing data set. As this data set does not have a RUL verification set, some metrics cannot be evaluated.

Table 1.2: Results of the aPCE prognostic model applied to the first 12 aircraft engines of the PHM Challenge testing data set.

Engine ID [-]	RUL _i [FC]	95% C.I. low [FC]	95% C.I. high [FC]	Std. dev σ_p [FC]
Engine_1	134	112	151	37
Engine_2	74	68	80	24
Engine_3	87	76	97	29
Engine_4	115	110	119	30
Engine_5	57	52	62	9
Engine_6	100	90	109	28
Engine_7	59	53	64	18
Engine_8	25	14	35	11
Engine_9	81	75	87	17
Engine_10	128	108	146	25
Engine_11	116	114	119	29
Engine_12	32	27	37	7
Engine_12	32	27	37	7

1.2. FD001 data set

1.2.1. Sensor measurements

In [Figure 1.11](#) all sensor measurements of all engines are provided for the FD001 training data set. This data set consists of only 1 operational mode.

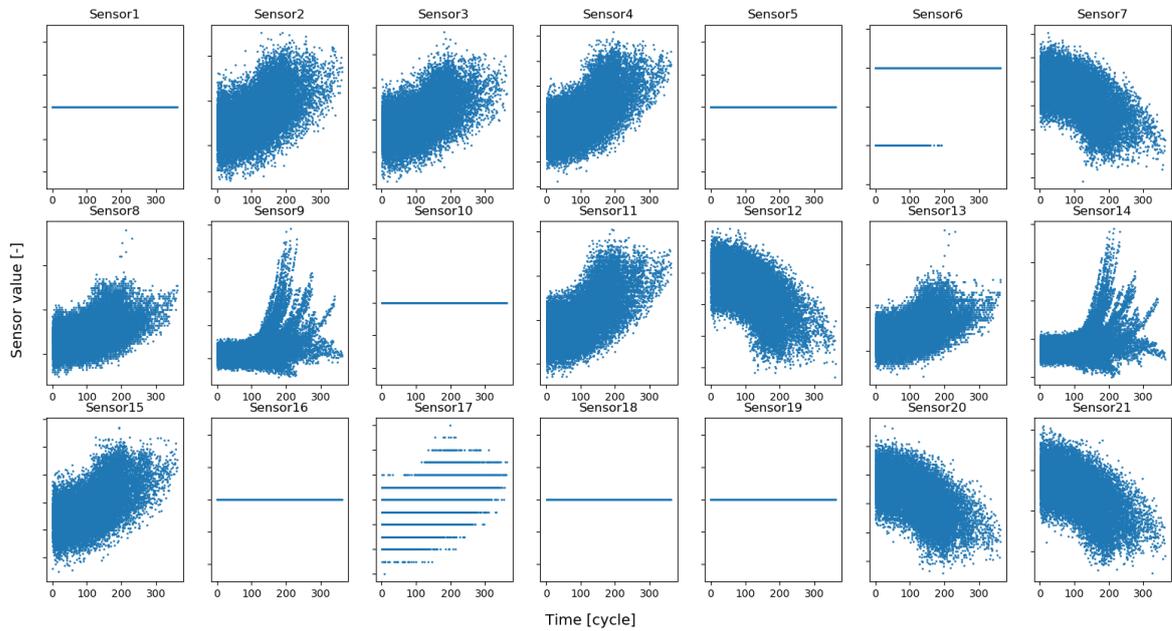


Figure 1.11: Sensor measurements of all sensors - FD001 training data set operational mode 1.

1.2.2. Principal component analysis

In [Table 1.3](#), the explained variance of the 7 principal components is provided of the FD001 training data set. It can be seen that the first two principal components capture more than 98% of the variance of the data set, see [Part I](#) for the methodology on how to obtain the principal components.

Table 1.3: Explained variance of the principal components of the FD001 training data set.

	Mode 1
PC1	55.42
PC2	43.34
PC3	0.67
PC4	0.30
PC5	0.23
PC6	0.04
PC7	0.00

1.2.3. Failure clusters

The failure cluster in the 2D principal component space can be viewed in [Figure 1.12](#).

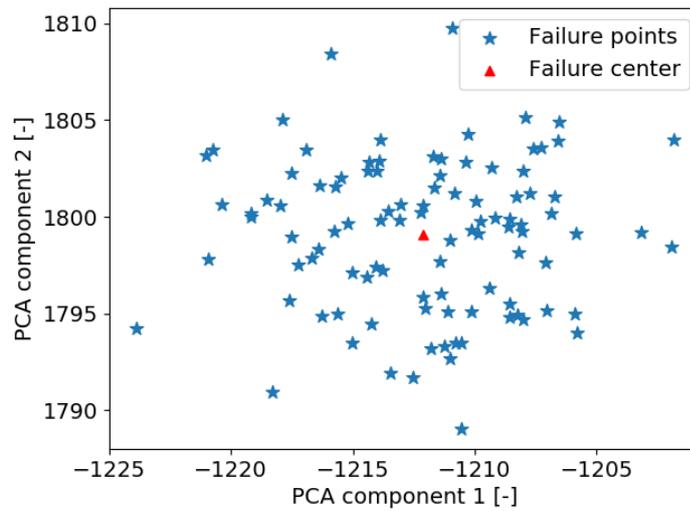


Figure 1.12: Projection of the failure cluster in the principal component space using the FD001 training data set.

1.2.4. Health indicator progression example

In [Figure 1.13](#) the health indicator progression of the first 2 engines of the FD001 training data set have been provided. The methodology on how to obtain these is provided in [Part I](#).

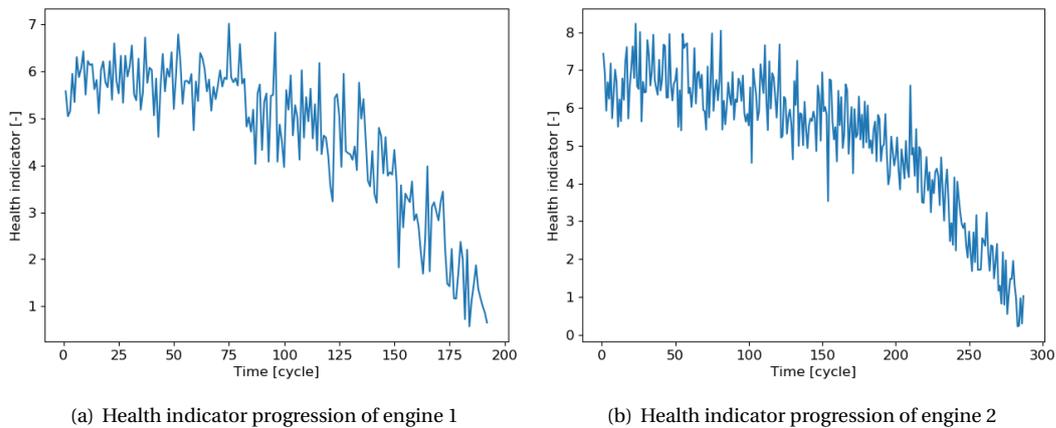
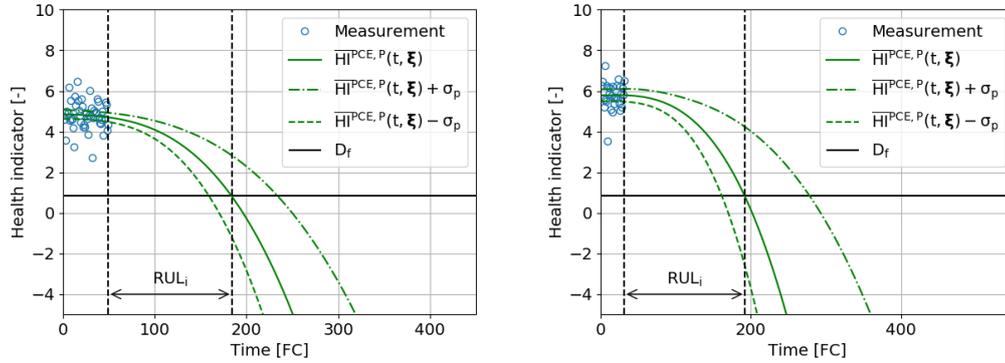


Figure 1.13: Constructed health indicator for the first 2 engines of the FD001 training data set.

1.2.5. RUL estimation example

In Figure 1.14, the first two engines of the FD001 testing data set have been provided with their corresponding RUL estimation. The methodology on how to obtain these is provided in Part I.



(a) RUL prediction is performed 31 FC after the first time the engine is used. Estimated RUL=192-31=161 FC. Engine 1 of FD001 testing data set.
 (b) RUL prediction is performed 49 FC after the first time the engine is used. Estimated RUL=184-49=135 FC. Engine 2 of FD001 testing data set.

Figure 1.14: RUL estimation for engines 1 and 2 of the FD001 testing data set

1.2.6. Results aPCE model

In Table 1.4 the aPCE RUL prognostic model results are provided for the first 12 engines of the FD001 testing data set.

Table 1.4: Results of the aPCE prognostic model applied to the first 12 aircraft engines of the FD001 testing data set.

Engine ID [-]	RUL _i [FC]	RUL _i ^a [FC]	95% C.I. low [FC]	95% C.I. high [FC]	Std. dev σ_p [FC]	Error ϵ_i [FC]	Score S_i [-]	CRPS [-]
Engine_1	160	112	150	168	36	48	120.5	27.2
Engine_2	136	98	118	151	25	38	43.7	92.4
Engine_3	61	69	39	79	19	-8	0.9	47.5
Engine_4	70	82	67	74	18	-12	1.5	22.3
Engine_5	87	91	71	101	23	-4	0.4	39.4
Engine_6	86	93	66	103	25	-7	0.7	32.4
Engine_7	94	91	82	105	18	3	0.3	18.6
Engine_8	100	95	88	112	20	5	0.6	6.3
Engine_9	126	111	103	144	32	15	3.5	53.7
Engine_10	91	96	75	106	13	-5	0.5	7.1
Engine_11	84	97	63	101	28	-13	1.7	10.8
Engine_12	93	124	34	136	25	-31	9.9	14.5

1.3. FD002 data set

1.3.1. Sensor measurements split per operational mode

In Figure 1.15 to Figure 1.20 all sensor measurements of all engines split per operational mode are provided for the FD002 training data set.

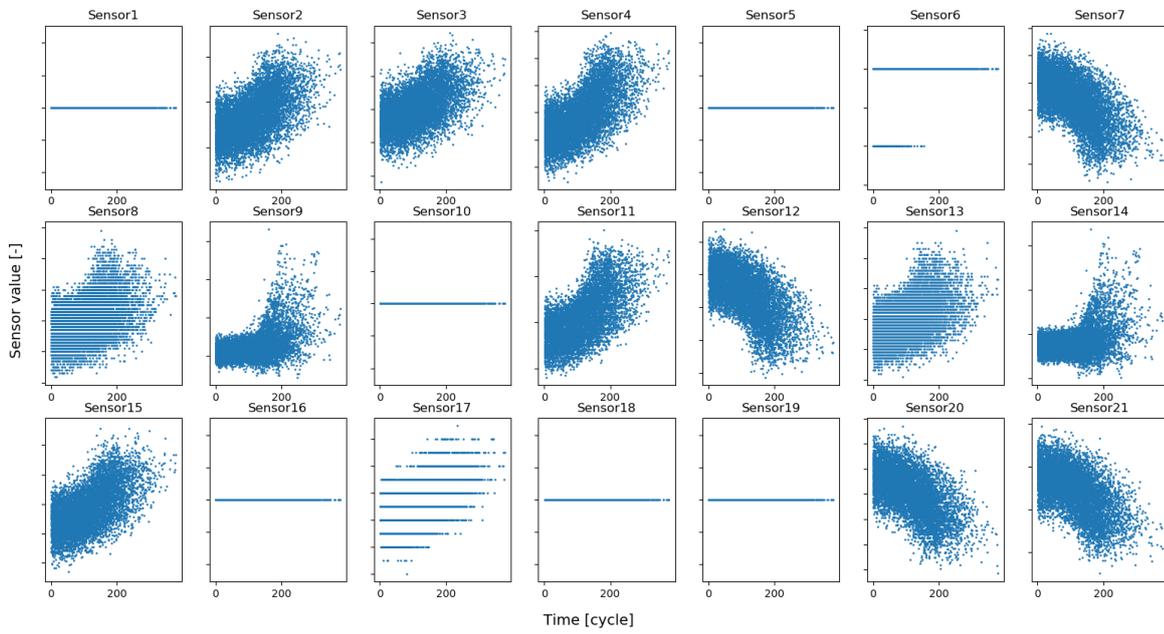


Figure 1.15: Sensor measurements of all sensors - FD002 training data set operational mode 1.

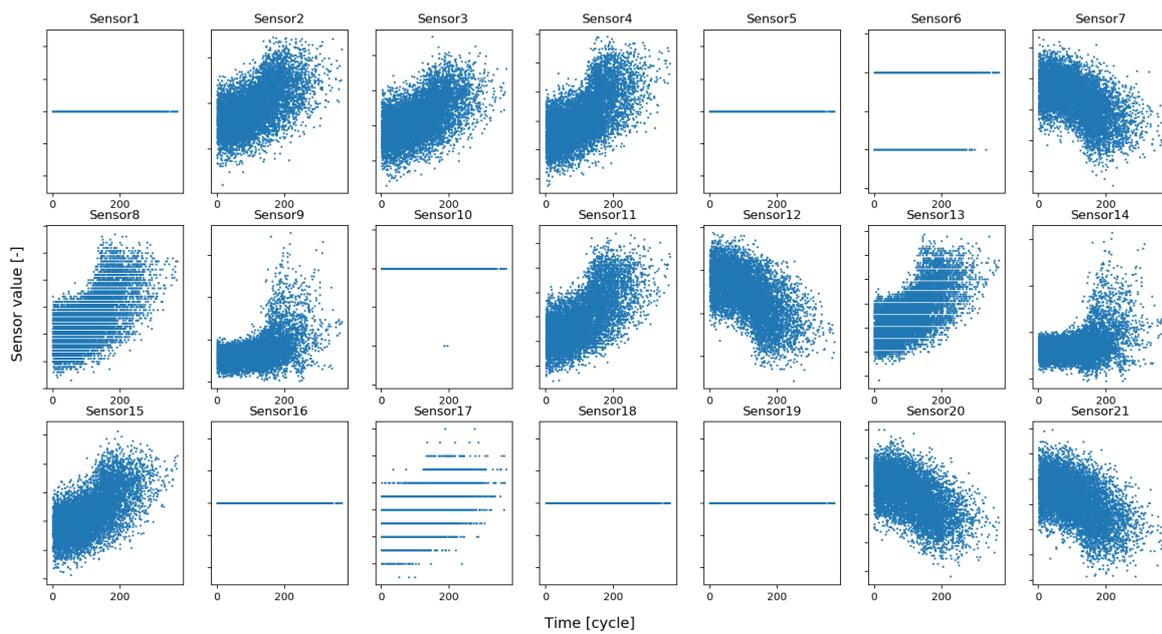


Figure 1.16: Sensor measurements of all sensors - FD002 training data set operational mode 2.

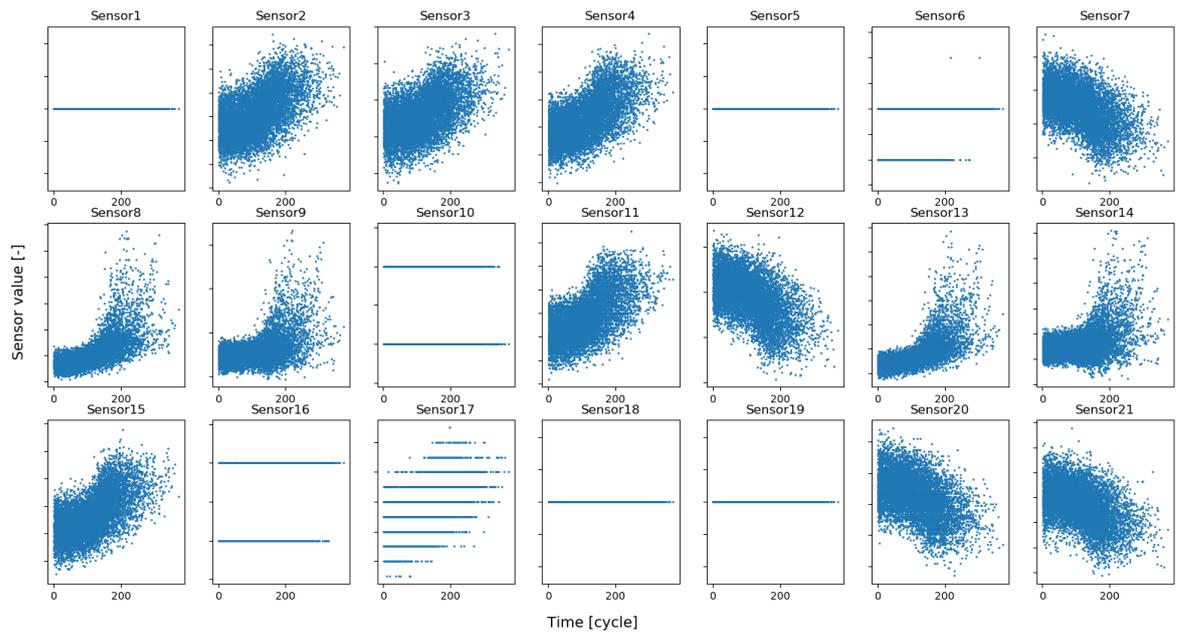


Figure 1.17: Sensor measurements of all sensors - FD002 training data set operational mode 3.

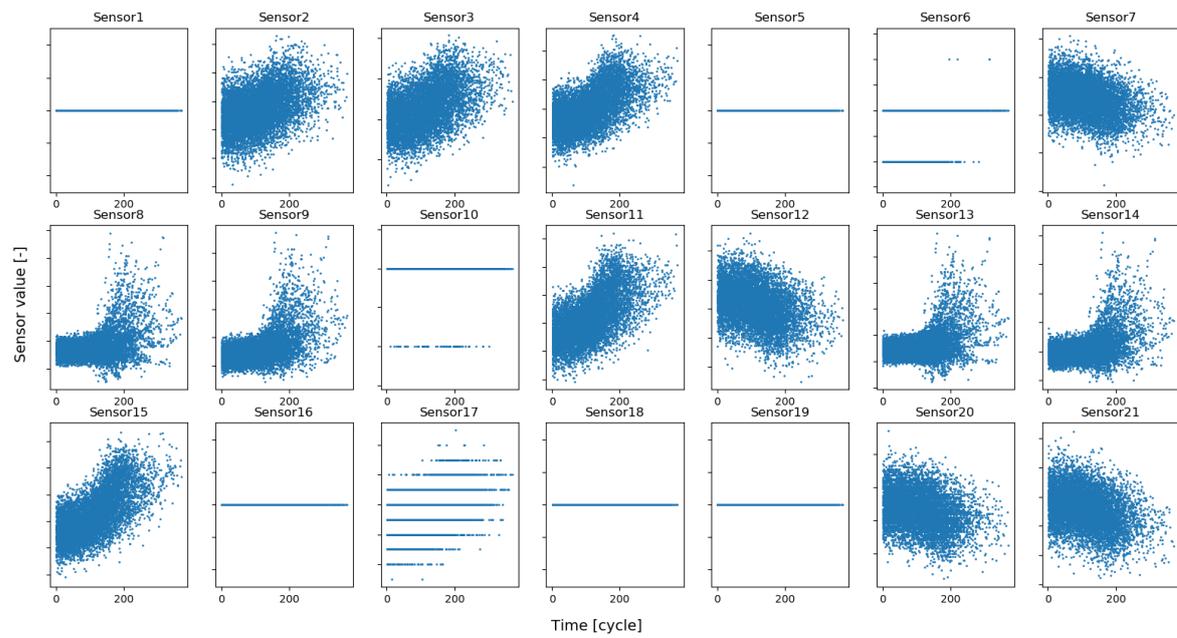


Figure 1.18: Sensor measurements of all sensors - FD002 training data set operational mode 4.

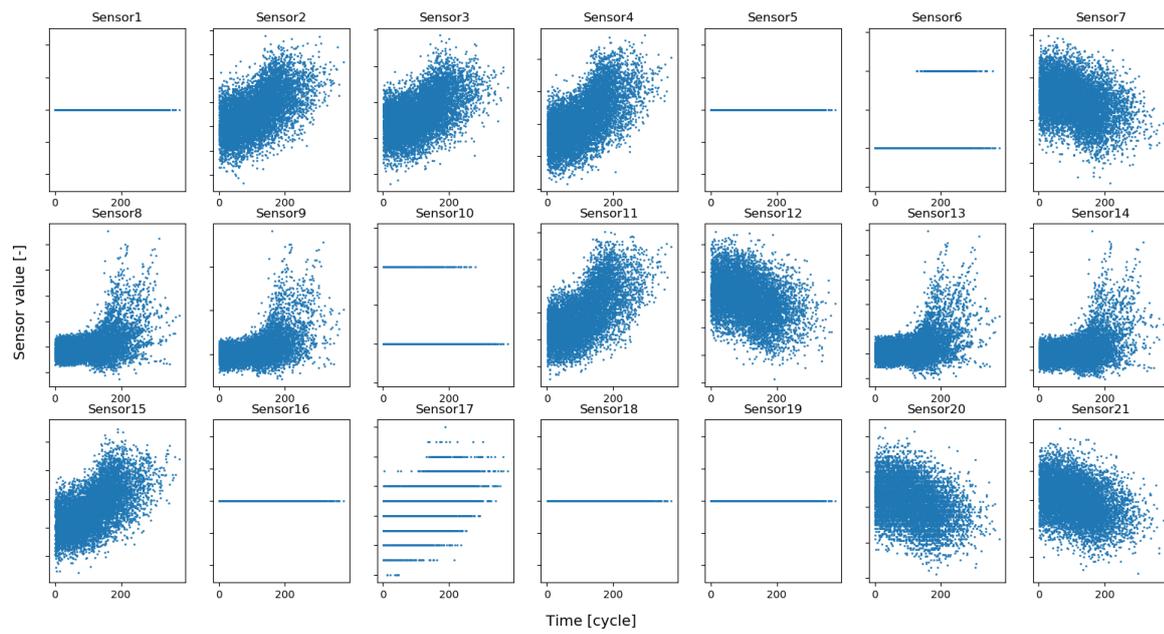


Figure 1.19: Sensor measurements of all sensors - FD002 training data set operational mode 5.

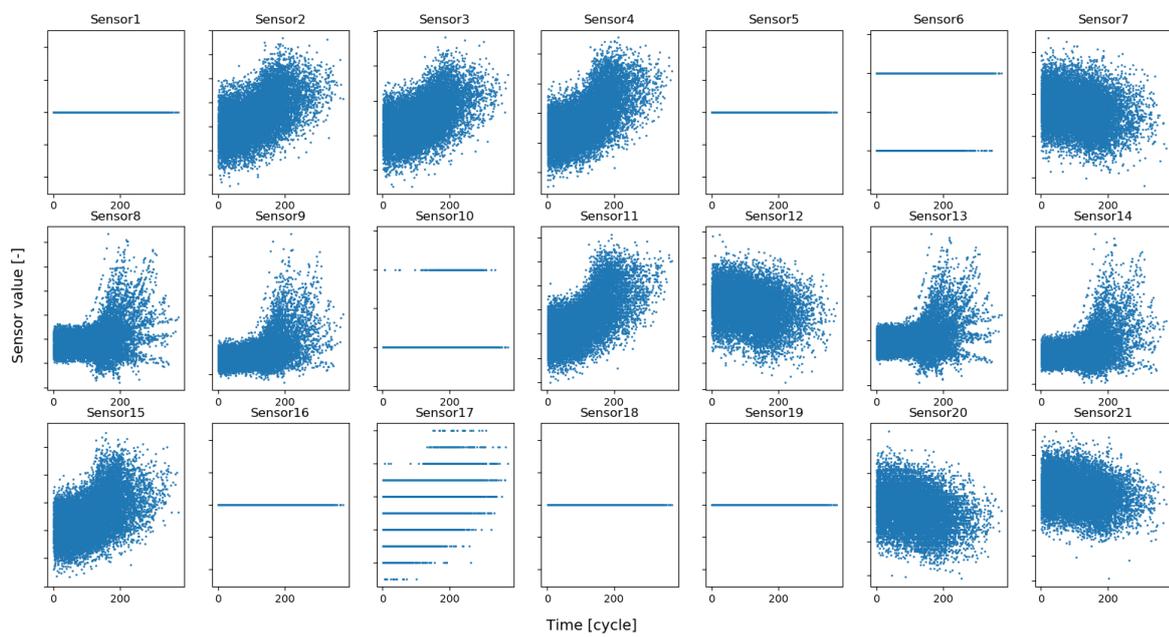


Figure 1.20: Sensor measurements of all sensors - FD002 training data set operational mode 6.

1.3.2. Principal component analysis

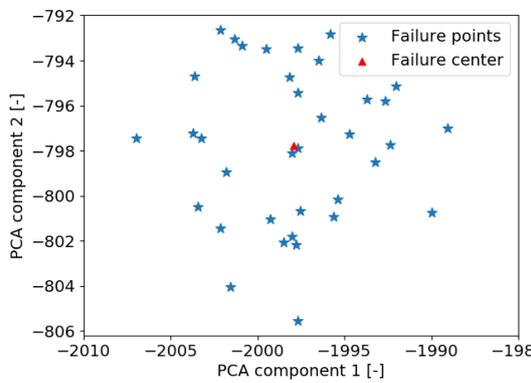
In [Table 1.5](#), the explained variance of the 7 principal components is provided of the FD002 training data set. It can be seen that the first two principal components capture more than 98% of the variance of the data set, see [Part I](#) for the methodology on how to obtain the principal components.

Table 1.5: Explained variance of the principal components of the FD002 training data set.

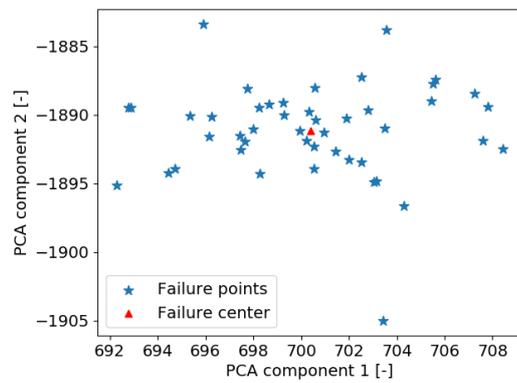
	Mode 1	Mode 2	Mode 3	Mode 4	Mode 5	Mode 6
PC1	58.97	57.86	70.65	59.89	60.98	59.92
PC2	39.87	40.89	28.75	39.03	38.28	38.99
PC3	0.59	0.74	0.44	0.53	0.35	0.56
PC4	0.29	0.26	0.10	0.41	0.26	0.29
PC5	0.25	0.22	0.04	0.12	0.11	0.21
PC6	0.02	0.02	0.02	0.02	0.01	0.03
PC7	0.00	0.00	0.00	0.00	0.00	0.00

1.3.3. Failure clusters per operational mode

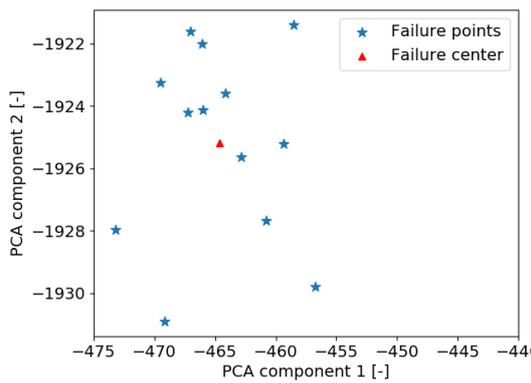
The six failure clusters as described in Part I for the FD002 training data set can be viewed in Figure 1.21. The overview of all failure clusters in the 2D principal component space can be viewed in Figure 1.22.



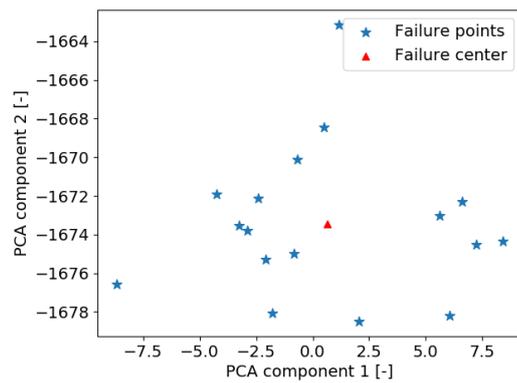
(a) Engines failing in operational mode 1 projected in the 2D principal component space.



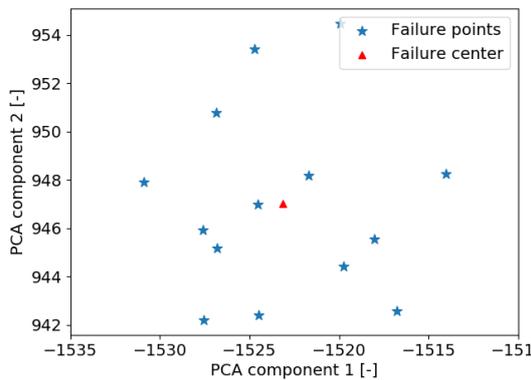
(b) Engines failing in operational mode 2 projected in the 2D principal component space.



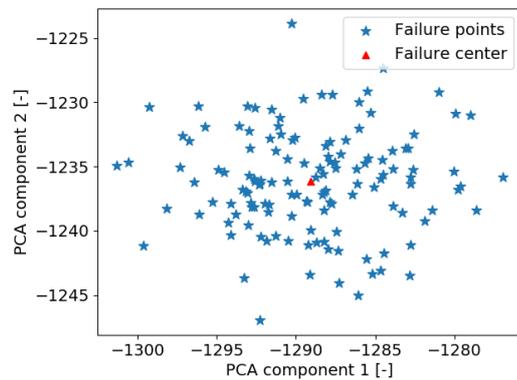
(c) Engines failing in operational mode 3 projected in the 2D principal component space.



(d) Engines failing in operational mode 4 projected in the 2D principal component space.



(e) Engines failing in operational mode 5 projected in the 2D principal component space.



(f) Engines failing in operational mode 6 projected in the 2D principal component space.

Figure 1.21: Engines failing per operational mode projected in the 2D principal component space for the FD002 training data set. 7 sensor measurements are used to develop the 2D space, see Part I.

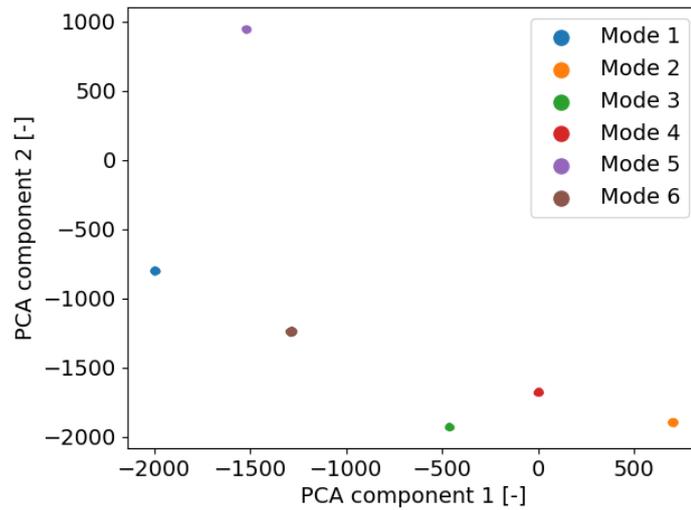
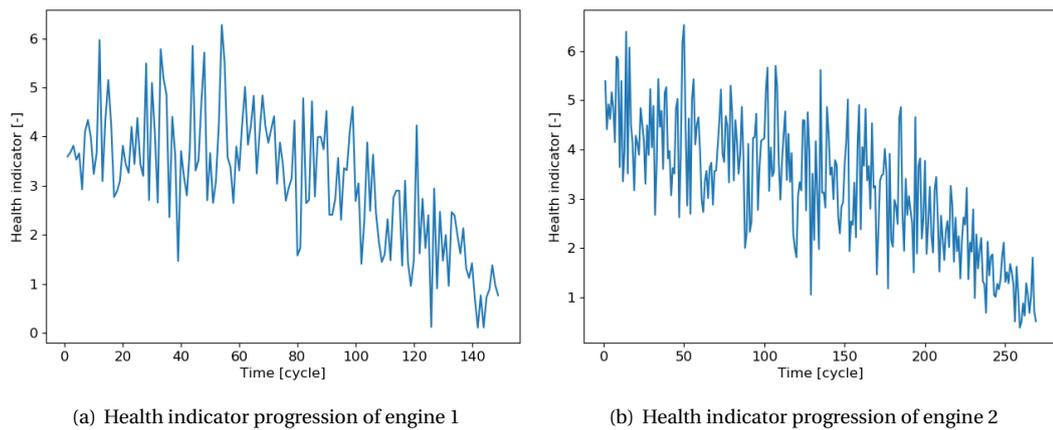


Figure 1.22: Projection of all failure clusters identified in Figure 1.21 in the principal component space.

1.3.4. Health indicator progression example

In Figure 1.23 the health indicator progression of the first 2 engines of the FD002 training data set have been provided. The methodology on how to obtain these is provided in Part I.



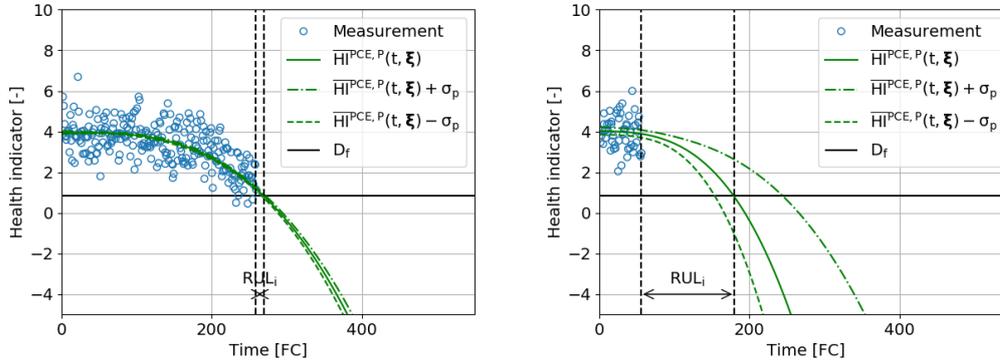
(a) Health indicator progression of engine 1

(b) Health indicator progression of engine 2

Figure 1.23: Constructed health indicator for the first 2 engines of the FD002 challenge training data set.

1.3.5. RUL estimation example

In Figure 1.24, the first two engines of the FD002 testing data set have been provided with their corresponding RUL estimation. The methodology on how to obtain these is provided in Part I.



(a) [RUL prediction is performed 258 FC after the first time the engine is used. Estimated RUL=271-258=13 FC. Engine 1 of FD002 testing data set.] (b) [RUL prediction is performed 55 FC after the first time the engine is used. Estimated RUL=180-55=125 FC. Engine 2 of FD002 testing data set.]

Figure 1.24: RUL estimation for engines 1 and 2 of the FD002 testing data set

1.3.6. Results aPCE model

In Table 1.6 the aPCE RUL prognostic model results are provided for the first 12 engines of the FD002 testing data set.

Table 1.6: Results of the aPCE prognostic model applied to the first 12 aircraft engines of the FD002 testing data set.

Engine ID [-]	RUL _i [FC]	RUL _i ^a [FC]	95% C.I. low [FC]	95% C.I. high [FC]	Std. dev σ_p [FC]	Error ϵ_i [FC]	Score S_i [-]	CRPS [-]
Engine 1	24	18	16	31	11	6	0.82	4.73
Engine 2	125	79	121	129	30	46	98.48	25.25
Engine 3	76	106	67	84	19	-30	9.05	22.57
Engine 4	90	110	83	96	29	-20	3.66	17.11
Engine 5	22	15	18	25	9	7	1.01	5.02
Engine 6	130	155	126	134	32	-25	5.84	19.73
Engine 7	8	6	6	10	6	2	0.22	2.24
Engine 8	76	90	66	86	22	-14	1.94	10.31
Engine 9	8	11	6	11	6	-3	0.26	1.71
Engine 10	91	79	86	96	27	12	2.32	7.02
Engine 11	6	6	4	8	5	0	0.00	1.39
Engine 12	106	73	101	111	29	33	26.11	15.62

2

Additional results: component level optimisation

In this section, the results of the component level optimisation model are presented for selected engines for the case study in [Part I](#). This entails, that the first 200 engines of the FD002 training data set have been used for training and the remaining 60 engines have been used for model evaluation. The MDP maintenance planning model has been used to obtain these results. The results are presented in [Table 2.1](#) to [Table 2.10](#). TTM is short for time to maintenance. Predicted TTM is the difference between the optimal MDP time t^* and the RUL prediction time. Perfect TTM is the difference between the engine actual lifetime and the RUL prediction time. The error is defined as the difference between predicted TTM and perfect TTM.

Table 2.1: Optimal aircraft engine maintenance times for engines 201 to 203 from the FD002 training data set

Engine ID [-]	RUL prediction time [FC]	Optimal state [-]	Optimal MDP time (t^*) [FC]	Predicted TTM [FC]	Perfect TTM [FC]	Error [FC]
Engine 201 (actual lifetime =191 FC)	47	45	189	142	144	-2
	67	46	188	121	124	-3
	87	46	208	121	104	17
	107	45	178	71	84	-13
	127	45	169	42	64	-22
	147	45	187	40	44	-4
	167	45	186	19	24	-5
	187	46	187	0	4	-4
Engine 202 (actual lifetime =197 FC)	49	45	173	124	148	-24
	69	45	177	108	128	-20
	89	45	171	82	108	-26
	109	45	173	64	88	-24
	129	45	179	50	68	-18
	149	46	180	31	48	-17
	169	45	181	12	28	-16
	189	50	189	0	8	-8
Engine 203 (actual lifetime =269 FC)	67	45	171	104	202	-98
	87	45	178	91	182	-91
	107	45	185	78	162	-84
	127	46	203	76	142	-66
	147	46	223	76	122	-46
	167	46	241	74	102	-28
	207	46	241	34	62	-28
	187	46	244	57	82	-25
	227	46	250	23	42	-19
	247	46	259	12	22	-10
267	50	267	0	2	-2	

Table 2.2: Optimal aircraft engine maintenance times for engines 204 to 209 from the FD002 training data set

Engine ID [-]	RUL prediction time [FC]	Optimal state [-]	Optimal MDP time (t*) [FC]	Predicted TTM [FC]	Perfect TTM [FC]	Error [FC]
Engine 204 (actual lifetime =237 FC)	59	46	174	115	178	-63
	79	45	167	88	158	-70
	99	46	178	79	138	-59
	119	46	181	62	118	-56
	139	46	183	44	98	-54
	159	46	202	43	78	-35
	179	46	220	41	58	-17
	199	46	222	23	38	-15
219	46	221	2	18	-16	
Engine 205 (actual lifetime =198 FC)	49	46	174	125	149	-24
	69	45	174	105	129	-24
	89	46	173	84	109	-25
	109	45	174	65	89	-24
	129	46	185	56	69	-13
	149	46	190	41	49	-8
	169	45	189	20	29	-9
189	48	189	0	9	-9	
Engine 206 (actual lifetime =221 FC)	55	44	187	132	166	-34
	75	45	193	118	146	-28
	95	45	197	102	126	-24
	115	45	216	101	106	-5
	135	45	227	92	86	6
	155	46	253	98	66	32
	175	45	242	67	46	21
195	45	248	53	26	27	
215	44	222	7	6	1	
Engine 207 (actual lifetime =184 FC)	46	46	180	134	138	-4
	66	46	179	113	118	-5
	86	45	175	89	98	-9
	106	46	182	76	78	-2
	126	45	175	49	58	-9
	146	45	172	26	38	-12
166	45	175	9	18	-9	
Engine 208 (actual lifetime =150 FC)	37	45	176	139	113	26
	57	45	173	116	93	23
	77	45	173	96	73	23
	97	46	171	74	53	21
	117	45	154	37	33	4
137	43	137	0	13	-13	
Engine 209 (actual lifetime =201 FC)	50	45	172	122	151	-29
	70	45	170	100	131	-31
	90	46	175	85	111	-26
	110	46	189	79	91	-12
	130	46	195	65	71	-6
	150	46	195	45	51	-6
	170	45	189	19	31	-12
190	46	190	0	11	-11	

Table 2.3: Optimal aircraft engine maintenance times for engines 210 to 216 from the FD002 training data set

Engine ID [-]	RUL prediction time [FC]	Optimal state [-]	Optimal MDP time (t*) [FC]	Predicted TTM [FC]	Perfect TTM [FC]	Error [FC]
Engine 210 (actual lifetime =184 FC)	46	45	178	132	138	-6
	66	46	180	114	118	-4
	86	45	169	83	98	-15
	106	46	181	75	78	-3
	126	46	181	55	58	-3
	146	45	189	43	38	5
Engine 211 (actual lifetime =214 FC)	166	45	179	13	18	-5
	53	45	177	124	160	-36
	73	45	183	110	140	-30
	93	46	183	90	120	-30
	113	46	205	92	100	-8
	133	46	217	84	80	4
	153	46	213	60	60	0
Engine 212 (actual lifetime =149 FC)	173	46	216	43	40	3
	193	45	212	19	20	-1
	213	49	213	0	0	0
	37	46	191	154	112	42
	57	46	187	130	92	38
Engine 213 (actual lifetime =196 FC)	77	45	178	101	72	29
	97	45	164	67	52	15
	117	43	145	28	32	-4
	137	43	140	3	12	-9
	49	45	193	144	147	-3
Engine 214 (actual lifetime =146 FC)	69	46	196	127	127	0
	89	45	186	97	107	-10
	109	46	201	92	87	5
	129	46	196	67	67	0
	149	46	213	64	47	17
	169	45	207	38	27	11
Engine 215 (actual lifetime =226 FC)	189	45	195	6	7	-1
	36	45	170	134	110	24
	56	45	172	116	90	26
	76	45	162	86	70	16
	96	45	167	71	50	21
Engine 216 (actual lifetime =229 FC)	116	44	146	30	30	0
	136	43	136	0	10	-10
	56	44	187	131	170	-39
	76	44	190	114	150	-36
	96	45	195	99	130	-31
	116	45	210	94	110	-16
	136	45	223	87	90	-3
	156	46	254	98	70	28
Engine 216 (actual lifetime =229 FC)	176	46	266	90	50	40
	196	45	245	49	30	19
	216	44	222	6	10	-4
	57	46	191	134	172	-38
	77	45	191	114	152	-38
	97	46	205	108	132	-24
	117	46	200	83	112	-29
Engine 216 (actual lifetime =229 FC)	137	46	213	76	92	-16
	157	45	205	48	72	-24
	177	46	213	36	52	-16
	197	45	215	18	32	-14
	217	47	217	0	12	-12

Table 2.4: Optimal aircraft engine maintenance times for engines 217 to 223 from the FD002 training data set

Engine ID [-]	RUL prediction time [FC]	Optimal state [-]	Optimal MDP time (t*) [FC]	Predicted TTM [FC]	Perfect TTM [FC]	Error [FC]
Engine 217 (actual lifetime =162 FC)	40	46	192	152	122	30
	60	46	184	124	102	22
	80	46	178	98	82	16
	100	46	175	75	62	13
	120	44	155	35	42	-7
	140	45	159	19	22	-3
Engine 218 (actual lifetime =237 FC)	160	48	160	0	2	-2
	59	45	174	115	178	-63
	79	45	171	92	158	-66
	99	45	176	77	138	-61
	119	45	191	72	118	-46
	139	46	220	81	98	-17
Engine 219 (actual lifetime =156 FC)	159	46	209	50	78	-28
	179	46	218	39	58	-19
	199	46	222	23	38	-15
	219	46	224	5	18	-13
	39	46	181	142	117	25
	59	45	174	115	97	18
Engine 220 (actual lifetime =234 FC)	79	45	167	88	77	11
	99	45	169	70	57	13
	119	45	169	50	37	13
	139	44	156	17	17	0
	58	45	175	117	176	-59
	78	45	179	101	156	-55
Engine 221 (actual lifetime =178 FC)	98	45	180	82	136	-54
	118	46	188	70	116	-46
	138	46	204	66	96	-30
	158	46	203	45	76	-31
	178	46	217	39	56	-17
	198	46	214	16	36	-20
Engine 222 (actual lifetime =195 FC)	218	48	218	0	16	-16
	44	46	174	130	134	-4
	64	46	175	111	114	-3
	84	46	174	90	94	-4
	104	45	173	69	74	-5
	124	46	170	46	54	-8
Engine 223 (actual lifetime =218 FC)	144	45	175	31	34	-3
	164	45	169	5	14	-9
	48	45	176	128	147	-19
	68	45	171	103	127	-24
	88	45	175	87	107	-20
	108	46	192	84	87	-3
Engine 223 (actual lifetime =218 FC)	128	45	188	60	67	-7
	148	45	188	40	47	-7
	168	45	190	22	27	-5
	188	45	188	0	7	-7
	54	45	183	129	164	-35
	74	45	186	112	144	-32
Engine 223 (actual lifetime =218 FC)	94	46	194	100	124	-24
	114	45	187	73	104	-31
	134	46	203	69	84	-15
	154	46	214	60	64	-4
	174	46	223	49	44	5
	194	45	217	23	24	-1
214	46	216	2	4	-2	

Table 2.5: Optimal aircraft engine maintenance times for engines 224 to 230 from the FD002 training data set

Engine ID [-]	RUL prediction time [FC]	Optimal state [-]	Optimal MDP time (t*) [FC]	Predicted TTM [FC]	Perfect TTM [FC]	Error [FC]
Engine 224 (actual lifetime =174 FC)	43	46	183	140	131	9
	63	46	180	117	111	6
	83	46	180	97	91	6
	103	45	163	60	71	-11
	123	45	169	46	51	-5
	143	45	173	30	31	-1
Engine 225 (actual lifetime =156 FC)	163	45	166	3	11	-8
	39	46	186	147	117	30
	59	46	180	121	97	24
	79	46	175	96	77	19
	99	45	172	73	57	16
Engine 226 (actual lifetime =178 FC)	119	46	170	51	37	14
	139	44	153	14	17	-3
	44	46	174	130	134	-4
	64	45	173	109	114	-5
	84	45	173	89	94	-5
	104	46	180	76	74	2
Engine 227 (actual lifetime =184 FC)	124	45	172	48	54	-6
	144	45	172	28	34	-6
	164	45	167	3	14	-11
	46	46	175	129	138	-9
	66	45	170	104	118	-14
	86	45	171	85	98	-13
Engine 228 (actual lifetime =262 FC)	106	46	183	77	78	-1
	126	46	185	59	58	1
	146	45	176	30	38	-8
	166	45	174	8	18	-10
	65	45	176	111	197	-86
	85	45	177	92	177	-85
	105	45	190	85	157	-72
	125	46	228	103	137	-34
145	46	231	86	117	-31	
Engine 229 (actual lifetime =176 FC)	165	46	261	96	97	-1
	185	46	251	66	77	-11
	205	46	254	49	57	-8
	225	46	257	32	37	-5
	245	46	250	5	17	-12
	44	44	177	133	132	1
	64	44	176	112	112	0
84	44	175	91	92	-1	
104	44	173	69	72	-3	
124	44	185	61	52	9	
144	44	188	44	32	12	
164	43	174	10	12	-2	
Engine 230 (actual lifetime =186 FC)	46	45	192	146	140	6
	66	45	189	123	120	3
	86	46	183	97	100	-3
	106	46	194	88	80	8
	126	46	190	64	60	4
	146	45	183	37	40	-3
166	45	177	11	20	-9	

Table 2.6: Optimal aircraft engine maintenance times for engines 231 to 237 from the FD002 training data set

Engine ID [-]	RUL prediction time [FC]	Optimal state [-]	Optimal MDP time (t*) [FC]	Predicted TTM [FC]	Perfect TTM [FC]	Error [FC]
Engine 231 (actual lifetime =144 FC)	36	45	192	156	108	48
	56	46	185	129	88	41
	76	45	173	97	68	29
	96	45	165	69	48	21
	116	44	138	22	28	-6
	136	44	136	0	8	-8
Engine 232 (actual lifetime =242 FC)	60	45	172	112	182	-70
	80	46	175	95	162	-67
	100	45	176	76	142	-66
	120	46	200	80	122	-42
	140	46	210	70	102	-32
	160	46	229	69	82	-13
	180	46	230	50	62	-12
	200	46	235	35	42	-7
	220	46	234	14	22	-8
240	50	240	0	2	-2	
Engine 233 (actual lifetime =190 FC)	47	46	187	140	143	-3
	67	46	184	117	123	-6
	87	45	178	91	103	-12
	107	46	189	82	83	-1
	127	45	191	64	63	1
	147	45	185	38	43	-5
	167	45	185	18	23	-5
	187	47	187	0	3	-3
Engine 234 (actual lifetime =159 FC)	39	45	172	133	120	13
	59	45	174	115	100	15
	79	46	177	98	80	18
	99	45	174	75	60	15
	119	45	161	42	40	2
	139	44	150	11	20	-9
Engine 235 (actual lifetime =183 FC)	45	45	170	125	138	-13
	65	45	168	103	118	-15
	85	46	173	88	98	-10
	105	46	172	67	78	-11
	125	45	175	50	58	-8
	145	45	170	25	38	-13
165	45	169	4	18	-14	
Engine 236 (actual lifetime =239 FC)	59	45	171	112	180	-68
	79	45	174	95	160	-65
	99	45	175	76	140	-64
	119	46	189	70	120	-50
	139	46	196	57	100	-43
	159	46	202	43	80	-37
	179	45	207	28	60	-32
	199	46	217	18	40	-22
	219	49	219	0	20	-20
Engine 237 (actual lifetime =209 FC)	52	46	174	122	157	-35
	72	45	168	96	137	-41
	92	46	183	91	117	-26
	112	46	196	84	97	-13
	132	46	198	66	77	-11
	152	46	202	50	57	-7
	172	46	210	38	37	1
	192	45	200	8	17	-9

Table 2.7: Optimal aircraft engine maintenance times for engines 238 to 244 from the FD002 training data set

Engine ID [-]	RUL prediction time [FC]	Optimal state [-]	Optimal MDP time (t*) [FC]	Predicted TTM [FC]	Perfect TTM [FC]	Error [FC]
Engine 238 (actual lifetime =159 FC)	39	45	171	132	117	15
	59	46	176	117	97	20
	79	45	177	98	77	21
	99	45	171	72	57	15
	119	45	156	37	37	0
	139	44	156	17	17	0
Engine 239 (actual lifetime =263 FC)	65	45	177	112	198	-86
	85	45	182	97	178	-81
	105	45	178	73	158	-85
	125	46	190	65	138	-73
	145	46	196	51	118	-67
	165	45	211	46	98	-52
	185	46	232	47	78	-31
	205	46	238	33	58	-25
	225	47	253	28	38	-10
245	49	245	0	18	-18	
Engine 240 (actual lifetime =266 FC)	66	46	181	115	200	-85
	86	45	183	97	180	-83
	106	45	185	79	160	-81
	126	46	213	87	140	-53
	146	46	222	76	120	-44
	166	46	244	78	100	-22
	186	46	243	57	80	-23
	206	46	253	47	60	-13
	226	46	254	28	40	-12
246	46	257	11	20	-9	
Engine 241 (actual lifetime =183 FC)	45	45	173	128	138	-10
	65	45	174	109	118	-9
	85	46	173	88	98	-10
	105	45	166	61	78	-17
	125	45	166	41	58	-17
	145	44	173	28	38	-10
165	45	171	6	18	-12	
Engine 242 (actual lifetime =273 FC)	68	45	178	110	205	-95
	88	46	168	80	185	-105
	108	46	213	105	165	-60
	128	46	219	91	145	-54
	148	47	268	120	125	-5
	168	47	279	111	105	6
	188	47	283	95	85	10
	208	47	288	80	65	15
	228	47	274	46	45	1
	248	46	268	20	25	-5
268	47	270	2	5	-3	
Engine 243 (actual lifetime =230 FC)	57	45	171	114	173	-59
	77	45	174	97	153	-56
	97	45	178	81	133	-52
	117	45	175	58	113	-55
	137	45	181	44	93	-49
	157	45	190	33	73	-40
	177	46	206	29	53	-24
	197	46	217	20	33	-13
	217	48	21	0	13	-13
Engine 244 (actual lifetime =128 FC)	32	44	171	139	96	43
	52	44	171	119	76	43
	72	43	164	92	56	36
	92	42	155	63	36	27
	112	41	126	14	16	-2

Table 2.8: Optimal aircraft engine maintenance times for engines 245 to 250 from the FD002 training data set

Engine ID [-]	RUL prediction time [FC]	Optimal state [-]	Optimal MDP time (t*) [FC]	Predicted TTM [FC]	Perfect TTM [FC]	Error [FC]
Engine 245 (actual lifetime =253 FC)	63	45	173	110	190	-80
	83	46	172	89	170	-81
	103	45	173	70	150	-80
	123	45	174	51	130	-79
	143	46	207	64	110	-46
	163	46	228	65	90	-25
	183	46	240	57	70	-13
	203	46	248	45	50	-5
	223	46	233	10	30	-20
243	48	243	0	10	-10	
Engine 246 (actual lifetime =194 FC)	48	45	172	124	146	-22
	68	45	177	109	126	-17
	88	46	183	95	106	-11
	108	46	191	83	86	-3
	128	46	194	66	66	0
	148	45	182	34	46	-12
	168	45	184	16	26	-10
	188	48	188	0	6	-6
Engine 247 (actual lifetime =197 FC)	49	45	173	124	148	-24
	69	45	169	100	128	-28
	89	45	166	77	108	-31
	109	45	165	56	88	-32
	129	45	163	34	68	-34
	149	45	175	26	48	-22
	169	45	185	16	28	-12
189	50	189	0	8	-8	
Engine 248 (actual lifetime =234 FC)	58	45	174	116	176	-60
	78	45	167	89	156	-67
	98	46	188	90	136	-46
	118	46	194	76	116	-40
	138	46	198	60	96	-36
	158	46	210	52	76	-24
	178	46	220	42	56	-14
	198	46	236	38	36	2
218	46	230	12	16	-4	
Engine 249 (actual lifetime =202 FC)	50	45	183	133	152	-19
	70	45	175	105	132	-27
	90	45	191	101	112	-11
	110	46	200	90	92	-2
	130	46	196	66	72	-6
	150	45	196	46	52	-6
	170	45	198	28	32	-4
190	45	198	8	12	-4	
Engine 250 (actual lifetime =184 FC)	46	45	183	137	138	-1
	66	46	189	123	118	5
	86	46	194	108	98	10
	106	46	183	77	78	-1
	126	45	174	48	58	-10
	146	45	175	29	38	-9
166	44	179	13	18	-5	

Table 2.9: Optimal aircraft engine maintenance times for engines 251 to 256 from the FD002 training data set

Engine ID [-]	RUL prediction time [FC]	Optimal state [-]	Optimal MDP time (t*) [FC]	Predicted TTM [FC]	Perfect TTM [FC]	Error [FC]
Engine 251 (actual lifetime =266 FC)	66	46	198	132	200	-68
	86	46	220	134	180	-46
	106	46	236	130	160	-30
	126	46	240	114	140	-26
	146	46	237	91	120	-29
	166	47	278	112	100	12
	186	46	257	71	80	-9
	206	46	249	43	60	-17
	226	46	252	26	40	-14
246	46	252	6	20	-14	
Engine 252 (actual lifetime =135 FC)	33	44	187	154	102	52
	53	44	183	130	82	48
	73	44	175	102	62	40
	93	42	142	49	42	7
	113	42	131	18	22	-4
	133	47	133	0	2	-2
Engine 253 (actual lifetime =149 FC)	37	45	177	140	112	28
	57	45	176	119	92	27
	77	45	165	88	72	16
	97	45	159	62	52	10
	117	45	148	31	32	-1
137	44	137	0	12	-12	
Engine 254 (actual lifetime =260 FC)	65	45	181	116	195	-79
	85	45	185	100	175	-75
	105	46	189	84	155	-71
	125	46	191	66	135	-69
	145	46	211	66	115	-49
	165	46	228	63	95	-32
	185	46	249	64	75	-11
	205	46	253	48	55	-7
	225	45	240	15	35	-20
245	47	245	0	15	-15	
Engine 255 (actual lifetime =340 FC)	85	46	202	117	255	-138
	105	45	171	66	235	-169
	125	46	210	85	215	-130
	145	46	257	112	195	-83
	165	47	279	114	175	-61
	185	47	310	125	155	-30
	205	47	318	113	135	-22
	225	47	329	104	115	-11
	245	47	319	74	95	-21
	265	47	328	63	75	-12
	285	47	330	45	55	-10
	305	47	326	21	35	-14
	325	47	325	0	15	-15
Engine 256 (actual lifetime =163 FC)	40	46	192	152	123	29
	60	46	186	126	103	23
	80	45	186	106	83	23
	100	45	178	78	63	15
	120	44	160	40	43	-3
	140	43	150	10	23	-13
	160	50	160	0	3	-3

Table 2.10: Optimal aircraft engine maintenance times for engines 257 to 260 from the FD002 training data set

Engine ID [-]	RUL prediction time [FC]	Optimal state [-]	Optimal MDP time (t*) [FC]	Predicted TTM [FC]	Perfect TTM [FC]	Error [FC]
Engine 257 (actual lifetime =309 FC)	77	46	192	115	232	-117
	97	45	193	96	212	-116
	117	47	237	120	192	-72
	137	46	232	95	172	-77
	157	47	289	132	152	-20
	177	47	307	130	132	-2
	197	47	301	104	112	-8
	217	47	312	95	92	3
	237	47	317	80	72	8
	257	47	306	49	52	-3
277	46	301	24	32	-8	
297	46	302	5	12	-7	
Engine 258 (actual lifetime =143 FC)	35	46	175	140	108	32
	55	46	174	119	88	31
	75	45	171	96	68	28
	95	46	167	72	48	24
	115	45	151	36	28	8
135	45	135	0	8	-8	
Engine 259 (actual lifetime =205 FC)	51	46	183	132	154	-22
	71	45	188	117	134	-17
	91	45	169	78	114	-36
	111	46	174	63	94	-31
	131	46	203	72	74	-2
	151	46	199	48	54	-6
171	45	199	28	34	-6	
191	46	203	12	14	-2	
Engine 260 (actual lifetime =316 FC)	79	44	190	111	237	-126
	99	45	207	108	217	-109
	119	46	254	135	197	-62
	139	46	256	117	177	-60
	159	46	298	139	157	-18
	179	46	300	121	137	-16
	199	46	321	122	117	5
	219	46	320	101	97	4
	239	46	324	85	77	8
	259	46	336	77	57	20
279	46	329	50	37	13	
299	46	311	12	17	-5	

3

Additional results: full case study results

In this chapter, the results of all engines of the case study as outlined in [Part I](#) are provided for the RUL maintenance strategy, see [Table 3.1](#).

Table 3.1: Optimised maintenance schedule for a pool of aircraft engines for 100 days.

Engine ID [-]	Planned date [day]	Failed engine [-]	Initial age engine [FC]	Optimal date [day]	Wasted life [FC]	AOG costs [MU]	Corrective m.costs [MU]	Preventive m. costs [MU]	Wasted life costs [MU]	Total costs [MU]
Engine_213	15	No	174	16	2	0	0	-50	-10	-60
Engine_256	25	No	117	33	12	0	0	-50	-60	-110
Engine_248	29	No	184	36	10	0	0	-50	-50	-100
Engine_231	37	No	82	44	11	0	0	-50	-55	-105
Engine_235	38	No	117	47	13	0	0	-50	-65	-115
Engine_233	42	No	124	46	7	0	0	-50	-35	-85
Engine_229	43	No	113	45	3	0	0	-50	-15	-65
Engine_203	51	No	177	65	21	0	0	-50	-105	-155
Engine_245	73	No	131	86	19	0	0	-50	-95	-145
Engine_212	77	No	34	81	6	0	0	-50	-30	-80
Engine_237	83	No	86	87	6	0	0	-50	-30	-80
Engine_205	90	No	59	98	12	0	0	-50	-60	-110
Engine_225	91	No	24	93	3	0	0	-50	-15	-65
Engine_219	99	No	15	99	1	0	0	-50	-5	-55
Engine_240	100	No	111	109	14	0	0	-50	-70	-120
Engine_224	113	No	8	117	6	0	0	-50	-30	-80
Engine_220	124	No	36	139	22	0	0	-50	-110	-160
Engine_216	127	No	33	138	16	0	0	-50	-80	-130
Engine_259	128	No	15	134	9	0	0	-50	-45	-95
Engine_257	145	No	87	156	16	0	0	-50	-80	-130
Engine_221	155	No	0	162	11	0	0	-50	-55	-105
Engine_246	157	No	0	166	13	0	0	-50	-65	-115
Engine_211	171	No	0	175	7	0	0	-50	-35	-85
Engine_244	187	No	0	190	5	0	0	-50	-25	-75
Engine_251	190	No	0	201	17	0	0	-50	-85	-135
Engine_204	191	No	0	205	20	0	0	-50	-100	-150
Engine_227	198	No	0	207	13	0	0	-50	-65	-115
Engine_254	212	No	0	226	20	0	0	-50	-100	-150
Engine_250	217	No	0	220	5	0	0	-50	-25	-75
Engine_228	220	No	0	226	9	0	0	-50	-45	-95
Engine_249	222	No	0	225	5	0	0	-50	-25	-75
Engine_223	223	No	0	226	5	0	0	-50	-25	-75
Engine_222	232	No	0	236	6	0	0	-50	-30	-80
Engine_209	246	No	0	254	12	0	0	-50	-60	-110
Engine_230	250	No	0	254	7	0	0	-50	-35	-85
Engine_208	253	No	0	260	11	0	0	-50	-55	-105
Engine_210	269	No	0	274	8	0	0	-50	-40	-90
Engine_234	275	No	0	283	12	0	0	-50	-60	-110
Engine_255	276	No	0	290	20	0	0	-50	-100	-150
Engine_242	277	No	0	281	7	0	0	-50	-35	-85

4

Verification and Validation

This additional chapter discusses the verification and validation of the model. Verification and validation of a model is an important step, as it provides a check to see if the model is meeting its requirements and if the model fulfils its intended purpose. Sargent [47] develops a methodology for model verification and validation, from which certain aspects of this V&V are derived such as predictive validation, sensitivity analysis of parameters and comparison to other models. In [section 4.1](#), the verification strategy is outlined and in [section 4.2](#) the validation strategy is outlined.

4.1. Model verification

Model verification ensures that the developed computer model is correct and that the model specifications are adhered to. Several steps can be taken to verify the model. In this section, model verification will be performed by applying model function verification. This entails that the correctness of functions and their computations is performed. This will be done for the prognostic model, optimisation model and the case study. All models are developed in Python 3.7.

Prognostic model verification

The prognostic model consists of two main functions, the health indicator model function and the polynomial chaos expansion model function. The health indicator model is verified by comparing the outcome with [24], who develop a similar health indicator. For the PCE model, a simple verification function is developed and tested, see [Equation 4.1](#).

$$y = U_1 \cdot t^4 - U_2 \cdot t^3 \quad (4.1)$$

Here, U_1 and U_2 are random uniform variables from a $U(0,1)$ distribution. Because this function uses a simple uniform distribution, the result of the PCE model should estimate the function $y = 0.5t^4 - 0.5t^3$, as the mean of a $U(0,1)$ distribution is equal to 0.5. This is indeed the case and the expansion performs as desired. Furthermore, all smaller functions which are used for the prognostic model are subject to unit testing. This ensures that each function performs the intended use of the function.

Optimisation model verification

The optimisation model consists of two main functions as well. The first one being the component level optimisation function and the second the system level optimisation function. The system level optimisation function is used for the case study and will be addressed during the case study verification.

The Markov Decision Process model is also subject to unit testing. This model can not be substituted by a simpler model, but as there are only two actions, it is relatively simple to obtain information from the value function and the corresponding state values. As such, some engines are randomly chosen and the MDP is paused every few iterations to see at which state the optimal action is to do maintenance and if this corresponds with the reward given at that state. Furthermore, the input of this model is the output of the prognostic model and the output is an optimal maintenance date. It is verified whether the optimal maintenance

date is not later than the predicted time of failure from the prognostic model. Also, the probability of failure of the final optimal time of maintenance is checked and checked if it is consistent with the PDF of the RUL prognostic model. Furthermore, all smaller functions used for this model are also unit tested to ensure that the computations are performed correctly.

Case study verification

The case study combines the prognostic model, the MDP model, the linear programming model and the rolling horizon approach. The prognostic model and MDP model have been verified at this point. The LP model is verified by performing the case study and identifying the engines which make use of the LP problem (only engines which have a clash due to hangar availability use this model). These engines are all checked and the probabilities of failure at future times, the driving factor of the LP problem, are checked. With this information, a manual calculation is performed to see if the model chooses the correct option. The model uses the Gurobi optimiser for this, and it is assumed that the commercial solver is verified and validated as well.

Finally, the rolling horizon simulation approach for the general case study is verified. As explained, every 14 days a new RUL prognostic is obtained and the schedule is fixed for 21 days. At various moments during the case study simulation it is checked whether engine maintenance is correctly planned, maintenance is being performed at the correct times and in case of failure, the engine is not operational and should be maintained at the next possible time slot at extra costs. The other three, simpler maintenance strategies are verified like this as well and will be used to validate the RUL maintenance model in the next section.

4.2. Model validation

Model validation ensures that the model is showing results with a satisfactory level of accuracy for the intended use of the model application. It ensures that the model results are realistic and are not an underestimation or overestimation of real world processes [47]. In this section the model validation will be split in validation of the results of the prognostic model and validation of the results of the case study, which integrates the prognostic model and the maintenance optimisation model.

Prognostic model validation

The results of the prognostic model can be validated if the FD001 or FD002 data set is used, because the actual failure times have been provided by creators of the data set. Therefore, the results can straightforwardly be validated. This has been done in the scientific paper in [Part I](#) as well. It is expected that the distribution of errors are centred around 0 or slightly less than 0, because RUL underestimations are favoured over overestimations by definition of the PHM scoring function and the error definition. The distribution can be found in [Figure 4.1](#) and it can be seen that this is indeed the case.

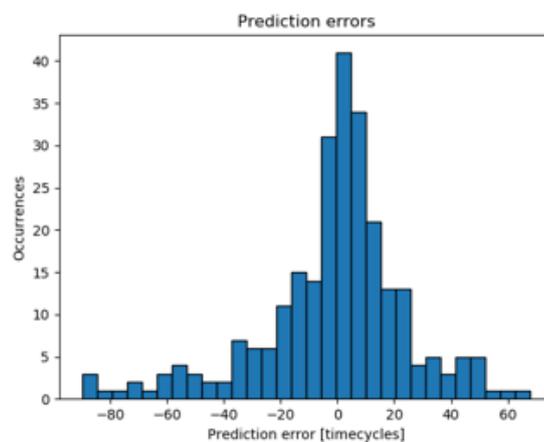


Figure 4.1: RUL prediction errors for the FD002 data set.

Furthermore, in the research paper in [Part I](#) a benchmark with results found in literature has been performed

to assess the accuracy and resulting applicability of the model. It appeared that the model is performing in the mid-range and the results have an acceptable level of accuracy.

Case study validation

The case study can be validated by comparing the results with results found in literature using different approaches. Nguyen and Medjaher [35] and Zhang and Zhang [62] both use the CMAPSS data set as well for RUL prognostics and develop a predictive maintenance framework as well. Both papers use an AI approach for RUL prognostics and decide when to do maintenance based on a set of rules. The data and results of these papers are used to validate the developed RUL maintenance model. The sensitivity analysis of cost parameters by Nguyen and Medjaher can be found in Figure 4.2.

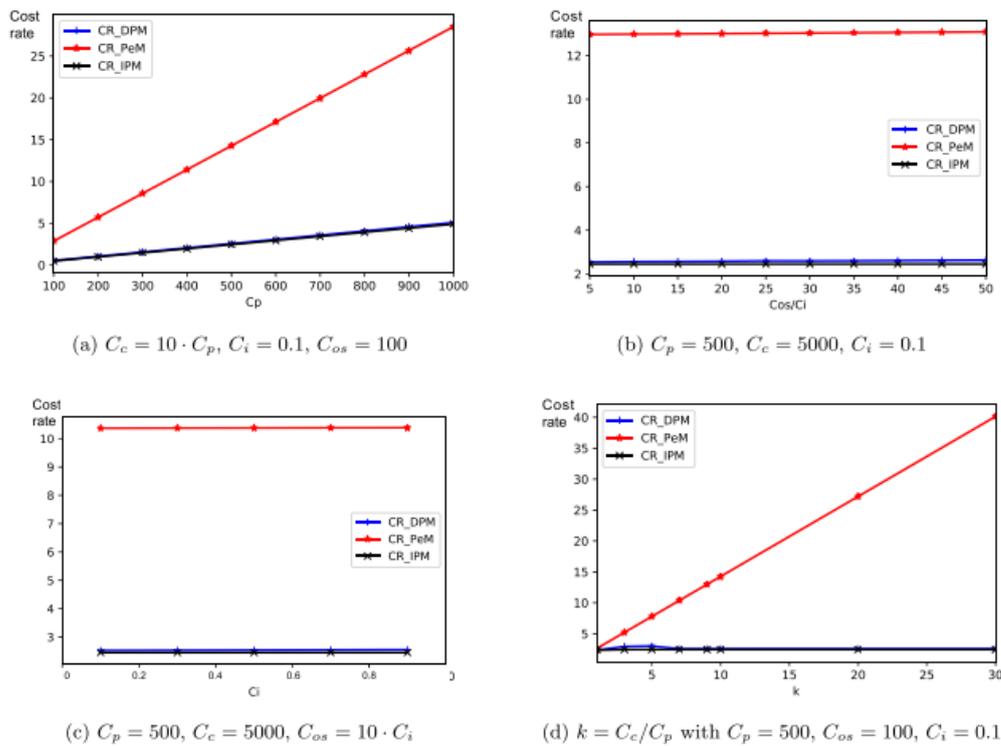


Figure 4.2: Sensitivity analysis results by Nguyen and Medjaher [35]. CR is short for cost rate, DPM for dynamic predictive maintenance, PeM for periodic maintenance and IPM ideally predicted maintenance.

The cost parameters used for this research are derived from these papers. The costs cannot be compared directly because different case study approaches are developed taking into account different aspects. However, the influence of changing the cost parameters on the cost rate in Figure 4.2 can be compared with the sensitivity analysis results found in Part I. When looking at changing the cost of preventive maintenance (Figure 4.2 a), it can be concluded that the behaviour is similar to the behaviour of increasing the cost of preventive maintenance for the research in this thesis, meaning that the relation is linear and the rate of increase is similar. Also, the difference in costs between the different maintenance strategies is in the same order of magnitude. A mayor difference is that the approach by Nguyen and Medjaher does not include costs for wasting useful engine lifetime and thus, the cost rate of predictive maintenance and perfect maintenance is almost equal. Similar conclusions are drawn from the comparison with Zhang and Zhang, who obtain the same linear relations for their sensitivity analysis, see Figure 4.3 [62].

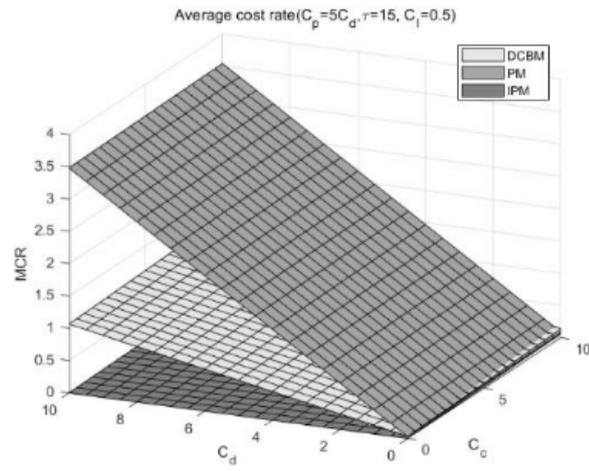


Figure 4.3: Sensitivity analysis results by Zhang and Zhang [62] by varying the cost of corrective repair C_c and cost of downtime C_d . DCBM is short for dynamic condition based maintenance, PM for periodic maintenance and IPM for ideally predicted maintenance.

Bibliography

- [1] D. An, J. H. Choi, and N. H. Kim. Prognostics 101: A tutorial for particle filter-based prognostics algorithm using Matlab. *Reliability Engineering and System Safety*, 115:161–169, 2013.
- [2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.
- [3] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Phil. Trans. of the Royal Soc. of London*, 53:370–418, 1763.
- [4] J. Cai, X. Li, and X. Chen. Optimization of aeroengine shop visit decisions based on remaining useful life and stochastic repair time. *Mathematical Problems in Engineering*, 2016:1–11, 01 2016.
- [5] E. Casado, M. L. Civita, M. Vilaplana, and E. W. McGookin. Quantification of aircraft trajectory prediction uncertainty using polynomial chaos expansions. *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, 2017-Sept(699274), 2017.
- [6] J. B. Coble and J. W. Hines. Prognostic algorithm categorization with PHM challenge application. *2008 International Conference on Prognostics and Health Management, PHM 2008*, 2008.
- [7] P. Do Van, E. Levrat, A. Voisin, and B. Iung. Remaining useful life (RUL) based maintenance decision making for deteriorating systems. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 45(31):66–72, 2012.
- [8] P. L. Duong and N. Raghavan. Uncertainty quantification in prognostics: A data driven polynomial chaos approach. *2017 IEEE International Conference on Prognostics and Health Management, ICPHM 2017*, pages 135–142, 2017.
- [9] X. Fang, K. Paynabar, and N. Gebraeel. Multistream sensor fusion-based prognostics model for systems with single failure modes. *Reliability Engineering and System Safety*, 159(March 2016):322–331, 2017.
- [10] Q. Feng, Y. Chen, B. Sun, and S. Li. An optimization method for condition based maintenance of aircraft fleet considering prognostics uncertainty. *The Scientific World Journal*, 2014, 2014.
- [11] M. Gerdes, D. Scholz, and D. Galar. Effects of condition-based maintenance on costs caused by unscheduled maintenance of aircraft. *Journal of Quality in Maintenance Engineering*, 22:394–417, 10 2016.
- [12] A. Giantomassi, F. Ferracuti, A. Benini, S. Longhi, G. Ippoliti, and A. Petrucci. Hidden markov model for health estimation and prognosis of turbofan engines. volume 3, pages 1–6, 01 2011.
- [13] N. Gordon, D. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. 1993.
- [14] L. Hawchar, C. P. El Soueidy, and F. Schoefs. Principal component analysis and polynomial chaos expansion for time-variant reliability problems. *Reliability Engineering and System Safety*, 167(June):406–416, 2017.
- [15] J. Hessburg. *Air Carrier MRO Handbook: Maintenance, Repair, and Overhaul*. An aviation week book. McGraw-Hill, 2001. ISBN 9780071361330.
- [16] Y. W. Hu, H. C. Zhang, S. J. Liu, and H. T. Lu. Sequential Monte Carlo Method Toward Online RUL Assessment with Applications. *Chinese Journal of Mechanical Engineering*, 2018.
- [17] Y. Hu, X. Miao, J. Zhang, J. Liu, and E. Pan. Reinforcement learning-driven maintenance strategy: A novel solution for long-term aircraft maintenance decision optimization. *Computers and Industrial Engineering*, 153(December 2020):107056, 2021.

- [18] R. Jiao, K. Peng, J. Dong, and C. Zhang. Fault monitoring and remaining useful life prediction framework for multiple fault modes in prognostics. *Reliability Engineering and System Safety*, 203(June):107028, 2020.
- [19] H. Jin, F. Han, and Y. Sang. An optimal maintenance strategy for multi-state deterioration systems based on a semi-Markov decision process coupled with simulation technique. *Mechanical Systems and Signal Processing*, 139:106570, 2020.
- [20] S. jin Tang, C. qiang Yu, Y. bao Feng, J. Xie, Q. he Gao, and X. sheng Si. Remaining useful life estimation based on Wiener degradation processes with random failure threshold. *Journal of Central South University*, 23(9):2230–2241, 2016.
- [21] I. T. Jolliffe and J. Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065): 20150202, 2016.
- [22] A. Keese and H. Matthies. Sparse quadrature as an alternative to monte carlo for stochastic finite element techniques. *PAMM*, 3:493 – 494, 12 2003.
- [23] J. W. Kim, G. Choi, J. C. Suh, and J. M. Lee. Optimal scheduling of the maintenance and improvement for water main system using Markov decision process. *IFAC-PapersOnLine*, 28(8):379–384, 2015.
- [24] K. Le Son, M. Fouladirad, A. Barros, E. Levrat, and B. Iung. Remaining useful life estimation based on stochastic deterioration models: A comparative study. *Reliability Engineering and System Safety*, 112: 165–175, 2013.
- [25] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, and J. Lin. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, 104:799–834, 2018.
- [26] H. Li and D. Zhang. Probabilistic collocation method for flow in porous media: Comparisons with other stochastic methods. *Water Resources Research - WATER RESOUR RES*, 43, 09 2007.
- [27] N. Li, Y. Lei, T. Yan, N. Li, and T. Han. A wiener-process-model-based method for remaining useful life prediction considering unit-to-unit variability. *IEEE Transactions on Industrial Electronics*, 66(3):2092–2101, 2019.
- [28] N. Li, N. Gebraeel, Y. Lei, X. Fang, X. Cai, and T. Yan. Remaining useful life prediction based on a multi-sensor data fusion model. *Reliability Engineering and System Safety*, 208(September 2020):107249, 2021.
- [29] R. Li, W. J. Verhagen, and R. Curran. A comparative study of Data-driven Prognostic Approaches: Stochastic and Statistical Models. *2018 IEEE International Conference on Prognostics and Health Management, ICPHM 2018*, (June):1–8, 2018.
- [30] T. Li, S. Wang, J. Shi, and Z. Ma. An adaptive-order particle filter for remaining useful life prediction of aviation piston pumps. *Chinese Journal of Aeronautics*, 31(5):941–948, 2018.
- [31] Z. Li and S. Zhong. An aero-engine life-cycle maintenance policy optimization algorithm: Reinforcement learning approach. *Chinese Journal of Aeronautics*, 32(9):2133–2150, 2019.
- [32] B. Liu, P. Do, B. Iung, and M. Xie. Stochastic Filtering Approach for Condition-Based Maintenance Considering Sensor Degradation. *IEEE Transactions on Automation Science and Engineering*, 17(1):177–190, 2020.
- [33] K. Medjaher, N. Zerhouni, and J. Baklouti. Data-driven prognostics based on health indicator construction: Application to PRONOSTIA's data. *2013 European Control Conference, ECC 2013*, (July):1451–1456, 2013.
- [34] J. B. Nagel, J. Rieckermann, and B. Sudret. Principal component analysis and sparse polynomial chaos expansions for global sensitivity analysis and model calibration : Application to urban drainage simulation. *Reliability Engineering and System Safety*, 195(January 2019):106737, 2020.
- [35] K. T. Nguyen and K. Medjaher. A new dynamic predictive maintenance framework using deep learning for failure prognostics. *Reliability Engineering and System Safety*, 188(March):251–262, 2019.

- [36] A. O'Hagan. Polynomial Chaos: A Tutorial and Critique from a Statistician's Perspective. *Tonyohagan.Co.Uk*, pages 1–16, 2011.
- [37] S. Oladyshkin and W. Nowak. Data-driven uncertainty quantification using the arbitrary polynomial chaos expansion. *Reliability Engineering and System Safety*, 106:179–190, 2012.
- [38] S. Oladyshkin, H. Class, and W. Nowak. Bayesian updating via bootstrap filtering combined with data-driven polynomial chaos expansions: Methodology and application to history matching for carbon dioxide storage in geological formations. *Computational Geosciences*, 17(4):671–687, 2013.
- [39] M. Orchard, B. Wu, and G. Vachtsevanos. A particle filtering framework for failure prognosis. *Proceedings of the World Tribology Congress III - 2005*, (May 2015):883–884, 2005.
- [40] P. Paris and F. Erdogan. A critical analysis of crack propagation laws. *Journal of Basic Engineering*, 85:528–533, 1963.
- [41] L. Peel. Data Driven Prognostics using a Kalman Filter Ensemble of Neural Network Models. *2008 IEEE International Conference on Prognostics and Health Management*, 2008.
- [42] M. Pitt and N. Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94, 11 1997.
- [43] A. Prabhakar, J. Fisher, and R. Bhattacharya. Polynomial chaos-based analysis of probabilistic uncertainty in hypersonic flight dynamics. *Journal of Guidance, Control, and Dynamics*, 33(1):222–234, 2010.
- [44] M. L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.
- [45] A. Rajabi-Ghahnavie and M. Fotuhi-Firuzabad. Application of Markov decision process in generating units maintenance scheduling. *2006 9th International Conference on Probabilistic Methods Applied to Power Systems, PMAPS*, pages 7–12, 2006.
- [46] B. Saha and K. Goebel. Model Adaptation for Prognostics in a Particle Filtering Framework. pages 1–10, 2011.
- [47] R. Sargent. Verification and validation of simulation models. volume 37, pages 166 – 183, 01 2011. doi: 10.1109/WSC.2010.5679166.
- [48] A. Saxena and K. Goebel. *PHM08 Challenge Data Set*. NASA Ames Research Center, Moffett Field, CA, 2008.
- [49] H. Shi and J. Zeng. Real-time prediction of remaining useful life and preventive opportunistic maintenance strategy for multi-component systems considering stochastic dependence. *Computers and Industrial Engineering*, 93:192–204, 2016.
- [50] Y. Shi, W. Zhu, Y. Xiang, and Q. Feng. Condition-based maintenance optimization for multi-component systems subject to a system reliability requirement. *Reliability Engineering and System Safety*, 202 (November 2019):107042, 2020.
- [51] D. Siegel, W. Zhao, H. Al-Atat, J. Lee, and M. Kumar. A particle filtering approach to remaining useful life prediction of aircraft engines. *Technical Program for MFPT: The Applied Systems Health Management Conference 2011: Enabling Sustainable Systems*, 01 2011.
- [52] J. P. Sprong, X. Jiang, and H. Polinder. A deployment of prognostics to optimize aircraft maintenance - A literature review. *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, 11(1), 2019.
- [53] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. John Wiley & Sons, Ltd, 2006.
- [54] K. Verbert, B. De Schutter, and R. Babuška. Timely condition-based maintenance planning for multi-component systems. *Reliability Engineering and System Safety*, 159(January 2016):310–321, 2017.

- [55] H. C. Vu, P. Do, M. Fouladirad, and A. Grall. Dynamic opportunistic maintenance planning for multi-component redundant systems with various types of opportunities. *Reliability Engineering and System Safety*, 198(January):106854, 2020.
- [56] P. Wang and R. Gao. Particle filtering-based system degradation prediction applied to jet engines. *PHM 2014 - Proceedings of the Annual Conference of the Prognostics and Health Management Society 2014*, pages 658–663, 01 2014.
- [57] T. Wang, J. Yu, D. Siegel, and J. Lee. A Similarity-Based Prognostics Approach for Remaining Useful Life Estimation of Engineered Systems. *2019 IEEE International Conference on Prognostics and Health Management, ICPHM 2019*, 2008.
- [58] P. Wen, S. Zhao, S. Chen, and Y. Li. A generalized remaining useful life prediction method for complex systems based on composite health indicator. *Reliability Engineering and System Safety*, 205(September 2020):107241, 2021.
- [59] N. Wiener. The homogeneous chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.
- [60] Z. Wu, B. Guo, Axita, X. Tian, and L. Zhang. A Dynamic Condition-Based Maintenance Model Using Inverse Gaussian Process. *IEEE Access*, 8:104–117, 2020.
- [61] D. Xiu and G. Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24:619–644, 10 2002.
- [62] L. Zhang and J. Zhang. A Data-Driven Maintenance Framework under Imperfect Inspections for Deteriorating Systems Using Multitask Learning-Based Status Prognostics. *IEEE Access*, 9(Ddm):3616–3629, 2021.
- [63] F. Zhao, Z. Tian, and Y. Zeng. A stochastic collocation approach for efficient integrated gear health prognosis. *Mechanical Systems and Signal Processing*, 39(1-2):372–387, 2013.
- [64] E. Zio and G. Peloni. Particle filtering prognostic estimation of the remaining useful life of nonlinear components. *Reliability Engineering & System Safety*, 96(3):403–409, 2011.