

**THÈSE DE DOCTORAT DE L'ÉTABLISSEMENT UNIVERSITÉ BOURGOGNE FRANCHE-COMTÉ**

**PRÉPARÉE À L'UNIVERSITÉ DE FRANCHE-COMTÉ**

École doctorale n°37  
Sciences Pour l'Ingénieur et Microtechniques

Doctorat d'Informatique

par

**OMAR BOUGACHA**

**Contribution aux décisions post-pronostic: nouveau framework basé sur les  
interactions pronostic-décision  
Contribution to Post-Prognostic Decisions: a New Framework Based on  
Prognostic-Decision Interactions**

Thèse présentée et soutenue à Besançon, le 16 octobre 2020

Composition du Jury :

JEAN-MARC NICOD	Professeur à l'ENSMM Besançon	Président
DAMIEN TRENTESAUX	Professeur à l'Université Polytechnique Hauts-de-France	Rapporteur
LABIB TERRISSA	Professeur à l'Université Mohamed Khider Algérie	Rapporteur
SONIA HAJRI-GABOUJ	Professeur à INSAT Tunisie	Examinatrice
VALÉRIE BOTTA-GENOULAZ	Professeur à INSA Lyon	Examinatrice
CHRISTOPHE VARNIER	MCF HDR à l'ENSMM Besançon	Directeur de thèse
NOUREDDINE ZERHOUNI	Professeur à l'ENSMM Besançon	Codirecteur de thèse
PIERRE DERSIN	ALSTOM Transport	Invité
JEAN-PHILIPPE RACLOT	ALSTOM Transport	Invité



**Titre :** Contribution aux décisions post-pronostic: nouveau framework basé sur les interactions pronostic-décision

Contribution to Post-Prognostic Decisions: a New Framework Based on Prognostic-Decision Interactions

**Résumé :**

Durant la dernière décennie, de nombreux travaux ont été menés dans le domaine du PHM (Prognostics and Health Management). La majorité des résultats publiés concerne l'étape principale du pronostic. Cependant, le but du PHM est de fournir les bonnes informations aux bonnes personnes pour mieux gérer le cycle de vie du composant ou du système. Ainsi l'estimation ou la prédiction du RUL n'est pas l'objectif ultime du PHM. Mais, il est important d'utiliser cette information pour améliorer la qualité de la décision opérationnelle et pour optimiser la politique de maintenance prédictive.

D'une part, les décisions prises pour un système modifieront l'évolution de son état de santé et par conséquent son RUL. D'autre part, une nouvelle durée de vie résiduelle engendra de nouvelles décisions à prendre. Ainsi les deux processus de prise de décision et de pronostic sont interdépendants. Le but de cette thèse est de concevoir un framework de PHM qui met l'accent sur cette interaction pronostic-décision. Dans cette thèse, des modifications sur le framework classique de PHM sont proposées pour traiter certaines de ces lacunes. Les verrous scientifiques suivants sont adressés: 1) Comment mettre en valeur les interactions pronostic-décision ? 2) Avec quelle fréquence il faut exécuter le cycle PHM ? 3) Sur quelle durée il faut prendre la décision post-pronostic ? 4) Comment améliorer l'intégration des informations prédites dans la prise de décision ? 5) Comment clarifier l'intégration du framework PHM dans les applications ? 6) Comment améliorer la robustesse du framework PHM vis-à-vis des potentiels changements dans les conditions opératoires du système étudié ? Les questions qui sont abordées dans cette thèse ont conduit à développer un framework de décision post-pronostic. Les principales contributions sont les suivantes:

- Les verrous scientifiques du domaine de PHM sont analysés et récapitulés. Les travaux de décisions post-pronostic sont collectés et étudiés dans un état de l'art détaillé pour détecter des lacunes dans ce domaine.

- Le schéma classique connu dans la littérature de mise en oeuvre du PHM est revue. Une contribution consiste à proposer d'ajouter des boucles de retours d'information à plusieurs niveaux: (i) une boucle de construction de décision, (ii) une boucle d'application de décision et (iii) une boucle d'information. Ces boucles assurent l'intégration des prédictions dans la prise de décision et la prise en considération des futures décisions par le module de pronostic. Elles améliorent la précision des prédictions en éliminant les incertitudes sur les futures décisions, améliorent les performances du module de prise de décision en intégrant dynamiquement les nouvelles données présentes et garantissent que le framework PHM soit réactif aux changements de conditions du système.
- Le framework de décision post-pronostique est instancié sur deux cas d'étude; une première application sur le problème conjoint de maintenance et de production pour une machine à usage multiple, et une deuxième application qui étudie le problème conjoint de maintenance et d'affectation des missions pour une flotte de matériels roulants.
- La pertinence de mise en oeuvre de stratégies de PHM est étudié pour différentes configurations de machine et de problème de décision. La mise en oeuvre de PHM apparait clairement plus utile dans des contextes de fortes dynamiques des dégradations et des décisions.
- L'importance de l'influence de la durée de l'horizon de décision sur les performances des algorithmes de prise de décision est démontrée pour différentes dynamique de dégradation et de configuration de problème de décision. Prouvant ainsi que la durée de l'horizon de décision est un paramètre important de l'implémentation du cycle PHM et nécessite une étude plus approfondie.

*The future depends on what we do in the present*  
- M. Ghandi

# Publications

## Peer-Reviewed International Conferences:

- ESREL2020 ESREL2020 Omar Bougacha, Christophe Varnier, Nouredine Zerhouni, and Pierre Dersin. *Impact of the Decision Horizon on Railway Systems Maintenance and Service Scheduling*, 30th European Safety and Reliability Conference (ESREL), Venice, Italy, November, 2020.
- PHM2020 PHM2020 Omar Bougacha, Christophe Varnier, and Nouredine Zerhouni. *On the Use of Prognostics and Health Management to Jointly Schedule Production and Maintenance on a Single Multi-purpose Machine*, Prognostics and Health Management Conference (PHM), Besançon, France, May, 2020.
- ICASET2019 ICASET2019 Omar Bougacha, Christophe Varnier, Nouredine Zerhouni, and Sonia Hajri-Gabouj. *Integrated Production and Predictive Maintenance Planning Based on Prognostic Information*, 3rd IEEE International Conference on Advanced Systems and Emergent Technologies (ICASET), Hammamet, Tunisia, March, 2019.
- CIE48 CIE48 Omar Bougacha, Christophe Varnier, Nouredine Zerhouni, and Sonia Hajri-Gabouj. *A Post-Prognostic Decision Approach for Production and Maintenance*, 48th Conference on Computers and Industrial Engineering (CIE48), Auckland, New Zealand, December 2-5, 2018.

## Journals:

- ISA2020 ISA2020 Omar Bougacha, Christophe Varnier, and Nouredine Zerhouni. *Impact of Decision Horizon on Post-Prognostics Maintenance and Missions Scheduling: A Railways Case Study*, ISA Transactions: Special Issue on Prognostics and Health Management, 2020 (In Review).
- IJPHM2020 IJPHM2020 Omar Bougacha, Christophe Varnier, and Nouredine Zerhouni. *A Review of Post-Prognostics Decision-Making in Prognostics and Health Management*, International Journal on Prognostics and Health Management (IJPHM), (In Review).
- IJPHM2020 IJPHM2020 Omar Bougacha, Christophe Varnier, and Nouredine Zerhouni. *Enhancing Decisions in Prognostics and Health Management Framework*, International Journal on Prognostics and Health Management (IJPHM), vol 11(1), 2020.



# Contents

<b>Publications</b>	<b>v</b>
<b>Glossaire of Notations</b>	<b>xiii</b>
<b>Introduction</b>	<b>1</b>
<b>1 Post-prognostics decision-making</b>	<b>5</b>
1.1 Prognostics and health management . . . . .	6
1.1.1 Overview . . . . .	6
1.1.2 Challenges and issues . . . . .	9
1.1.2.1 Observation related issues . . . . .	9
1.1.2.2 Analysis related issues . . . . .	11
1.1.2.3 Decision support related issues . . . . .	14
1.1.3 Synthesis . . . . .	15
1.2 Review of post-prognostic decisions . . . . .	16
1.2.1 Definition . . . . .	16
1.2.2 Post-prognostic maintenance . . . . .	18
1.2.2.1 Single machine maintenance planning . . . . .	18
1.2.2.2 Multiple machines maintenance planning . . . . .	22
1.2.3 Operational decisions . . . . .	24
1.2.3.1 Production and mission planning . . . . .	24
1.2.3.2 Logistics planning . . . . .	25
1.2.3.3 Automatic control . . . . .	26
1.2.4 Mixed decisions . . . . .	27
1.2.4.1 Mission or production planning jointly with maintenance . .	27
1.2.4.2 Automatic control jointly with maintenance . . . . .	28
1.2.4.3 Logistics jointly with maintenance . . . . .	29
1.2.5 Synthesis . . . . .	30

1.2.5.1	System health state and degradation models . . . . .	31
1.2.5.2	The integration of prognostic information in decision-making . . . . .	31
1.2.5.3	Recurrence of RUL estimation . . . . .	32
1.2.5.4	Objective and resolution methods . . . . .	33
1.3	Open issues and problem statement for the thesis . . . . .	33
1.3.1	Decision-prognostics interactions . . . . .	34
1.3.2	Toward decision-enhanced PHM . . . . .	36
1.3.2.1	Issues to be addressed . . . . .	36
1.3.2.2	Assumptions . . . . .	37
1.3.2.3	Objective and contributions . . . . .	37
1.4	Summary . . . . .	37
<b>2</b>	<b>Decision Enhanced PHM Framework</b>	<b>39</b>
2.1	Systems and applications in PHM context . . . . .	40
2.1.1	Systems . . . . .	40
2.1.2	Applications . . . . .	43
2.2	Main Concepts . . . . .	44
2.2.1	Elementary actions . . . . .	44
2.2.2	Local decisions . . . . .	47
2.2.3	Global decisions . . . . .	50
2.2.4	Estimators and Decisions Builders . . . . .	52
2.3	PHM framework adaptation . . . . .	54
2.3.1	Decision building loop . . . . .	55
2.3.2	Decision applying loop . . . . .	57
2.3.3	Information/data loop . . . . .	58
2.3.4	Overview of the Proposed Framework . . . . .	60
2.4	Metrics and parameters . . . . .	61
2.5	Summary . . . . .	61
<b>3</b>	<b>Manufacturing Study Case</b>	<b>63</b>
3.1	Decision problem definition . . . . .	65
3.1.1	Related works and motivation . . . . .	65
3.1.2	Scope of the study . . . . .	67
3.1.2.1	Hypothesis . . . . .	68
3.1.2.2	Constraints . . . . .	69
3.1.3	System description . . . . .	69



3.1.3.1	Machine structure . . . . .	69
3.1.3.2	Degradation model . . . . .	69
3.1.4	Problem description . . . . .	72
3.1.4.1	Production Scheduling Problem . . . . .	73
3.1.4.2	Maintenance Problem . . . . .	74
3.1.4.3	Joint Problem . . . . .	74
3.2	Initialization of the proposed framework . . . . .	75
3.2.1	System definition . . . . .	76
3.2.2	Application definition . . . . .	77
3.2.3	Elementary Actions . . . . .	79
3.2.4	Local Decisions . . . . .	80
3.2.5	Global Decisions . . . . .	81
3.2.6	Loops settings . . . . .	82
3.3	Implementation . . . . .	83
3.3.1	Sequencing problem . . . . .	83
3.3.2	Ant Colony Optimization . . . . .	84
3.3.3	ACO's prognostics-enhancement . . . . .	85
3.4	Numeric application . . . . .	87
3.4.1	ACO parameters' tuning . . . . .	87
3.4.1.1	ACO Parameters . . . . .	88
3.4.1.2	Decision Horizon . . . . .	90
3.4.1.3	ACO Convergence . . . . .	92
3.4.2	Adapting conventional CBM methods . . . . .	92
3.4.2.1	Moore with condition-based maintenance . . . . .	93
3.4.2.2	Genetic algorithm with condition-based maintenance . . . . .	94
3.4.3	PHM+ vs. CBM . . . . .	95
3.4.4	Investigating the use of PHM+ . . . . .	99
3.5	Discussion . . . . .	102
3.5.1	Limits . . . . .	102
3.5.2	Results Summary . . . . .	103
3.6	Summary . . . . .	104
<b>4</b>	<b>Rail Transportation System Study Case</b>	<b>105</b>
4.1	Decision problem definition . . . . .	107
4.1.1	Related works and Motivation . . . . .	107

4.1.1.1	Related works . . . . .	107
4.1.1.2	Synthesis . . . . .	109
4.1.1.3	Motivation . . . . .	109
4.1.1.4	Contributions . . . . .	110
4.1.2	Scope of the study . . . . .	110
4.1.2.1	Hypothesis . . . . .	111
4.1.2.2	Constraints . . . . .	111
4.1.3	System description . . . . .	112
4.1.3.1	Rolling stock fleet structure . . . . .	112
4.1.3.2	Degradation model . . . . .	112
4.1.4	Problem description . . . . .	113
4.1.4.1	Vehicle scheduling problem . . . . .	114
4.1.4.2	Maintenance problem . . . . .	116
4.1.4.3	Joint Problem . . . . .	117
4.1.5	Decision Horizon Study . . . . .	117
4.2	Instantiation of the proposed framework . . . . .	118
4.2.1	System definition . . . . .	118
4.2.2	Application definition . . . . .	120
4.2.3	Elementary Actions . . . . .	122
4.2.4	Local Decisions . . . . .	123
4.2.5	Global Decisions . . . . .	124
4.2.6	Loops settings . . . . .	125
4.3	Implementation of the decision module . . . . .	127
4.3.1	Genetic Algorithm . . . . .	127
4.3.1.1	Solution Representation . . . . .	128
4.3.1.2	Initial Generation . . . . .	129
4.3.1.3	Genetic Operators Results . . . . .	129
4.3.1.4	Mutation . . . . .	130
4.3.1.5	Crossover . . . . .	131
4.3.1.6	Fitness Evaluation . . . . .	131
4.3.1.7	Construction of New Generation . . . . .	132
4.3.1.8	Overall Algorithm . . . . .	132
4.3.2	Heuristics . . . . .	132
4.3.2.1	H1: . . . . .	133
4.3.2.2	H2: . . . . .	135

4.4	Numeric application . . . . .	137
4.4.1	Case study . . . . .	137
4.4.2	Methods Comparison . . . . .	139
4.4.3	Decision Horizon Duration Study . . . . .	142
4.4.3.1	Genetic Algorithm . . . . .	143
4.4.3.2	Heuristic H2_v1 and H2_v2 . . . . .	144
4.4.3.3	Synthesis . . . . .	145
4.4.4	The effects of the problem's size on the decision horizon . . . . .	146
4.4.5	The effects of the trains' configuration on the decision horizon . . . . .	146
4.5	Discussion . . . . .	148
4.5.1	Limits . . . . .	148
4.5.2	Results Summary . . . . .	150
4.6	Summary . . . . .	151
<b>Conclusion &amp; future works</b>		<b>153</b>
<b>A Post-Prognostic Decisions Summary Tables</b>		<b>175</b>
A.1	Maintenance Decisions . . . . .	175
A.1.1	Single Machine . . . . .	175
A.1.2	Multiple Machines . . . . .	175
A.2	Operational Decisions . . . . .	180
A.2.1	Production and Mission Scheduling . . . . .	180
A.2.2	Logistics Planning . . . . .	180
A.2.3	Automatic Control . . . . .	187
A.3	Mixed Decisions . . . . .	187
A.3.1	Mission or Production Planning Jointly with Maintenance . . . . .	187
A.3.2	Logistics Jointly with Maintenance . . . . .	187
<b>B Manufacturing Study Case Results Tables</b>		<b>193</b>
B.1	Single Speed . . . . .	193
B.2	Multiple Speeds . . . . .	194



# Glossary of Notations

## Notations

<b>Chapter 2:</b>	
$\alpha_i$	Elementary action i
$d_{i,j}$	Descriptor j of elementary action i
$a_j$	Attribute j
$v_{i,j}$	Value of elementary action i for attribute j
$\mathcal{A}$	Set of elementary action for a given application
$e_i$	Local decision i
$d_{i,j}$	Descriptor j of local decision i
$a_j$	Attribute j
$v_{i,j}$	Value of local decision i for attribute j
$\mathcal{E}$	Set of local decisions
$\mathcal{F}(t)$	Set of feasible local decisions at instant t
$S(t)$	Set of possible sequences of local decisions
$s^*$	Selected sequence of $S(t)$ .
$g_u(t)$	Global decision $u$ at instant t
$D_{u,j}$	Descriptor j of global decision u
$A_j$	Global attribute j
$V_{u,j}$	Value of global decision $u$ for global attribute j
$\mathcal{G}(t)$	Set of global decisions at instant t
<b>Chapter 3:</b>	
$\Pi$	Number of product types
$\pi$	Product type $\pi$
$J$	Number of operating profiles
$j$	Operating profile j
$K$	Number of predictive components
$k$	Index of component k
$H_k$	Health indicator (degradation) of component k
$a_k, b_k$	Degradation parameter of component k
$\Delta_k$	Failure threshold of component k
$SH$	Duration of simulation horizon
$DH$	Duration of decision horizon
$I$	Number of periods in a decision horizon
$i$	Index of period
$\Delta T$	Duration of a period

$Sch_i$	Schedule of period $i$
$C_i$	Cost of period $i$
$G_i$	Estimated gain of period $i$
$Maint_i$	Time spent in maintenance during period $i$
$Prod_i$	Time spent in production during period $i$
$Idl_i$	Time spent while machine is idle during period $i$
$\mathcal{P}(i)$	Set of production orders of period $i$
$p$	Index of order
$O_{i,p}$	Production order $p$ of period $i$
$\pi_p$	Product type of order $p$
$Q_p$	Quantity to produce in order $p$
$r_p$	Release date of order $p$
$d_p$	Due date of order $p$
$D_p$	Deadline of order $p$
$p_{p,j}$	Processing time of order $p$ with profile $j$
$c_{p,j}$	Completion time of order $p$ with profile $j$
$Cp(p, j)$	Cost of production of order $p$ with profile $j$
$Gp(p)$	Estimated gain of order $p$
$U_p$	Lateness indicator of order $p$
$cost_{\pi_p,j}$	Cost of producing one product of $\pi_p$ with profile $j$
$\mathcal{PM}$	Set of components to replace
$M_k$	Cost of replacing component $k$
$CM_k$	Failure indicator of component $k$
$P_{CM}$	Penalty on corrective maintenance
$C_m$	Cost of maintaining $\mathcal{PM}$

---

**Chapter 4:**

$\mathcal{M}$	Set of rolling units
$M$	Number of rolling units
$m$	Index of rolling unit
$K$	Number of predictive components
$k$	Index of predictive component
$H_{m,k}$	Degradation level of component $k$
$\mu'_k$	Scale parameter of component's $k$ degradation model
$\alpha_k$	Shape parameter of component's $k$ degradation model
$\Delta_k$	Failure threshold of component $k$
$\Lambda_k$	Minimum level of component's $k$ health indicator to consider maintenance scheduling
$\hat{H}_{m,k}$	Estimated (virtual) degradation level of component $k$
$L$	Number of preventive component
$l$	Index of preventive component
$\theta_{m,l}$	Mileage traveled by component $l$ of unit $m$ since its last maintenance operation
$\Theta_l$	Mileage threshold for component's $l$ end-of-life
$\Phi_l$	Mileage minimum level for component's $l$ to consider maintenance scheduling
$\hat{\theta}_{m,l}$	Estimated (virtual) traveled mileage of component $l$
$SH$	Duration of simulation horizon
$DH$	Duration of decision horizon

$\Delta T$	Duration of a period
$I$	Number of Periods in a Decision Horizon
$i$	Index of period
$\mathcal{P}$	Set of planned tasks
$P$	Number of planned tasks
$p$	Index of planned task
$d_p$	Computed distance of mission $p$
$s_p$	Severity of mission/task $p$
$\delta_{p,k}$	Estimated degradation rate of component $k$ during mission $p$
$Sch$	Schedule of global decision
$Sch_i$	Schedule of period $i$
$\pi_m(i)$	Indicator if train $m$ is at rest during period $i$
$\beta_{p,m}(i)$	Indicator of assigning mission $p$ to train $m$ during period $i$
$\omega_m(i)$	Indicator if train $m$ undergoes maintenance during period $i$
$\sigma_{m,k}(i)$	Indicator if component $k$ of train $m$ is scheduled for maintenance during period $i$
$\sigma_{m,l}(i)$	Indicator if component $l$ of train $m$ is scheduled for maintenance during period $i$
$f_m(i)$	Indicator if train $m$ fails during period $i$
$C_i$	Cost of period $i$
$C_{lost}$	Cost of missing a mission
$CR_k$	Replacement cost of predictive component $k$
$CR_l$	Replacement cost of preventive component $l$
$LP_x$	Penalty cost on the lost mileage of component $x$
$C_{m,x}(i)$	Cost of maintenance of component $x$ during period $i$
$C_{cor}$	Cost of corrective maintenance
$Total\_Cost(DH)$	Total cost during a decision horizon

## Acronyms

<b>PHM</b>	Prognostics and Health Management.
<b>CBM</b>	Condition-Based Maintenance.
<b>RUL</b>	Remaining Useful Life.
<b>HI</b>	Health Indicator.
<b>PPDM</b>	Post-Prognostics Decision-Making.
<b>DM</b>	Decision-Making.
<b>GA</b>	Genetic Algorithm.





# List of Figures

1.1	Prognostics and Health Management 7-layers architecture . . . . .	8
1.2	Sources of Uncertainty . . . . .	13
1.3	Synthesis of PHM Challenges Grouped by Phase . . . . .	15
1.4	PHM place related to the system process . . . . .	30
1.5	The interactions between the prognostics and the decision-making modules	35
2.1	Block diagram of the decision building process . . . . .	53
2.2	Example of the estimator output . . . . .	54
2.3	Time Feasibility of Local Decisions . . . . .	54
2.4	Health Feasibility of Local Decisions . . . . .	55
2.5	The Decision Building Sequential Diagram . . . . .	56
2.6	The Decision Applying Loop . . . . .	57
2.7	Illustration of the results of the decision applying loop . . . . .	58
2.8	Information Loop . . . . .	59
2.9	Degradation Evolution of Component Number 1 . . . . .	59
2.10	Evolution of RMSE of Component Number 1 . . . . .	60
2.11	The Closed Loops PHM Process . . . . .	61
3.1	Example of regressions for an iron with cut depth of 0.75 and a feed of 0.25 based on the dataset of Goebel and Agogino [66] . . . . .	71
3.2	Exponential models for iron products . . . . .	71
3.3	Exponential models for steel products . . . . .	72
3.4	Illustration of Cumulative Benefit . . . . .	75
3.5	Implemented Loops . . . . .	83
3.6	Integrating Decision Building Loop in ACO . . . . .	85
3.7	ACO's Decision Building Sequential Diagram . . . . .	87
3.8	Execution Time = $f(\text{Number of Ants}, \text{Number of Cycles})$ . . . . .	89
3.9	Benefits $B_i = f(\text{Number of Ants}, \text{Number of Cycles})$ . . . . .	90

3.10 Average Execution Time = $f(\text{Duration of Decision Horizon})$ . . . . .	91
3.11 Average Cumulative Benefit = $f(\text{Duration of Decision Horizon})$ . . . . .	91
3.12 Average Execution Time = $f(\text{Duration of Decision Horizon})$ . . . . .	92
3.13 Evolution of the Benefits of the Best Schedule . . . . .	93
3.14 ACO Convergence Validation . . . . .	93
3.15 Benefits of Using PHM+ vs CBM . . . . .	101
3.16 Use map of PHM+ vs CBM for a single speed test case . . . . .	102
4.1 Health State Illustration. . . . .	113
4.2 Illustration of the Cumulative Total Cost. . . . .	118
4.3 An Example of Global Decision Schedule Structure. . . . .	124
4.4 Proposed Framework Implementation. . . . .	126
4.5 An Example of Solution Representation. . . . .	129
4.6 Example of Single-Point Order Crossover. . . . .	131
4.7 Results of the Different Methods. . . . .	141
4.8 Mean Missed Missions of the Different Methods. . . . .	141
4.9 Mean Number of Failures Over the Simulation Horizon. . . . .	142
4.10 Mean Lost Mileage of the Different Methods. . . . .	142
4.11 Execution Time of the Different Methods. . . . .	143
4.12 Cumulative cost of one-day and ten-days decision horizon. . . . .	143
4.13 Total Cost evolution for different initial conditions. . . . .	144
4.14 Box plot of total cost variation for the GA. . . . .	144
4.15 Box plot of total cost variation for heuristic H2_v1. . . . .	145
4.16 Box plot of total cost variation for heuristic H2_v2. . . . .	145
4.17 Mean Total Cost Variation Under Different Problem Sizes . . . . .	147
4.18 Mean Total Cost Variation Under Different Trains Configurations . . . . .	149

# List of Tables

1.1 Publications on Post-Prognostic Decision per Decision Type . . . . .	18
2.1 Example of a maintenance elementary action . . . . .	45
2.2 Example of a production elementary action . . . . .	45
2.3 Example of a maintenance local decision . . . . .	48
2.4 Example of a production local decision . . . . .	49
3.1 Performances of regressions . . . . .	70
3.3 Definition of a production elementary action . . . . .	80
3.4 Definition of a maintenance elementary action . . . . .	80
3.5 Structure of a maintenance local decision . . . . .	80
3.6 Structure of a production local decision . . . . .	81
3.7 Structure of a Global Decision . . . . .	81
3.8 Methods comparison . . . . .	98
3.9 Machines configurations . . . . .	100
3.10 Initial conditions configurations . . . . .	100
4.2 Definition of a mission assignment elementary action . . . . .	122
4.3 Definition of a put-to-rest elementary action . . . . .	122
4.4 Definition of a maintenance elementary action . . . . .	123
4.5 Structure of a maintenance local decision . . . . .	123
4.6 Structure of a mission assignment local decision . . . . .	123
4.7 Definition of a put-to-rest local decision . . . . .	124
4.8 Structure of a Global Decision . . . . .	125
4.9 Characteristics of the Missions. . . . .	139
4.10 Rolling Stocks Predictive Components Characteristics. . . . .	139
4.11 Rolling Stocks Preventive Components Characteristics. . . . .	140
4.12 The Values of the Numeric Application. . . . .	140

4.13 Best Decision Horizon per Method. . . . .	146
4.14 Components Configuration . . . . .	146
4.15 Configurations for Different Problem Sizes . . . . .	146
A.1 Existing papers for single machine maintenance scheduling. (Part 1) . . . .	176
A.2 Existing papers for single machine maintenance scheduling. (Part 2) . . . .	177
A.3 Existing papers for single machine maintenance scheduling. (Part 3) . . . .	178
A.4 Existing papers for single machine maintenance scheduling. (Part 4) . . . .	179
A.5 Existing papers for multi-machine maintenance scheduling. (Part 1) . . . .	181
A.6 Existing papers for multi-machine maintenance scheduling. (Part 2) . . . .	182
A.7 Existing papers for multi-machine maintenance scheduling. (Part 3) . . . .	183
A.8 Existing papers for production and mission planning. (Part 1) . . . . .	184
A.9 Existing papers for production and mission planning. (Part 2) . . . . .	185
A.10 Existing papers for logistics. . . . .	186
A.11 Existing papers for automatic control. (Part 1) . . . . .	188
A.12 Existing papers for automatic control. (Part 2) . . . . .	189
A.13 Existing papers for jointly production/mission and maintenance scheduling. (Part 1) . . . . .	190
A.14 Existing papers for jointly production/mission and maintenance scheduling. (Part 2) . . . . .	191
A.15 Existing papers for jointly spare parts ordering and maintenance scheduling.	192
B.1 Results of the single-speed tests . . . . .	193
B.2 Results of tests on multiple profiles . . . . .	194

# Introduction

In the era of big data, companies have a direct access to huge data resources. The growing volume of data represents a big opportunity to the maturity of data analysis, machine learning techniques and artificial intelligence algorithms. Companies are tending to invest in the data science field to take fully advantage of the available data by drawing useful insights from data about products, customers behavior, and market research. However, due to the variety of the data types and their huge volume, integrating data-based analysis into the decision-making process seems challenging. This results in a more complex decision-making process due to the huge number of variables and conditions to take into consideration.

In most companies, big data in its variety is processed to draw valuable insights and to build predictive models to optimize the decision-making process. Hence, the need to integrate future information in the decision-making methods is constantly growing. In particular, the industrial companies are focusing on this point. Due to the growing competitive environment driven by the market requirements, the industrial sector is experiencing an economic pressure. This motivates the manufacturers to optimize their process in term of time, cost and quality which could be done by optimizing production and maintenance activities. To gain advantage in the market and with the emergence of industry 4.0, the companies have several sources of big data that they want to fully exploit. These factors contribute to the emergence of prognostics and health management (PHM).

Any system during its service time is subject to wear and tear. Its state is constantly deteriorating with the passage of time until it reaches a breaking point where the asset is no longer capable of achieving its intended function, i.e. it fails. Assets failure, gave place to define maintenance as the combination of technical, administrative and managerial actions to retain or restore the asset to a state where it is able to perform its function [52].

Before the 1960s, maintenance was only performed after the system fails. This fail-fix maintenance is called corrective maintenance. Then systems were getting more complex and critical. In addition, industries were subject to a strong competitive environment, environmental and social constraints. To face these new challenges, industries needed to reduce the maintenance costs and to increase its equipment availability, reliability and operational safety. Therefore, instead of waiting until the systems fails, maintenance activities are now performed before failure occurrence based on a predefined maintenance schedule. This strategy is known as preventive maintenance. Where the intervention schedule is defined according to reliability metrics such as the Mean Time Between Failures (MTBF). Later on, the development of condition monitoring technologies resulted in a maintenance strategy breakthrough. Critical components of the system are now identified, studied and supervised to monitor their degradation level and their health state. Maintenance is then performed according to the current condition of the equipment. This

is the condition-based-maintenance (CBM). Although CBM brought many advantages to the industries, it still presented some drawbacks. For instance, defining the maintenance activities based on the current state of the component was not enough to avoid the occurrence of failures. Thus, CBM could not anticipate the failure. Consequently, there is a need to forecast the evolution of degradation in the future and to estimate the remaining useful life (RUL) of the asset. Thus introducing the prognostics techniques. Goebel *et al.* in [67], defined prognostics as where one is studying the conditions of a fielded engineering system and whether it is behaving within nominal operation bound, if not, predicting where and when the system would no longer fulfills certain functional requirements. Emphasizing prognostics in the CBM framework was the initiative to the apparition of the Prognostics and Health Management (PHM) concept. In this thesis, we define PHM as a synthesized engineering discipline employing a series of tools and techniques to acquire process data and analyze information about the condition of the studied equipment, to predict its RUL and to propose the suitable decisions to better manage its health. In addition, in this thesis PHM is not limited to industrial maintenance. Instead, several natures of decisions are considered according to the kind of activity.

In general, integrating predictive information in the decision-making context would result in decisions that are highly dependent on the future information. However, these decisions in their turn would influence the evolution of the future. Hence, they could and maybe would result in a different outcome than the predicted one. Therefore, the future states also depend on the decisions to be made. Thus, the predictive information influences the decisions to be made, which in their turn would change the future and consequently result in another predictive information. Hence, the predictive analysis and the decision-making processes are highly interdependent. In particular, in the PHM context, the evolution of degradation is highly influenced by the changing operating and environmental conditions. Making the prognostics information highly dependent on the future loads and conditions of the system. On the other hand, to fully exploit the PHM value, the predictive information should be more integrated in the decision-making process.

In the PHM context, the condition monitoring and the prognostics have been extensively studied in the literature. Although, one can find several works about the decision-making process based on prognostics information, most of these works do not emphasize the prognostics information. In other words, in post-prognostics decision-making (PPDM) most research present some foggy points about the integration and the use of RUL in the decision process. Nevertheless, commonly the literature work in PPDM do not explicitly present the feedback of applying decisions on the systems prognostics. Actually, this feedback is considered implicitly by running the decision function several time. However, to our knowledge no previous work has studied or mentioned the problem of defining how often the decision function should be executed. Besides, the integration of future decisions in prognostics phase and the integration of RUL in the decision phase remains understated.

In this dissertation, the developments are achieved following a thorough literature review on the different types of post-prognostic decisions, i.e., maintenance, operational and mixed decisions. This allows identifying the key challenges related to the interactions between the decision phase and the prognostics phase. To account for such challenges, a new adaptation for the PHM framework is proposed by enhancing decision-making. The new PHM framework is tested on two applications and the overall performances are derived. The proposed framework also gives a new direction to decision-making in PHM context and provides new possible research questions for further investigation.

## Overview of Dissertation

The structure of the thesis manuscript is organized as follows.

Chapter 1 gives an overview of PHM framework and the role of each process. Afterwards a meticulous survey of decision-making research in the PHM context is presented including a comprehensive comparison between the approaches. Challenges in the PHM domain are highlighted to define open issues and consequently the problem statement for the thesis.

Chapter 2 presents the proposed new PHM framework. Therefore, the directive assumptions are presented along with definition of the needed concepts. The enhancement of decisions in the PHM is introduced through feedback loops. Parameters and constraints of the suggested framework are discussed and some directives to the implementation of the framework are given.

Chapter 3 is dedicated to the application of the proposed framework on a manufacturing case study. The study case consists of jointly optimizing production and maintenance activities on a single machine. Results are compared to a classic method of production scheduling combined with CBM. Moreover, the choice of the suitable decision method is discussed.

Chapter 4 presents another case study of post-prognostics decision-making in railways transportation systems. The application deals with the joint problem of scheduling maintenance and assigning missions for a fleet of railways assets. The initialization of the proposed framework is presented for the discussed problem. Moreover, the frequency of execution of the post-prognostic decision framework is studied for several decision methods. The influence of the decision horizon on the decision method is captured. In addition, the effects of some problem characteristics on the decision horizon are discussed.

Chapter 5 concludes this research work by summarizing the contributions through new adaptations of the PHM framework, improvements and limitations from case studies. A detailed discussion on future work perspectives is also laid out.





# Chapter 1

## Post-prognostics decision-making

### Contents

---

<b>1.1 Prognostics and health management</b>	<b>6</b>
1.1.1 Overview	6
1.1.2 Challenges and issues	9
1.1.2.1 Observation related issues	9
1.1.2.2 Analysis related issues	11
1.1.2.3 Decision support related issues	14
1.1.3 Synthesis	15
<b>1.2 Review of post-prognostic decisions</b>	<b>16</b>
1.2.1 Definition	16
1.2.2 Post-prognostic maintenance	18
1.2.2.1 Single machine maintenance planning	18
1.2.2.2 Multiple machines maintenance planning	22
1.2.3 Operational decisions	24
1.2.3.1 Production and mission planning	24
1.2.3.2 Logistics planning	25
1.2.3.3 Automatic control	26
1.2.4 Mixed decisions	27
1.2.4.1 Mission or production planning jointly with maintenance	27
1.2.4.2 Automatic control jointly with maintenance	28
1.2.4.3 Logistics jointly with maintenance	29
1.2.5 Synthesis	30
1.2.5.1 System health state and degradation models	31
1.2.5.2 The integration of prognostic information in decision-making	31
1.2.5.3 Recurrence of RUL estimation	32
1.2.5.4 Objective and resolution methods	33
<b>1.3 Open issues and problem statement for the thesis</b>	<b>33</b>
1.3.1 Decision-prognostics interactions	34
1.3.2 Toward decision-enhanced PHM	36
1.3.2.1 Issues to be addressed	36
1.3.2.2 Assumptions	37
1.3.2.3 Objective and contributions	37
<b>1.4 Summary</b>	<b>37</b>

---

In the prognostics and health management context, post-prognostic decisions are based on the integration of future information about the system's health. On one hand, these decisions are highly dependent of the estimated evolution of system's degradation. On the other hand, the evolution of the degradation is influenced by various conditions including environmental (e.g. temperature, humidity, etc..) and operational (e.g. speed, type of action, etc..) which are set up by the future loads of the system. Therefore, the future health states also depend on the decisions to be made. Thus, the predictive information influences the decisions to be made, which in their turn would change the future and consequently result in another predictive information. Hence, the predictive analysis and the decision-making processes are highly interdependent. This interdependency holds a major scientific lock for the PHM community and requires further investigation. In this aim, this chapter presents a overview of the prognostics and health management processes and its issues. A thorough survey on post prognostic decision-making is detailed. The classification of the literature works on the decision-making module is detailed. Methods, assumptions, degradation models, and objectives are derived from the literature works to map the existing works. Thereby, the gaps and the remaining challenges are identified. Thanks to that, the problem statement and objectives of this thesis are finally given at the end of the chapter.

## 1.1/ Prognostics and health management

The implementation of smart systems in the industry 4.0 made human machine cooperation, monitoring, process control, and high reliability and availability of the machines, major assets of the modern corporations. The digital transformation and big data emergence paved the way for the maintenance evolution from reactive to proactive. In this context, prognostics and health management have been an important milestone in the maintenance history by introducing predictive maintenance.

### 1.1.1/ Overview

Prognostics and Health Management (**PHM**) has gained the attention of researchers and industrial companies over the few last decades. As a matter of fact, PHM appears to meet the expectations of modern industry in terms of availability, reliability and operational safety. Indeed, PHM aims to ensure smooth functioning of critical machinery and to avoid undesirable events by managing the system based on its past, current and estimated future conditions. However, no consensual definition has yet been proposed. Uckrun *et al* in [171], defined PHM as an engineering discipline that joins the study of the failure mechanism and the management of the systems life cycle. Sun *et al.* in [165], described PHM as a methodology to not only predict the component responsible of the failure and when it will fail, but to also reducing the risks by studying the reliability of a system in its environmental, operational and usage conditions. The Center for Advanced Life Cycle Engineering (CALCE 2012) presented PHM as "*the mean to predict and protect the integrity of equipment and complex systems and avoid unanticipated operational problems leading to mission performance deficiencies, degradation and adverse effects to mission safety.*" Skima in [161], used the same definition as Sun *et al.* but specified that the objective behind the use of PHM systems is to manage the health of a system by minimizing its operations and maintenance costs. Feng *et al.* [55], defined PHM as an approach to manage

the assets life-cycle management. It seeks to reduce or to eliminate artificial inspections and planned preventive maintenance through accurate monitoring, fault detection and RUL prediction and maintenance decision-making techniques. Zhang *et al.* [194] presented PHM as a methodology to manage dynamically the life cycle of an asset based on its RUL. Moreover, the asset RUL is predicted from information collected from the equipment usage conditions including environmental, operational and future loads. Therefore, in this dissertation we consider PHM as described in Definition 1.1.1.

**Definition 1.1.1**

PHM is a synthesized engineering discipline employing a series of tools and techniques to acquire equipment data and analyze information about the condition of a given equipment, to predict its remaining useful life (RUL) and to propose the suitable decisions to better manage its health to optimize a given objective. PHM decision problems are generally multi-objective by trying to maximize the equipment's availability, reliability, and operational safety while minimizing the total operation and maintenance costs.

Historically, PHM started as an enhancement of prognostics in the Condition-Based Maintenance (**CBM**) context. Hence, the definition of PHM and CBM have always been blurred in the literature. Some authors consider PHM as part of CBM, other consider PHM as improvement of CBM, etc. To avoid confusion, in this thesis we differentiate PHM and CBM as follows:

- CBM** Condition-based maintenance is the process of scheduling maintenance intervention on a system based on its **current** and **historical** condition (i.e. health indicator, degradation level).
- PHM** Prognostics and health management manages the system (i.e. maintenance and operations) based on **future estimated**, **current** and **past** conditions.

The PHM framework is defined by means of 7 layers as described in the Open Standard Architecture of CBM [108]. This 7-layers architecture links the failure mechanisms with the management of the equipment over its life cycle (Figure 1.1). These layers are described as follow:

- **Data Acquisition** This layer is responsible for data gathering from traditional and smart sensors, data bus for communication, and from maintenance, operation and quality control history.
- **Data Processing** In most cases, gathered data are not readily usable. They require cleaning, filtering, denoising, averaging, and aggregation. These function are performed in the processing layer. Moreover, signal processing techniques are used to extract suitable feature through statistical and spectrum analysis. The data processing layer is considered as mature field of research. Moreover, one can find several works in the literature that studied this layer.
- **Condition Monitoring** Features obtained from the data processing layer describe the failure mechanism but usually these features need to be combined and/or fused into a feature that describes the system's health (i.e., health indicator). The condition monitoring layer is responsible for the building of health indicator (e.g., degradation

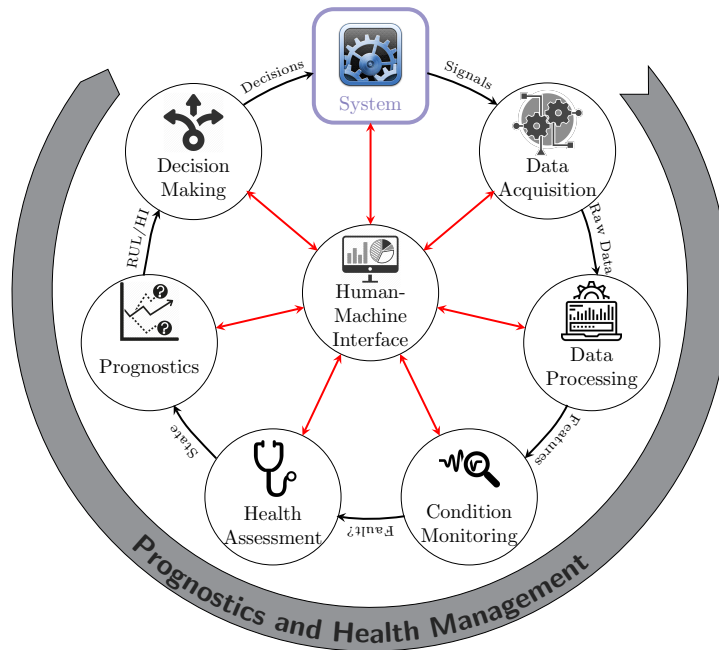


Figure 1.1: Prognostics and Health Management 7-layers architecture

model). Moreover, it compares the real-time characteristics of data to some expected or known values (i.e., thresholds) to detect anomalies, output the state of enumeration type, raise alarms, and assess the environmental conditions.

- **Health Assessment** This module estimates the current condition of the system. It helps the prognostics phase by specifying the starting point along the continuum of condition. A good health assessment method is required to obtain high prognostics performance.
- **Prognostics** This module is responsible for the remaining useful life estimation through modeling the degradation evolution and fault progression. Several works in literature defined prognostics. Hess *et al.* [83] stated that prognostics is the estimation of the remaining operational life of a component by forecasting its condition through a degradation model. Heng *et al.* [77] described prognostics as the forecast of an equipment's condition to determine its future health state, the remaining time to failure, or the probability of reliable operations. Goebel *et al.* [67] defined prognostics as "*the science of making prediction*" by estimating how long the system can still fulfill its purpose before failure occurrence. Different methods have been used for the prognostics layer. These methods can be classified into three categories: (i) data-driven methods, (ii) physics-based methods, and (iii) hybrid methods.
- **Decision-Making** Health management is an important layer of the PHM framework. It provides intelligent, informed suitable decisions for maintenance and operational activities. The decisions made in this layer, are based on the use of prognostics information such as the remaining useful life, operational reliability, degradation level, and failure probability.

- **Human-Machine Interface** This layer is responsible for information visualization concerning all different layers. The PHM framework is designed to convey the right information through this layer to the right person.

In other words PHM focuses on sensors data acquisition and processing to obtain features that describe the health condition of the system. The obtained features are used as support to monitor the equipment condition, detect and diagnosis its faults and estimate how long will it take until the system is no longer able to achieve its intended function, i.e., the remaining useful life. This information is used to efficiently manage the system by taking the suitable decisions that optimize the industry objectives.

Recently, the 7 layers of the PHM framework have been grouped into three processes [71]; (i) observation process that contains the data acquisition and processing; (ii) analysis process in which the system's conditions will be assessed, followed by diagnostics and prognostics, and (iii) action process that involves decision-making and applying through the human-machine interface.

### 1.1.2/ Challenges and issues

The first articles that referred to the prognostics and health management context goes back to the early years of the 21<sup>st</sup> century. Hence, the study of PHM process is still considered as a recent field of research. Therefore, PHM phases and modules present several issues and challenges. In this section, we try to cover most challenges of PHM grouped by phase and layer.

#### 1.1.2.1/ Observation related issues

The observation phase is mainly about data acquisition, processing, and feature extraction. In this context, data can be classified into two types, (i) condition monitoring data (i.e., system's parameters) these data are acquired from equipment's sensors or manually during inspections, and (ii) process related data that includes quality control data, maintenance history, operational data such as operational schedule, and event data (i.e., failure and fault data). The acquisition of both types of data can represent several issues. First issue of the observation process is data accessibility. In some situations, data required to develop predictive models must be acquired from operators that should be convinced to release it. In other situations, relevant data could require an extraction of signals from subsystems manufactured by subcontractors. In some of these cases, the subcontractors are not willing to share the internal logic or the design of their components nor the access to the installed sensors. Furthermore, implementing PHM in already operational systems could present some challenges especially when the intended system has a complex design. In some cases, PHM designers are forced to work with the equipped sensors and could not install more due to the high complexity of the system's design. In these situations, they are obliged to monitor the system's conditions via those sensors which could present some challenges such as their resolution or their purpose. Therefore, engineers should measure certain parameters indirectly because the purpose of the acquired signals could be different from their objectives. Despite the accessibility challenges, condition monitoring data are usually automatically collected and available in big amounts. While, process related data entry is often manual. Thus, these data are much less available than the

measured parameters and they tend to be incomplete and more error-prone. They also require much more efforts in their acquisition and processing. Consequently, it is hard to correlate problems in the quality control data or fault presence in the event data with condition monitoring data.

The quality of the built prognostics model and its performance are directly linked to the quality of the used data. Data quality depends on several factors like; characteristics of the used sensors, transportation and storage supports, presence of irregularities and so on. On the one hand, data that describe the evolution of component's degradation take a huge time (months or years) to accumulate. Therefore, it is hard to obtain several experiments without accelerating the deterioration process. Accelerated tests are performed in highly controlled environment (i.e., labs). Although they describe the evolution of degradation, they eliminate other factors that could affect the deterioration process for example loads variation, environmental conditions (humidity, temperature), and the deterioration of other components. On the other hand, in real life applications, industrial communities and asset owners do not allow their equipment to run to failure. Usually, these assets are stopped at the presence of the first fault. Therefore, the collected data describe the degradation evolution until the first fault and do not reach the asset's failure. Such data are called censored or suspended data [77]. The use of censored data to design the predictive module implies that the replacement time is considered as failure. However, this defeats the main purpose of PHM, which is the time interval during which the failing unit can still fulfill its intended functions beyond the first fault detection. Nevertheless, omitting the suspended data will worsen the data unavailability problem.

Acquired data are not in ready to use format. Usually, these data are in different format and different time base. This is mainly due to the nature of data (condition monitoring and event data), different types of measurement, several types of technologies, and frequencies of data acquisition. Therefore, there is a need to put all acquired data in a useful common format to facilitate their exploitation. Moreover, the acquired data are not always of a good quality. Meaning, during their acquisition, transport and storage they could be contaminated. Data corruption includes noisy signal due to electromagnetic interference, missing redundant values (i.e., if one of the redundant sensors fails), irregularities in measures, and the confusion of redundant measures (i.e., for example the system is equipped with two sensors to measure the same parameter and the acquired values are completely different).

In real life applications, it is relatively common to collect redundant or even unnecessary information. This results in a high dimension data. It is stated that increasing data dimension reduces the uncertainty [54]. When data classifiers are used, high dimension data produces low error rate at the beginning, but increasing dimensions carelessly could produce higher rates of error. This is known as the curse of "*dimensionality*" or the peaking phenomenon [89, 98]. In this context, feature extraction and selection are of a great importance. The compression of data could help data analysis to better understand information. It could also improve the computational efficiency while keeping memory requirements within practical limits. However, selecting the suitable data and relevant features is a delicate process. Pertinent feature would result in a good wide separation of the machine's health states. Adding inappropriate features could cause a smaller separation, in which normal health states would overlap the failed status. Thus, the selection of feature is a great challenge in the observation phase.

Finally, one big challenge of the observation phase in the case of on-board, decentralized,



geographically distributed, and mobile/moving applications is the data storage problem. For these applications, often it is hard or even impossible to implement the PHM system on the assets, e.g., in airplanes or trains implementing PHM on the asset will require a great computation capacity, i.e., calculators of big size. Such applications require a real-time transmission system or a storage one. Moreover, due to the huge size of acquired data it is often inappropriate to use transmission. Therefore, data storage is often the suitable solution. This solution requires a precise study of the size of the storage capacity and additional activities should be scheduled on the assets to recover the stored data and reset the storage system.

#### 1.1.2.2/ Analysis related issues

An important step of implementing PHM framework is the understanding of features and combining them to capture the evolution of the system's health state. Understanding features and data and mapping them to the health states requires a model of the system's deterioration. One can define degradation model as a computational model that describes to a certain certitude how a given system progresses in time to a state or an event [67]. Degradation models could either be physical [97], statistical [146, 198], or data-based [93]. Several works in literature have studied degradation modeling [70]. However, choosing the right model for a specific application is still a difficult task. Usually this is done in a heuristic way based on experience and expertise. Yet, this method is not efficient especially in cases of complex systems, inexperienced designers, or systems/components that are not extensively studied.

In general, each layer of the PHM framework requires a choice of algorithms. This choice should be based on the systems requirements and characteristics, data properties, technical and non-technical constraints, and eventually the performances of the algorithm outputs. In most literature works, the choice of the suitable methods for prognosis or decision-making was not discussed nor justified explicitly. Lee *et al.* [109] proposed a methodology to select the appropriate prognostics algorithm. The tool is based on quality function deployment and the algorithm is chosen based on data characteristics, available signals, working conditions, system dynamics, historical data, and expertise.

Condition monitoring is responsible for alarm generation and fault detection by comparing the recent acquired signals to some predefined thresholds. Therefore, the performance of such module is judged by its capacity to detect as soon as possible the deviation of system operations from the normal behavior. This capacity is directly related to signals quality and to the defined thresholds. One can conclude on the importance of threshold definition, but this task was rarely discussed in literature. Defining suitable thresholds allows the PHM system to early detect behavior deviations thus giving the PHM system the opportunity to anticipate failure and to plan suitable decisions. An early fault detection guaranties a better performance of the overall PHM process. Thus, defining threshold should be further studied.

The main purpose of PHM is to predict the date of failure occurrence (i.e., RUL) during the operation of a given system, and to propose/make suitable decisions to eliminate, delay, or mitigate the failure and its effects. Unlike deterministic method used to diagnose the failure source, health assessment methods often face some uncertainties in case the degradation process is non-observable or partially observable. When the degradation is observable such the case of the airplane crack growth [134], the assessment of the current health state

can be accurately estimated through inspections. While in partially observable degradation for example the rolling element bearing [149] the current health of the component is at best approximated and most likely contains uncertainties. In the PHM context, one needs to know the current health state to be able to forecast its evolution. Therefore uncertainties in the health assessment is propagated in the prognostics results. Moreover, prognostics methods involve several unknowns (i.e., uncertainties) about the current and future system health states and operations. Uncertainties present a big challenge for the prognostics process as they could be caused by different sources. Thus, they need to be carefully addressed in mathematical and statistical manner. Several works have studied the PHM uncertainties, their influence, and their sources [53, 112]. Some authors even proposed methods to quantify and manage these uncertainties [138, 155, 168]. Lopez and Sarigul-Klijn [123] have stated that uncertainty can be caused by five factors; (i) system knowledge (e.g., boundary conditions, complexity); (ii) signal processing methods; (iii) model uncertainties (e.g., parameters, type); (iv) future decisions; and (v) environmental and operational sources (e.g., weather, loading conditions). Goebel *et al.* [67] proposed a classification for the sources of uncertainties according to the system current and future conditions, modeling uncertainties, and chosen methods uncertainties. These sources of uncertainties can be grouped into two classes; (i) uncertainties that are related to the PHM design and implementation (e.g., model and methods uncertainties, future decisions); and (ii) uncertainties that are external to the PHM system (i.e., uncertainties that already exist in the system and its environment). In this dissertation, we propose the use of a two level classification. First, we classify the uncertainties according to their relevance to the PHM system design and implementation. Then, we group for each class uncertainties under their common source. This work may not be a complete one with an exhaustive survey on all existing uncertainties' sources but it is committed to understanding the causes of uncertainties and drawing attention to their importance.

**PHM design related uncertainties** In this paragraph, we discuss the possible sources of uncertainty faced during the design and implementation of a PHM system. First, while designing the PHM system, the degradation of the system/component is modeled. The design of the deterioration model (e.g., choice of the model's type, estimating its parameters) is a source of uncertainty. Some degradation process cannot be modeled through exact physical models (i.e., based on differential equations that describe fundamental principles of mass, energy and momentum conservation) due to a lack of physical understanding of the phenomena. These processes are modeled through machine learning techniques in which the model is learned from historical data. Therefore, they present some uncertainties in their estimation. Even for physics-based models it is often difficult to get the exact value of some parameters and their estimation cannot be fully accurate due to computation resources limits. Once the model is set, the current health state is estimated based on the acquired data. This process is influenced by; input data quality (e.g., noises); signal processing techniques (e.g., valuable information can be lost in the filtering process); and the health assessment algorithm. Estimating RUL is a delicate process and is influenced by several unknowns related to the chosen model. For instance, the choice of the algorithm input influences the prognostics results, i.e., considering future loads of the system in the prognostics algorithm can increase the accuracy of the estimated RUL. The parameters of the algorithm can also cause uncertainties, e.g., approximations and sampling errors. Finally, each step of the PHM observation and analysis phases could present uncertainty and since these phases are interdependent the uncertainty of one process output is



propagated to the next process. Uncertain health assessment would worsen the prediction results [83]. Thus, the propagation of uncertainty is an uncertainty in itself.

**1.1.2.2.1/ System and environment uncertainties** In this paragraph, we discuss the uncertainties that are not caused by the PHM system design but are originated in the system, its future loads, and its environment. A given engineering system is a source of uncertainty. They are caused by the system's complexity, its different failure modes and dynamics and their influence of its health, and the interactions between the system components. Other uncertainty aspect of a given system is its future operating conditions. Future operating conditions include the future load of the system, future decisions, and its operating profile (e.g., speed, feed rate). Most of the works on prognostics algorithms assume that the future loads of the system are constant with constant operating profile. However, in real life applications systems are not that static. It was also proven that degradation is a function of changes in the work schedule and the operating parameters (e.g., rolling element bearing degradation is highly influenced by the rotating speed and the load [136]). Moreover, effects of decisions like maintenance are not considered on the system health. In fact, maintenance is usually considered as perfect i.e., it restores the system to a as good as new state. However, the degradation rate of the system/component could change after such decisions. Finally, the degradation of the system is also influenced by the environmental conditions like temperature, humidity, and corrosive exposure. Baghdadi *et al.* [7] included the influence of the storage temperature of lithium batteries in the capacity degradation model.

To summarize the uncertainties in the PHM context are presented in Figure 1.2. It is certainly useful and beneficial to quantify and estimates future conditions. However, one cannot be certain about what will happen. Therefore, uncertainties can only be minimized but not completely eliminated. However, PHM designers should be careful when addressing uncertainties, for minimizing them could require an enormous computation resources. Thus, we believe that a compromise between the acceptable amount of uncertainty and computation requirements should be established.

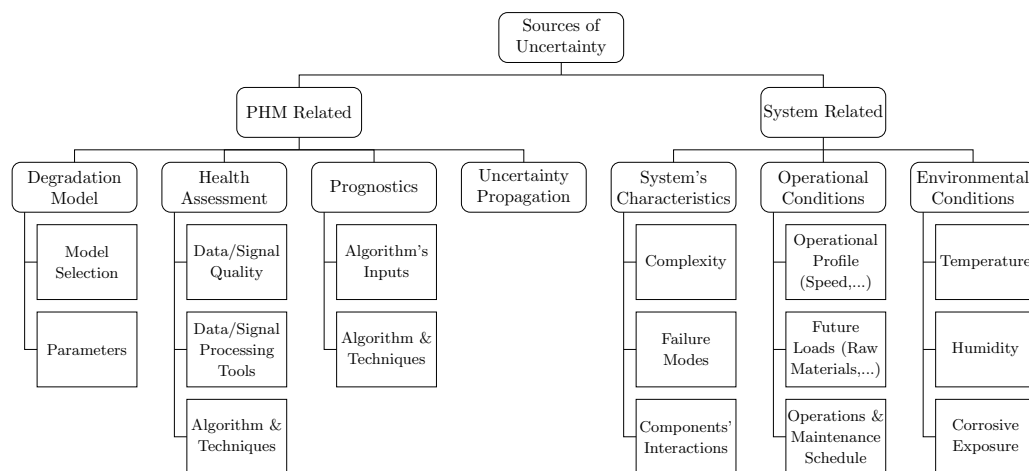


Figure 1.2: Sources of Uncertainty

Due to uncertainties, the RUL is usually expressed as a probability distribution function.

Therefore, estimating the exact time of failure (i.e., a single deterministic value) has a zero probability of being correct. While predicting an interval of time that covers all the distribution is certain (i.e., has a 100% probability of occurrence), it is completely useless because usually it is a large interval. Therefore, there is a need to define some limits to the predicted RUL value. These limits are called upper and lower bounds of predictions. This is known as the prognostic paradox [83], in which the more precise the estimated RUL, the less likely the failure occurrence will happen at the predicted date.

Another common behavior of prognostics algorithm is the accuracy variation regarding the duration of prediction. In other words, it is more common that short term prognostics (i.e., predicting the behavior of a system for a short period of time) are more accurate. Moreover, the more far ahead predictions are made, the more they are subject to greater uncertainty and thus the resulted RUL has a higher error rate and is more inaccurate. Therefore, designers of PHM system should consider this prognostics behavior and should define the suitable prognostics horizon. Prognostics horizon is defined by finding a balance between the intended prognostics accuracy and the dynamic of the given system/component degradation. Moreover, the definition of the prognostic horizon should also take into consideration the possible decisions and their action time. For example, if the designed prognostics horizon is shorter than the action time of any possible decision then the PHM system becomes useless, because the user does not have the time to implement decisions to avoid the predicted failure.

Performance evaluation of the used prognostics method has been a great challenge in the PHM community. However, several metrics were proposed to evaluate these performances. Basically, these metrics represent the computation of the difference between the estimated RUL and the real date of failure occurrence. Vachtsevanos [172] defined a set of possible performance metrics to evaluate the prognostics algorithms according to the main objective of prognosis and the quantity of uncertainty. However, the proposed metrics are obtained by comparing the estimated and the actual failure dates. Moreover, they are obtained on historical run-to-failure data. But, in real life application, systems are usually not allowed to run to failure. In these cases, how can one evaluate the prognostics performances?

### 1.1.2.3/ Decision support related issues

The estimation of the remaining useful life i.e., prognostics is performed to provide new important information to the decision making process to better manage the system life cycle. The decision support system can be seen as an optimization problem solver. As defined in the PHM architecture, the decision-making process seems to be a consequence of the prognostics module. Therefore, the challenges present in the decision support phase can be highly related to those of the analysis phase.

The decision-making is supposed to use available prognostics information. Therefore, one may directly wonder how these information are integrated in the decision process?

However, as we mentioned earlier, RUL and health indicators are presented as distribution functions or as values with lower and higher bounds. Therefore, one may wonder how does this RUL and HI uncertainties affect the decisions? Does the made decisions reflect these uncertainties? Can we talk about decision uncertainty?

The prognostic module is sensitive to the duration of prediction (i.e., the prognostic horizon). We mentioned that the performance of the prognostics method is highly influenced by the

parameter. Since the accuracy of the prediction decreases over time, does this implies that the decision-making process performance also decreases? If the decision efficiency is a function of the duration of decision-making (i.e., the decision horizon), what is the optimal value for this parameter? And what factors influence this horizon?

Finally, like any other module of the PHM framework, the decision-making requires a method selection. Through out the literature, one can find several methods for optimization problem solving; exact methods , to heuristics , and meta-heuristics .With a large variety of methods, one can have a hard time finding the suitable method for his problem.

### 1.1.3/ Synthesis

The prognostics and health management research domain is still new which explains the various challenges still not addressed. The domain is far from being mature. Thus, it allows a big opportunities for research. The aforementioned challenges are grouped by phase in diagram presented in Figure 1.3.

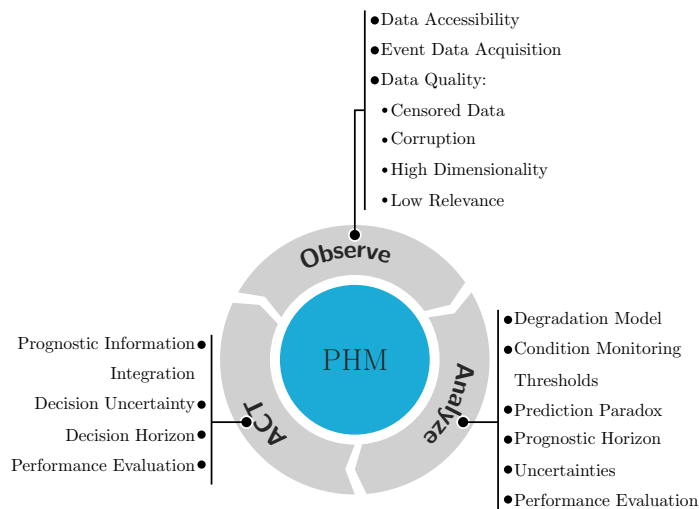


Figure 1.3: Synthesis of PHM Challenges Grouped by Phase

We are aware that the scientific development of new disciplines requires a lot of small cumulative contributions. It is a process of collaborative research efforts. One cannot expect to address many challenges at once. After some bibliographical research we noticed that despite the importance of the decision support module, it is not studied with the same frequency of the prognostics phase. Therefore, an opportunity is presented to fine tune this process by addressing its various challenges. Hence, the work presented in this thesis will mainly focuses on the prognostics and health management framework from a decision making point of view. We start by identifying the literature work in this context and analyzing their omissions.

## 1.2/ Review of post-prognostic decisions

According to the origin of PHM, one can define a decision, in PHM, as set of maintenance actions. This decision is usually the solution of an optimization problem consisting in defining the adequate interventions dates. But the use of prognostics evolved to include other sorts of decisions. For example, prognostics information were used also to define processes set points and controllers parameters in closed-loop actuators control as mentioned in [148] or in [104]. Other works integrated the prognostics information in production scheduling and mission assignment, where the health prognostic of the system was defined as a constraint or included in the optimization objective which was the case in the works of [80].

Gouriveau *et al.* stated in [71] that the PHM is not limited to industrial maintenance but can be applied to any kind of activity as long as it fits the elementary process defined by the OSA/CBM decomposition. The only difference is in the nature of the decisions to be taken. The evolving nature of the integration of prognostics information in different kind of decisions calls for a review of the existing works in this domain.

### 1.2.1/ Definition

Literally decision making is the name given to the process of selecting the logical best choice from a list of available options. In such a process one should weight the pros and cons of each possible option, consider all the presented alternatives and forecast the outcome of each option. Such process can either be performed by human operators or a systemic approach that can support the human operators decision making process. As a result, one can model the decision making problem with an optimization problem. Since human decision makers are limited in their cognitive capacity to process and analyze large quantities of information, in the context of industry, the decision making process will be performed by an algorithm.

In PHM context, we are supposed to take the adequate actions to prevent or to mitigate the degradation of the studied system. Thus the ultimate goal of PHM is to make decisions based on prognostics information. Iyer *et al.* in [87] were first to introduce the term of post-prognostic decisions to describe decisions made in the PHM context using prognostic information from the system to choose the most adequate actions for logistics platforms, maintenance, supply chain management, mission planning and mission allocation. Lately, Balaban *et al.* in [8], defined the post-prognostic decision as a decision that takes in consideration the prognostic information (i.e. RUL) in order to define the future utilization of the system in question so as to optimize a predefined objective. This same definition was used by Herr in [78] and by Chebel-Morello *et al.* in [30]. Recently in [67], Goebel *et al.* gave a more precise definition for post-prognostic decisions. As a result, one call post-prognostic decision the set of actions at time  $t$  that best overcome an undesirable future event, predicted to take place at  $t + t_E$ , by satisfying a given set of constraints and optimizing a set of objectives expressed in a cost function.

In the same context, in this dissertation we define post-prognostic decisions in the industrial world as described in Definition 1.2.1.

**Definition 1.2.1**

Post-Prognostic Decision, in the prognostics and health management context, is a sequence of configurable actions built over a predefined duration, called decision horizon. This series of actions is built at a given instant to best manage the health of the system, its operations, and its missions under the prospect of an undesirable future event, foreseen to take place at an estimated date in the future. The built decision must satisfy a given set of constraints and optimize a set of objectives expressed in a cost function.

Moreover, prognostics are widely used in different fields and contexts but under another name for economics [51, 61] or weather [63, 65] it has been conventionally called forecasting, predictions in health care [156, 163], in the ecological field [3], politics [57], etc. All these names are used to defined the same concept of estimating future events, trends in variables, or outcomes of decision to enhance the decision-making process by taking into consideration past, present, and future information. Therefore, one can deduce a more general definition of post-prognostic decisions as proposed in Definition 1.2.2.

**Definition 1.2.2**

Post-prognostic decision, in general, is a strategy of operations over a predefined duration called decision horizon. At a given instant, this strategy is built by sequencing configurable actions based on past, present, and future information to optimize a set of objectives.

In this thesis, we focus on the industrial context to study the applications of the prognostics and health management. In the next section, we present the overview of the industrial post-prognostic state-of-the-art.

## Industrial Literature Overview

The growing number of publications per year proves that the studies on post-prognostic decision is an expending research domain. Also we can notice the absence of papers that reviews the integration of prognostics information in decision making and the importance such a paper can have to defines the current and future challenges in post-prognostic decision making.

## Classification

In [67], Goebel *et al.* proposed a classification of the possible prognostic based actions to take based on the duration of the prognostic horizon. The proposed classification contained; controller reallocation, mission rescheduling, and maintenance plan optimization. In this paper, we propose another classification of the post-prognostic decisions based on the nature of the action in itself. The analysis of the type of decisions discussed in the obtained papers helped in defining the categories. As a result, we propose to classify the post-prognostic decisions into three categories using the type of the decision as a criterion: (i) Decisions that describes the schedule of maintenance activities, (ii) Decision that influence the operational condition of the system and (iii) Mixed Decision that optimize

Table 1.1: Publications on Post-Prognostic Decision per Decision Type

Type of Decision	Maintenance	Operational	Mixed
Publications	50	25	28
Percentage	48.5%	24.3%	27.2%

jointly the maintenance of the system and its operational conditions. Table 1.1 represents the distribution of articles according to the three categories mentioned above. One can easily note that the maintenance decisions are more dominant. indeed, operational and mixed decisions are a more recent field of research. The aforementioned categories are detailed in the next sections, where a further analysis is conducted on subcategories, an analysis of the related work by reviewing their ideas and by investigating their main assumptions.

## 1.2.2/ Post-prognostic maintenance

The works that considered maintenance scheduling and maintenance interventions optimization based on prognostics information are studied. In the literature, post-prognostics maintenance planning can be divided into two categories according to the number of the considered machines. Single machine maintenance planning and multiple machine maintenance scheduling are detailed in the following subsections.

### 1.2.2.1/ Single machine maintenance planning

In the context of scheduling post-prognostic maintenance activities, some authors focused on single machine applications. These works are grouped according to the application and presented in the following paragraphs.

**1.2.2.1.1/ Manufacturing** Zhao *et al.* [199] presented a new method for optimizing the maintenance policy for a system with a single deterioration, to minimize long-run maintenance costs. Camci used genetic algorithms in [27] to schedule maintenance interventions for a multiple-component single machine. The schedule developed by this method minimizes the maintenance cost and satisfies the constraints on available resources. Tian *et al.* [170] considered the maintenance scheduling of a multiple-component single system in which the identical components are economically dependent. In this paper, Tian *et al.* used the proportional hazard model instead of RUL to estimate failure probability. Based on the obtained failure probability, suitable maintenance actions are made. Van Horenbeeck and Pintelon in [173] scheduled the maintenance activities of a multiple-component single machine on a finite rolling horizon. The proposed approach consists of selecting the suitable dates for maintenance interventions and then updating them when new prognostics information is available. In addition, a grouping algorithm was used to group maintenance activities of the different components based of their dependencies. The three considered components that presented different kinds of dependencies. Finally, the paper presented the effects of dependencies of the component lifetime with different maintenance policies. In the same context of a rolling decision horizon, Rodrigues *et al.* [151] proposed a dynamic maintenance strategy for a group of similar components



monitored with a PHM system. At each inspection time, the prognostics information of the components is estimated. Based on the failure probabilities of the components a two-level decision-making process is executed. At the first level the cost of maintenance of each component is minimized. Next, the maintenance activities are grouped to minimize the overall maintenance cost. The obtained schedule is applied until the next inspection, when new health information is available and the rolling horizon is shifted. Khoury *et al.* [95] presented a framework for predictive maintenance of a single machine over a finite rolling “visibility” horizon. The decision horizon is characterized by two maintenance opportunities in which maintenance actions could be exclusively planned. The main objective is to find a trade-off between doing premature maintenance at the first opportunity or planning the intervention too late at the second opportunity or later based on a failure probability obtained from the RUL distribution. If the failure probability at the maintenance opportunity exceeds a certain threshold, then a maintenance action is scheduled in that opportunity; otherwise, it is postponed. Huynh *et al.* in [85] studied the case of a single-component single machine with aging effects. In this paper, the machine could fail due to degradation or to the effect of a shock caused by the accumulation of degradation and the aging of the machine. For this reason, Huynh *et al.* proposed two maintenance policies based on the rate of the machine’s degradation and the mean residual life. In [86], Huynh *et al.* not only proposed unusual new prognostic-based maintenance strategies built on various variants of the systems RUL, but also compared these strategies to diagnostic-based strategies and CBM and then quantified the performance and robustness of the diagnostic and prognostic indicators for maintenance decision-making. Tang *et al.* [166] proposed two control limit maintenance policies based on the current degradation of a single component machine. The first policy consists of periodic inspections using a CBM approach, while the second policy consists of delaying the first inspection time under the assumption that a slowly degrading system is less likely to cross the failure threshold at an early stage. The paper also presented a way to estimate the RUL of the machine, but this RUL was not integrated into the decision-making process. Do *et al.* [45] used a gamma process to describe the degradation of the system. The authors proposed an inspection-based CBM strategy with the possibility of choosing the quality of the preventive maintenance intervention. This work also presents a novel method for the definition of the inspection dates based on the RUL of the system. The failure probability of the system is estimated from its RUL, and the next inspection is scheduled to guarantee that the failure probability does not exceeds a required reliability level. Langeron *et al.* [103] applied the maintenance scheduling on a controlled system subject to a random deterioration of the actuator. The maintenance optimization was carried out on the controller settings along with the RUL threshold and the inspection period, at which point two possible actions were considered: either a preventive replacement or a corrective one. Shi *et al.* [158] considered the case of stochastic dependencies of components when the degradation of a component affects the degradation of another. The proposed approach in this paper used a dynamic opportunistic maintenance scheduling based on a real-time RUL prediction. Liu *et al.* [121] developed a new approach for maintenance intervention scheduling on a manufacturing machine to optimize the maintenance cost based on prognostics information. The approach consists of dividing the health state of the machine into four levels with two different maintenance actions. The problem is then solved using dynamic programming model. Wang *et al.* [183] applied a PHM framework to electro-mechanical systems to make decision about components refurbishment policy. Actually refurbishment consists of making component replacements to restore the system to satisfy its initial specification. In other words refurbishment could be considered maintenance intervention

by replacement. The method is based on the estimated RUL of the considered system and its components to define an optimal replacement policy for the deteriorating components while considering the impact of such policies on the system's health for the purpose of minimizing the refurbishment cost.

**1.2.2.1.2/ Wind turbines** Lei *et al.* [110], studied maintenance decision-making based on PHM information applied to wind turbines. The approach consists of using real options analysis to schedule maintenance action. To take uncertainty into account, the authors used simulation paths by which each path represents one possible future scenario to finally determine the optimum predictive maintenance opportunity. Mazidi *et al.* [127], presented a hybrid method of neural networks and proportional hazard model to wind turbine behavior for prognostics and maintenance management. The approach evaluates the efficiency of previously applied maintenance plans. First, a neural network is built to model the normal behavior of the wind turbine with data gathered from a supervisory control and data acquisition (SCADA) system. Then a deviation signal that describes the stress condition and the health status of the wind turbine is extracted by comparing the real-time data to the neural network predictions. The obtained signal is used to assess the goodness of previous maintenance actions and propose suggestions for future maintenance planning. This work actually takes into consideration the effect of decisions on the system.

**1.2.2.1.3/ Aerospace** The aerospace domain is intensively studied in the case of single machine systems. Si *et al.* [159] proposed a CBM replacement strategy for stochastically deteriorating aircraft components. The framework consists of a classical periodic inspection CBM policy, and even although the reliability of the components is computed it was not considered in the decision-making process. Instead, decisions are made on the current observed degradation level. In [177], the authors used a Kalman-filter-based prognostic method to estimate the future degradation and RUL of a multi-component redundant system while assuming different wear profiles. The obtained prognostic information from the multiple model of prognostics is used to specify the maintenance intervention dates that minimize the maintenance cost. Several operational aspects were considered in this work by integrating their cost into the maintenance cost. Wang *et al.* [187] used a model-based prognostic framework to monitor and estimate the evolution of the size of cracks in the fuselage of an aircraft. The PHM process is done periodically through inspections and decision-making. The decision-making process includes the obtained prognostic information to schedule the maintenance intervention in the current available stop or to postpone it to the next one if the safety level allows it. The maintenance cost is minimized in this framework by finding a trade-off between the probabilities of failure occurrence and the waste of RUL of the fuselage. Later in [188], the authors integrated unscheduled maintenance to work in tandem with the scheduled opportunities and considered the engine and some non-structural components.

**1.2.2.1.4/ Railways** Lin *et al.* [118] used a partially observable Markov decision process (POMDP) to plan maintenance interventions for a traction power supply based on its RUL. The degradation of the power supply is modeled by Gauss-Poisson process, in which natural and sudden deterioration are considered. Health indicators are deduced from the degradation to define the health of the system. The POMDP is solved using a one-pass algorithm that is based on n-computation through dynamic programming. In [55], the



authors developed a novel framework that combines PHM and active maintenance to better manage the health and maintenance activities of high speed railways' traction power supply system. The proposed framework consists of classical PHM technology in which RUL, reliability and risk are estimated from different data resources. Active maintenance is used in the PHM decision-making module. This module has the objective to select the suitable maintenance strategy for each component of the system (CBM, corrective, periodic preventive or RUL-based predictive) to avoid accidents and guarantee a predefined reliability requirement.

**1.2.2.1.5/ Other type of applications** Nzukam *et al.* [143] proposed a maintenance decision support system for heating, ventilation and air-conditioning (HVAC) of non residential buildings. The method considers the components' RUL and their criticality on the one hand and the planned stoppages of the system on the other. The RUL distributions of each component are estimated and compared to the stoppage dates to produce a list of opportunities for maintenance actions. A grouping algorithm takes into account the RUL, the maintenance duration and the severity of each component and the duration of the maintenance opportunities, to produce a schedule of grouped maintenance interventions. Later on, the authors integrated uncertainties into their model in [144] by supposing that the duration of the stoppages are stochastically defined. Effects of the uncertainties were studied and the consistency of their method was proved using Monte Carlos simulations.

Later, and still in single machine maintenance optics, a new structure of maintenance decision-making is proposed. Instead of using a central unique maintenance decision-making process, these papers proposed to use a two-level method for maintenance scheduling. Huynh *et al.* [84], proposed a two-level maintenance decision-making for multi-component complex systems. The novel RUL-based maintenance scheduling framework combines the system-level maintenance decisions and the component-level one. A periodic inspection is implemented, and at each inspection, the component's degradation level is updated and corresponding new RULs are estimated. Then the RUL of the system is determined by the  $k - out - of - n : F$  structure. If the system's RUL fall below a predefined threshold, at the system level, an intervention should be made, and so the component level of the framework is triggered. At this level, a list of the components to maintain is made based on the components' RULs and the economic dependencies using opportunistic maintenance strategies. Nguyen *et al.* in [140] used the same concept of the framework presented in [84], except they improved the framework by implementing the use of reliability obtained from the RUL distribution instead of using the RUL directly. Also in this updated framework, the objective at the component level is now defined as selecting and grouping the components to be maintained based on their predicted reliability and economic and structural dependencies while the system level remains unchanged by comparing the reliability of the system to a threshold. Unlike the previous works, Verbert *et al.* [176] started by optimizing the maintenance strategy at the component level and then moved to the system level. First, for each component, the maintenance options are evaluated considering the risk tolerance and the predicted degradation level, and then the optimal option is chosen. After choosing the maintenance option for each component, at the system level, the maintenance strategy is optimized to account for economic and structural dependencies and the component's chosen maintenance options.

### 1.2.2.2/ Multiple machines maintenance planning

In this subsection, works on maintenance planning on multiple machines are described. These papers are classified according the type of application.

**1.2.2.2.1/ Production shop floor** Yang *et al.* proposed in [193] an approach to schedule maintenance activities on manufacturing systems with different typologies of machines. The decisions were based on the forecasted degradation of the machines. Ambani *et al.* [5] defined an approach to chose the adequate maintenance policy for multiple serial machines to maximize overall profit. To avoid conflicts caused by the use of common material resources, they proposed two policies for prioritization based on stoppage time and failure sensibility. In [126], the authors proposed a framework for maintenance planning in automotive manufacturing industries based on data collected from programmable logic controller (PLCs), quality control services, machine failures and production planning. This framework correlates real-time data to historical failures to predict failure events, health indicators (like RUL) and quality deterioration. The intervention dates are then suggested by the perspective maintenance support system, thus allowing the operators to make the final decision. In [128], Meraghni *et al.* developed a PHM framework based on Internet-of-Things and cloud computing and cyber physical systems to help maintenance decision-making in big industries. The framework includes maintenance resource allocations, maintenance planning, and traveling cost oriented for maintenance services based on the estimated RUL of the different machines. An example of maintenance technicians traveling cost minimization is given in which the positions of the machines are known and their relative RUL is provided by the prognostic service.

**1.2.2.2.2/ Wind farms** Tian *et al.* [169] addressed the problem of maintenance scheduling for multiple-component wind Turbines in Which a wind turbine is considered a serial connected components (i.e., if one component fails, the whole system fails). The only dependency considered in this work is economic factors. The policy is based on the failure probabilities that are extracted from the RUL of the wind turbines. Haddad *et al.* developed, in [75], a new real-option model to optimize the maintenance of offshore turbines based on prognostic indications. The real-option analysis is used to value the maintenance options depending of the RUL of the turbines, thus leading to resolution of the optimization problem of whether to send a maintenance vessel, and if the maintenance team is to be sent, which turbines should be maintained. Lei *et al.* developed in [111] a prognostic-based maintenance schedule for wind turbines that are subject to power purchase agreements and prefixed maintenance opportunities. The estimated RUL of the wind turbines is used to define the suitable date of the predictive intervention from the available opportunities. The RUL is periodically estimated to reduce the uncertainty level. At each sampling time, the decision is updated to make sure that the power purchase agreement constraints are satisfied. Wang *et al.* [185] developed a prognostic informed maintenance decision-making process that takes into consideration the different operational and maintenance costs. In this framework, the degradation level is monitored through periodic inspections. Once it exceeds a predefined threshold, the next maintenance intervention is set based on the estimated RUL by finding a trade-off between operational and maintenance costs.

**1.2.2.2.3/ Aircraft fleets** Goebel *et al.* [87], proposed a framework for post-prognostic decision-making for airplanes under the constraints of workshops and spare parts availability. Although the work did not include the use of prognostic information, it was a promising approach that was referenced by almost all of these papers. Balaban and Alonso proposed, in [8], an approach for maintenance decision-making based on prognostic information in the aerospace domain. Rodrigues *et al.* [153] developed an approach to select system components that need to be replaced to lead the system to a desired safety level. The method is based on the combination of system RUL and the system architecture using a fault tree analysis to obtain the failure probability. Cai *et al.* [24] presented a decision support module for aero-engine maintenance that integrates the RUL of the aero-engine and the repair time distribution to optimize the cost of maintenance of an airplane. Li *et al.* described, in [115], a prognostic-based maintenance scheduling for an Air Force fleet. The maintenance scheduling optimizer considers several inputs like the predicted aircraft health indicators, the amount of workload of each maintenance activity, the uncertainties and the remaining flying hours from the already elaborated flying hour program. Lin *et al.* developed in [117], a decision support system for aircraft fleets maintenance planning to jointly optimize the fleets availability and its maintenance cost. A novel reliability function is introduced that integrates the real-time load assessment with the current health state of the machine. The RUL is estimated from the reliability function and is used in the decision-making process by penalizing the waste of RUL. In the same context, Feng *et al.* [56], proposed a fleet maintenance strategy to minimize overall maintenance costs while meeting the mission risk requirements. The approach is based on dynamic two-level decision-making that supports a competition and a cooperative game. The RUL of each aircraft is estimated, and the respective failure probability is deduced. According to their RUL and failure probability, the aircrafts are classified into two groups: a maintenance group and a standby group. Then a cooperative game takes place between two randomly chosen members of each group to satisfy the mission requirements. The results explain how the combination of the two-level games provides an optimal solution for the maintenance problem. Luo *et al.* [124] solved the maintenance decision problem for aircraft fleets using a support vector regression method. The RUL of each aircraft is estimated to determine its reliability level. Then it is integrated into the maintenance cost by penalizing the waste of RUL. The proposed approach aims, on the one hand, at minimizing the maintenance cost by minimizing the RUL waste and at maximizing the fleet availability and reliability on the other hand.

**1.2.2.2.4/ Geographically distributed assets** Jin *et al.* [90] proposed a framework for planning maintenance activities for a geographically distributed manufacturing system. The resulting schedule optimizes the spare parts inventory, resources management and remaining useful life of the machines. Meraghni *et al.* [129] introduced a framework that provides a schedule of maintenance activities and assigns a maintenance team to each activity for a geographically distributed assets. The methods was tested on cellphone towers in which the PHM system used Internet-of-Things and cloud computing. Aizpurua *et al.* [2] offered new system-level dynamic maintenance planning based on cost-effective grouping of assets. This approach integrates prognostics information, especially the RUL of components, and takes into consideration the dynamic economic and stochastic dependencies between components. The main idea is to schedule predictive maintenance activities for critical components while run-to-failure the uncritical ones.

**1.2.2.2.5/ Railways** Camci [28] defined a new variant of the traveling salesman problem that takes into consideration the travel time between machines that need maintenance, that are modeled as cities and the time of their maintenance. This variant was tested on geographically distributed railways switches. This approach aims at minimizing overall cost by scheduling maintenance activities under the constraint of the travel time. In his later work, Camci [29] used the same principle of the traveling maintainer problem [28] with new constraints on the number of working hours of the maintenance team. In this work, Camci defined the frequency of maintenance scheduling and considered schedule changes in case of failures. In [50], the authors designed a maintenance decision support system for railways based on massive data fusion and systems engineering. The proposed system uses collected data to evaluate the degradation levels and the health states of the railways. The pattern of the degradation is then matched with historical data, and if the level of degradation exceeds a threshold, an alarm is generated and maintenance interventions are scheduled according to the operation schedule and the availability of maintenance resources. Villarejo *et al.* [180], proposed a hybrid model for fault diagnosis, prognostics and maintenance decision-making for railways system. Information is combined from expertise of maintenance workers, rolling stock data, operating conditions and infrastructure data to assess the health of the system and estimate its RUL. The resulting prognostics information is used to plan maintenance interventions to reduce the slowdowns and shutdowns of a rail track.

### 1.2.3/ Operational decisions

In this section, the works that considered operational decision-making based on the health information of the systems are studied. As mentioned, operational decisions in the PHM context can be divided into three categories: (i) production planning and mission assignment based on the RUL of the machines, (ii) logistics planning in which the spare parts ordering is optimized to avoid stock shortage and minimize the cost of storage and ordering, and (iii) automatic control decisions in which the control parameters are optimized while considering the health information of the actuator. These categories and their related works are discussed in the next subsections.

#### 1.2.3.1/ Production and mission planning

Tang *et al.* [167] presented a real-time autonomous vehicle PHM and contingency management application framework. The framework has the ability to proactively and autonomously adapt to changes in the health states of the vehicle while achieving all acceptable subsets of the mission objective. This is done by low and medium control levels (by controller configuration) and high-level mission re-planning and optimization while integrating prognostics indicators like the end-of-charge and RUL. The authors focused on mission re-planning so that when a fault occurs, the system periodically estimates the RUL values, which are then used as new constraints or additional elements in the cost function of the mission planning algorithm. Cholette *et al.* [34] implemented prognostic information in the production scheduling of semi-conductors. The solution proposed the scheduling of products between different machines by rerouting products from a degraded machine to a less degraded one to increase yield and the probability of mission success. To satisfy a production demand and maximize the production horizon, Herr *et al.* proposed different heuristics in [79]. The algorithm consists of choosing  $q$  of the  $m$  parallel machines that are capable of performing

independent and identical tasks. Their adequate running profiles are also determined to fulfill the mission (production level). Zhang *et al.* developed in [195] and [196] a mission-planning algorithm for an autonomous vehicle, that enhanced prognostics information in the decision-making process. The algorithm is based on a  $D^*$  search algorithm to find a suitable road on an unknown map. The best road was chosen gradually as the rover moved. Different objectives were considered in the objective function, including RUL optimization. Medeiros *et al.* [40] presented a task assignment algorithm that considers health monitoring information obtained by the distribution of components RUL and the system fault tree representation. The proposed receding horizon task assignment (RHTA) algorithm aims at increasing the probability of the tasks' success. Rodrigues *et al.* [154] proposed a resolution of the task assignment problem for unmanned aerial vehicle (UAV) subject to the remaining useful life of the vehicles. The modified Receding Horizon Task Assignment algorithm uses a rejection list to reduce the number of mission combinations. At each step of computation, the obtained sub problem (after applying the reject list) is considered a multiple choice multidimensional knapsack problem. Chretien *et al.* [35] proposed a post-prognostic decision framework for scheduling multi-stack fuel cell systems under service constraints. The problem considers the fuel cells parallel independent machines. It was solved using two convex resolution methods, mirror prox algorithm and adaptive lasso algorithm. Herr *et al.* developed a framework in [80] for managing fuel cell stacks to satisfy a load demand for as long as possible. The problem combined production decisions by selecting the power output of each fuel cell with the task assignment decisions by choosing which fuel cell stacks to run. The problem was then solved using mixed integer linear programming that integrated prognostics information about the health state of fuel cell which are subject to wear and tear behaviors. Skima *et al.* [162] applied post-prognostics decision-making for a conveying surface made of micro-electro-mechanical systems. The problem in this paper is choosing the proper path to transport micro-objects from a source block to a destination block in a way that maintains the system's best conditions for as long as possible. To improve the conveying surface performance, the proposed algorithm combined the use of already obtained prognostics information to maximize the lifetime of the surface and the use of travel time of the objects to optimize the utilization of the conveying surface.

### 1.2.3.2/ Logistics planning

Li and Ryan [114] modeled the deterioration of components with a Wiener process and used a Bayesian approach to estimate the distribution of the RUL. The predicted end-of-life is updated periodically by integrating the new available system data through condition monitoring. The estimated RULs define the demand for spare parts in the future. This demand is then used to order the spare parts.

To meet the delivery commitment of at least possible cost, Julka *et al.* [91] integrated the use of prognostic information in the discrete event logistics systems (DELS). The RUL was integrated into the D-SIMPAIR system to optimize the inventory and movement of spare parts by predicting the system failure date and moving the spare parts to a location where maintenance will be conducted. Based on the same context, Cui *et al.* [36], proposed a modification of the DELS. The approach is based on a two step RUL estimation in which the first estimated RUL is used to define the logistics movements according to the maintenance events activities, while the second estimated RUL is used to calibrate the fault occurrence time. The obtained system aims at arranging the spare allocations



ahead of the fault occurrence, which largely reduces the logistic delays. Lin *et al.* [119] considered the case of a group of identical machines that each contained one critical component. The degradation of these components is monitored and the remaining useful life distribution of each part is estimated. The RULs are updated at each inspection period, and the results are used to determine the distribution of the demand for spare parts in the upcoming periods while considering a non zero lead time for spare parts delivery.

### 1.2.3.3/ Automatic control

One can easily confuse the works done on post-prognostics control designs and the fault tolerant control works. In our understanding, the fault tolerant control are a designed control loop to guarantee a minimum functioning of the actuator after the occurrence of a fault [4, 14, 147, 197]. Such control loop are widely used in the aerospace technology. They are generally implemented to guarantee a minimum functioning to allow a safe emergency landing of airplanes. Hence, fault tolerant control are designed to work even when the fault occurs while prognostics-based controls are designed to avoid the occurrence of the fault by setting the controller according to the prognostics information. Readers can find more information about the fault tolerant control in these references [4, 14, 147, 197].

Bogdanov *et al.* [15], designed a linear-quadratic regulator (**LQR**) controller that has a single scalar as a parameter that establishes a trade-off between performance, the desired lifetime and the control power. The actuator in the paper is supposed to be under stochastic external load that affects its lifetime. A constraint is made on the desired lifetime, and the main work is to optimize the parameter of the LQR controller. Brown *et al.* [20] presented a methodology for designing a fault-tolerant controller that re-configures the actuators' control activity using prognostics information by trading off performance and control power. Composed of electro-mechanical actuators, the system is supposed to complete a critical mission within a time window. This defines the limits of the performance of the actuator and thus the controller has to be able to reconfigure the control activity to secure an acceptable performance level that satisfies the mission need. Later, Brown *et al.* [21] presented an approach in elaborating a prognostic-based reconfigurable control for electro-mechanical actuators. The framework is based on a PHM module parallel to the controller. The main idea consists of comparing the RUL to a desired value. If the system can secure the desired lifetime, then no action has to be taken. Otherwise, a reconfiguration is triggered by allowing new acceptable minimum and maximum tracking errors that will change the cost function of the model predictive controller (**MPC**). Consequently, the control signal will change and some set-point adjustments will be obtained. Bole *et al.* [16] developed a prognostic-based controller that will adaptively allocate the actuator's load to optimize a risk metric based on the uncertain RUL. The prognostic-based fault-adaptive controller finds a trade-off between minimizing the failure risks and the system's overall performance "without becoming overly conservative". Pereira *et al.* [148] developed a predictive control based on the prognostics and health monitoring of the actuators. The main idea is to determine the remaining allowed degradation until the end of life of the system and then distribute it uniformly on the remaining time until the next scheduled maintenance. This approach has been tested on systems with redundant actuators. In the same context, Langeron *et al.* [104], proposed a LQR controller, in which the matrices  $Q$  and  $R$  are modified with the evolution of the system degradation. Nguyen *et al.* [139] improved the RUL estimation process of the closed loop systems. The output RUL is then used to configure an adaptive controller. Vieira *et al.* [178] proved the existing relationship

between the degradation of the actuator and the loss of its effectiveness. They also used the results in the configuration of the LQR controller parameters. The results obtained in [178], were used by Langeron *et al.* in [102] to model a closed loop system that can fully implement the control law regardless of its degradation. The system is controlled by a LQR in which its parameters are adapted using the degradation level of the actuators through its RUL. Grosso *et al.* [74] proposed an improved reliability-based economic model predictive control strategy that is similar to the predictive control proposed by Pereira *et al.* in [148]. The idea consists of computing and distributing the remaining allowed degradation level on the remaining time to maintenance but with an additive dynamic stock policy in the optimization problem. The system has to guarantee the spatial and temporal re-allocation of water resources under demand uncertainty. This strategy was used on a centralized control for distributed actuators of the water network distribution of Barcelona.

#### 1.2.4/ Mixed decisions

Mixed decision-making in this work refers to works that jointly optimize operational and maintenance decisions. As noted in the previous section, operational decisions are classified into three subcategories. Therefore, mixed decisions can also be classified into three subcategories: (i) mission and production planning jointly with maintenance, (ii) automatic control with maintenance scheduling, and (iii) logistics or spare parts ordering jointly with maintenance planning. These subcategories are described in the following subsections. In addition, some works do not present a specific type of mixed decision-making, but they present frameworks to mixed decision-making on an enterprise level. These works are also discussed in this section.

##### 1.2.4.1/ Mission or production planning jointly with maintenance

Medeiros *et al.* [39] proposed a joint task assignment and maintenance scheduling for unmanned aerial vehicle (UAVs). A fault tree is used to determine the failure probability of the system based on the components RUL, while a RHTA algorithm with PHM information is responsible for the task assignments. Herr *et al.* in [81] jointly optimized the rolling stock assignment and maintenance scheduling of trains. The main idea is to assign the suitable train to each trip and minimize the waste of useful life of components while avoiding their failure by taking into consideration the predefined train timetables and prognostics information. Niknam *et al.* [142] developed a method to optimize the decision-making to control the system and schedule its maintenance. The method was tested on a variable speed wind turbine by controlling its speed and planing the maintenance actions. Pan *et al.* [145] proposed a mathematical programming formulation to solve the joint problem of production and maintenance scheduling on a single manufacturing machine. In this work, the RUL of the system is estimated and then remaining maintenance life (RML) is deduced by respecting a minimal reliability level. Once the RML is obtained, production and predictive maintenance are scheduled in a way that minimize the total tardiness of jobs and respects the RML constraint. Another contribution presented in this paper is the influence of the age of the machine on its degradation dynamic. It is assumed that after a certain number of maintenance interventions, the machine deteriorates more quickly. Wang [186] used a prognostic information based method to schedule the suitable task and preventive maintenance on a stochastic degrading single manufacturing machine. The method aims at maximizing the long-term expected profit per unit time. Fitouri *et*

*al.* [59] proposed a heuristic to solve the problem of job shop production and predictive maintenance scheduling. The approach aims at minimizing the Makespan and the total cost of maintenance based on prognostic information. In their approach the RUL of each machine depends on the task in progress. Ladj *et al.* [100] solved the integrated production and maintenance scheduling for a multi-functional single machine, to minimize the total maintenance cost, in which each job is characterized by a degradation level. The solution contains two methods mixed-integer linear programming and pro-genetic algorithms, and took into consideration the consumption of each job in terms of degradation. Later, the same authors in [99] proposed a method for jointly planning production and maintenance activities based on prognostic information in the case of permutation flow-shop scheduling problems. The predicted RUL and degradation values are associated with each machine when processing each kind of job. Uncertainties in this work are modeled through the use of fuzzy logic to model the RUL distributions. The main objective is to find the best sequencing of jobs to optimize the maximum completion time (Makespan) and the maintenance cost simultaneously. Desforges *et al.* [42] presented a generic equation to evaluate the system's capacity to satisfy the requirements of future production planning. The object oriented Bayesian network based equation includes different levels of system modeling: a functional model, a structural model and a behavior model. The prognostics of different components are used with the future load to assess the system capacity at fulfilling its future plans while suggesting which components to maintain to guarantee a predefined minimal level of reliability. Cheng *et al.* [32] proposed a method that optimizes the production, quality control and condition-based maintenance for a stochastic deteriorating manufacturing system. A periodic inspection CBM policy is used to define imperfect maintenance actions. The degradation level of the component is combined with the quality control inspection data to define the maintenance dates. While the system is producing, the degradation of its components is monitored, and the ratios of defectives from the quality control inspections are compared respectively to a preventive threshold and a quality threshold. The excess of degradation level or the ratio defect causes the system to stop for preventive or corrective maintenance. Bencheikh *et al.* [13] proposed a multi-agent-based system to solve the problem of jointly scheduling production and maintenance on a multi-purpose multi-machine workshop. Each machine is capable of different functions, and the scheduling of the production activities and maintenance interventions takes into consideration the current and future health state of the machine. In [120], the health state and the RUL of the machine are used to schedule jointly production activities and maintenance actions to improve the machine utilization, decrease the failure rate, and minimize the total costs, including the production cost, maintenance cost, downtime cost and tardiness cost. In this work, Liu *et al.* used a Weibull distribution to model the degradation of the machine.

#### 1.2.4.2/ Automatic control jointly with maintenance

Langeron *et al.* [105], presented an approach that defines operational decision-making by changing the matrix  $Q$  and  $R$  of the LQR controller and scheduling the maintenance actions based on the RUL of the system. This approach was tested on a DC motor by varying its speed and planning the replacement of its bearing. Jain *et al.* [88] established a relationship between product quality and the tool degradation level. Based on this relationship, they designed a dynamic policy for optimizing process quality control and preventive maintenance. The method integrates real-time RUL estimations. The production



process starts with its optimal pre-established parameters. Then, at each sampling time, the RUL of the tool and its health status are estimated, upon which the parameters of the production process are re-adapted to the RUL. Maintenance interventions are scheduled in parallel in a way that balances the cost of waste of remaining life and the cost of loss of quality computed through the ratio of defects. Griffith *et al.* [73] proposed a structural health and prognostic management system to enhance the use of prognostic information like health indicators and RUL in the maintenance decision-making and prognostic control jointly. The health information in this case are used to define the operation controls and the maintenance intervention scheduling.

#### 1.2.4.3/ Logistics jointly with maintenance

Wang *et al.* in [189] presented a RUL-based spare parts ordering and predictive maintenance scheduling for a stochastic deteriorating non-repairable critical system. Wiener process was used to model the degradation of the system, in which its parameters are estimated from the real-time condition monitoring data. The obtained Wiener process is used to estimate the RUL distribution that are used to update ordering and the maintenance schedule. This method aims at minimizing the expected cost rate over an infinite time horizon. Wang *et al.* in [190] proposed a prognostic-based spare part ordering and maintenance intervention for an aerospace system with a random lead time. The dynamic method consists of periodically collecting new monitoring data related to the condition of the system. Based on these data, the degradation model parameters is estimated, and the RUL is predicted. The RUL is then used to feed the optimization of the ordering date and the maintenance intervention schedule. Wang *et al.* [182] proposed a multi-spare ordering policy for complex systems with multiple continuously deteriorating components. The method consists of computing the overall reliability of the system from the components' reliability. To guarantee a predefined level of reliability of the system, some components are chosen to be maintained or replaced and their respective spare parts orders are issued to minimize the overall cost of maintenance, ordering and storage. Cai *et al.* [26] studied a joint optimization of maintenance and spare part inventory with appointment policy. The authors combined the classic CBM approach with periodic inspection of components with the estimation of the RUL to appoint spare parts to the component with an estimated RUL below a specific threshold. Once the spare part inventory is lower than a predefined security level, the spare parts orders are placed. Later, the authors of [25] considered the same approach but this time instead of using a periodic inspection policy, the inspection dates are defined according to the degradation phase of the components. In this work, the inventory is kept empty for a certain duration under the assumption that during the first part of its lifetime the aircraft does not need any maintenance interventions. Afterwards, the demand for spare parts grows gradually and stabilizes at a certain level. At this point, the inventory will be monitored and compared to a security level to place the orders.

Chen *et al.* [31] introduced a method to predict RUL of an individual component when the degradation cannot be reflected by historical data in totality. The estimated RUL is formulated into failure probability function, upon which maintenance interventions and spare part ordering are optimized jointly. Bousdekis *et al.* [17] developed a proactive decision-making framework for condition-based maintenance. This framework optimizes jointly the maintenance and the spare parts inventory based on prognostics information. The module defines the maintenance activity time by minimizing the long-term maintenance cost and the best time to order spare parts by minimizing the long-term inventory cost.

Liu *et al.* [122] developed a heuristic based on a Markov decision process to solve the problem of maintenance replacements and spare parts ordering in the case of single manufacturing machine. The Markov decision process integrated health information of the system component when this information is directly obtained from the degradation model. The method aims at maximizing the net revenue of the workshop.

Moghaddass *et al.* [133] proposed a dynamic decision policy to jointly optimize ordering and replacement dates for a single-unit inventory system. The proposed method consists of periodically collecting and observing data related to the system operations and then determining whether to start the setup of a maintenance intervention on one level. On the second level, the method is used to determine when a maintenance activity should take place. At each inspection, the current degradation level and the system health state are assessed and then the failure probability until the next inspection date is computed. This information is integrated in the decision-making process to generate warnings and schedule maintenance to minimize the long-run expected cost per unit of time.

### 1.2.5/ Synthesis

The papers analyzed for this state-of-the-art are summarized in tables in the appendix section A. All the analysis in this section are based on information included in the tables.

Considering the classification of post-prognostics proposed in the previous sections we can now draw a parallel between PHM process and system process. Figure 1.4 presents the synthesis of the PHM role in the industrial applications. The resource planning bloc is responsible for the definition of the spare parts, raw materials and tool storage and orders. The objectives bloc uses these resources to define the strategy of the system and plan the work to achieve it. Once the tasks are defined, they are transformed into a group of set points used to control the system or the actuator. The closed loop here represents a classic automatic control loop. Aside from the sensors' data inputs, the PHM process acquires the outputs of the system process like the resources status, the objectives of the machines and its control law. These data are considered in either the prognostic process or the decision-making one. Finally, the decisions are used to fine-tune the respective module to the decisions type.

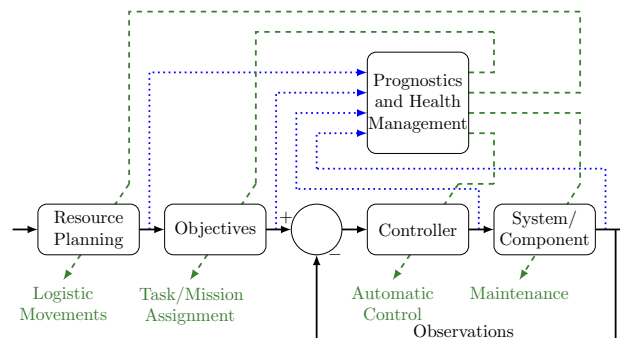


Figure 1.4: PHM place related to the system process

### 1.2.5.1/ System health state and degradation models

To describe the system's health state, the authors have used several methods. In some cases, the system's health state is described directly through its degradation level. In these works, the authors choose several degradation models for the components/system. The most common ones are: (i) failure probability distributions (e.g. Weibull distribution), (ii) stochastic processes (e.g. gamma process or Weiner process), and (iii) physic-based models (e.g. Paris equation for crack growth). In other cases, the health of the system is described directly through modeling its different observed status using data-driven methods (e.g. machine learning classification methods), system design methods (e.g. Petri-nets, fuzzy logic), or the different varieties of Markov processes. However, several works omitted the used method to model the system's health states or mentioned that the method is data-driven without specifying the method type. Few works supposed the existence of a black-box prognostic module that provides them with the health indicators and/or RUL without specifying details about degradation models, evolution, or nature.

### 1.2.5.2/ The integration of prognostic information in decision-making

After analyzing the literature works on post-prognostics decision-making, we notice that these works dealt differently with integrating the prognostic information in the decision process. We propose to classify this integration into four categories:

- **As a classification criterion** This is the most common way of integrating the prognostic information in the decision process. The RUL and/or the future health state of the system (e.g. future degradation level) are used to select an option from a predefined range of decisions. For example, in [95], the authors supposed that maintenance activities can only be executed during fixed predefined time windows. Therefore, they used the estimated RUL to select the suitable maintenance opportunity by simply comparing the RUL to the date of the maintenance window. In other works, the prognostic information are used to directly decide the maintenance date after comparing the RUL to a predefined threshold.
- **As a constraint** In this case, the prognostic information are used to validate the proposed solutions by the decision-making algorithm. The works that integrating PI as constraints generally assume that the decisions influence the system's state differently. For example, they assume that each task has its degradation rate or consumes a known duration of the RUL. In the work of Herr *et al.* [81], each trip has its degradation rate on the system. The authors implemented a constraint in the proposed linear program to ensure that the sum of the degradation rates for a train remains lower than a threshold at all times.
- **As a term in the objective function** The objective of PHM is to make full usage of the system or/and its components while avoiding its failure. Therefore, the second most common way of integrating prognostic information is by adding it in the objective function. These works generally try to minimize the overall cost of implementing solutions. Therefore, it is common to put a penalty cost on the considered lost time of RUL. For example, Camci has a habit of integrating the prognostic information in the to be minimized cost function like in [28]. It is also possible to use an objective function that depends on the RUL in a way that maximizes the RUL value after the

decision-making process. This is common in studies of the battery state-of-charge in which the objective is to maintain the battery charge as long as possible. For example, Zhang *et al.* [196] proposed an algorithm to find the optimal path that maximizes the remaining charge in the battery of the rover.

- **As a decision variable** It is less common to use prognostic information to compute future decisions. In this case, this information construct the future decision instead of evaluating it. For example, in [148], the authors used the remaining degradation rate authorized for the actuator to define its control parameters. Almost all the works that dealt with the reconfiguration of the controller's parameters or that proposed health aware predictive controllers used the prognostic information as decision variable. They directly implemented the RUL or the future state of the actuator in the formula that updates the controller's parameters.

Several works have combined two or more of these categories in their decision-making process. The combination are mainly parallel in which RUL and/or HI are implemented as constraints and are counted for in the objective function. However, in some few cases the prognostic information is used as a constraint then as a classification criterion in a hierarchical way. For example, in [144], the authors used the component's RUL to optimize the maintenance plan for each subsystem individually (on a component level) then the maintenance scheduling for the overall-system by regrouping subsystem's maintenance activities. The prognostic information is then used to select the suitable maintenance opportunity (on a system level).

### 1.2.5.3/ Recurrence of RUL estimation

As defined, the post-prognostic decisions are a sequence of actions built based on the assessed current and estimated future state of the system and/or its components. This implies that these decisions are constructed by integrating the prognostic information obtained from the analysis phase. In the previous paragraph, we explained the various methods used to integrate the prognostic information in the decision process. However, one might wonder how many times are the prognostic information estimated for a single decision-making run. We noticed through the most of the literature works that the authors assumed the existence of a prognostic algorithm that provides them with the RUL of the system and/or its components, future trends on the system health indicators, or the failure probability of the equipment. However, the authors failed to mention how many times is the prognostic algorithm sought for a single decision cycle. With the analysis of the paper, we discovered that usually the PHM process is considered in a sequential manner. In other words, the analysis phase finishes by estimating the RUL and/or system's future HI. This information is feed to the decision process which usually determine the future schedule of the system for a long horizon. Therefore, several works proposed decisions based on a single RUL estimation. After the first maintenance intervention, the authors, usually, use reliability rules on the remaining decision horizon to model the system behavior. However, one can find works that used several prognostic values to build their decision. For example, Herr *et al.* [79] proposed that each task has a degradation rate over the system. The authors supposed that the prognostic algorithm provides them with the degradation rates of all possible combinations of (mission, profile). While this assumption seems more realistic, the authors assumes that the degradation model of the systems is linear and that the order of missions does not have an influence on the degradation's evolution. The

works on real-time decision-making, particularly when adapting the controller's parameters are considered based on a dynamic RUL estimation. Finally, Zhang *et al.* [196] used a real-time decision-making process in which the RUL of the rover is estimated several times while considering the possible paths. Therefore, the rover paper is considered also as a dynamic RUL estimation.

#### 1.2.5.4/ Objective and resolution methods

We defined the decision-making problem as an optimization problem. Therefore, we analyze the literature works from an optimization point-of-view. First, let's focus on the objective of the optimization problem. The analysis of PPDM works suggests that we categorize them into three categories according to the objective of they optimized:

- **Cost oriented** The PHM framework is used to minimize the cost of maintenance and/or operational activities or to maximize the profits. Most commonly, we can find the minimization of the total or long run cost, and the maximization of benefits or profits.
- **Reliability oriented** In this case, the authors try to maximize the availability, reliability and operational safety of the components. Works with reliability oriented objectives usually are either maintenance decision-making or mixed. Very few authors used reliability objectives to solve operational decision-making problems. One can cite the work of Herr *et al.* [79] in which the authors proposed an integer linear program to maximize the production horizon of the machine.
- **Task oriented** In these words, the task becomes the objective of the decision-making process. The prognostic information is used to maximize the likelihood of the mission's success or to optimize an attribute of the mission for example its time or quality. RUL and/or HI are also used to maximize the performance of the system when achieving a certain task.

To satisfy these objectives, the authors proposed several optimization methods: simulation based methods, exact methods, heuristics, and meta-heuristics (principally genetic algorithms).

### 1.3/ Open issues and problem statement for the thesis

As one can notice in the previous works, none of these works has compared their results to the one obtained by previous works. This is a major drawback in the post-prognostics decision-making community. It is slowing the development of this field and preventing its maturation. The various types of objective functions and the way they are defined are also preventing this comparison from happening. Thus, it is difficult to position one's work to the current research in the domain if the chosen objective function is not the same. This explains why most works in the field of post-prognostic decision-making lack positioning. Most authors prefer to compare their results to other maintenance strategies, namely, systematic preventive maintenance, cyclic maintenance and corrective maintenance. Therefore, there is a need to define some common metrics to evaluate decisions and rank research according to these metrics. Ranking the works in the field

by different metrics would help identify the gaps in the research and provide support for future work. Thus, the post-prognostic decisions will evolve rapidly. However, defining such metrics is hard due to the lack of some information about the decisions in the literature. Moreover, only few works provide full information about their decision-making process and performances, such as:

- the execution time of the algorithm,
- the quality of the provided decision,
- the effects of RUL uncertainties on decisions and how they are expressed,
- the type of decision-making process (i.e., online or offline algorithms),
- the frequency of executing the decision-making process,
- the duration of the decision horizon, and
- the consequences of the decision on the different work processes, mainly the load of the maintenance shop.

The presented works contain a serious lack of information. As most of them do not present the used assumptions. They also deliberately omit the used databases, parameters' values, and the configurations of the algorithms. It is almost impossible to reproduce any of the results presented in their papers.

### 1.3.1/ Decision-prognostics interactions

In literature, works on PHM have usually been studying the analysis phase in general and the prognostic algorithms in particular. Recently, the decision support phase have gained a considerable attention of the scientific community. However, it seems like the works on post-prognostic decisions are a little detached of the previous phases. In this context, researchers always assume the existence of the prognostic algorithm without discussing it. Some works do not even mention it. This could give the impression that the PPDM is just a special form of general optimization triggered by the RUL. In addition, most works solve their decision problem using one estimate of RUL and/or HI over a horizon of a duration that could include several maintenance interventions. The common method consists of making the first decision based on prognostic information then assuming reliability rules or failure probability for the rest of the horizon. Another major issue with the PPDM is using deterministic RUL and/or HI values. As we discussed earlier, prognostic modules provide a distribution of the system's remaining useful life. However, in the decision-making process, authors use the RUL a deterministic value without explaining how they obtained this value. Moreover, they generally assume that the system's future conditions are constant during the decision horizon. This implies a highly controlled environment of the system which can only be achieved in a laboratory.

Even though prognostic algorithm are extensively studied for various applications and under various conditions, it is not common to find works that included the future loads or system's conditions in the process. Actually, except for works that proposed physic-based degradation models, almost no other works considered adding future loads into the model. The authors always tend to assume that either the system is doing the same exact tasks



without any variability or these future loads constitute a portion of the uncertainties. Both Daigle and Goebel [37] and Zhang *et al.* [196] integrated information about the future loads of the system in their model-based prognostic methods to estimate the RUL of the system. In [196], the resulting RUL was more involved in the decision-making process by using it as a variable in the optimization objective function. Welz *et al.* [191] integrated maintenance information into the prognostic process. Maintenance actions, whether perfect or imperfect, were modeled and then used as inputs to the prognostics. The approach was tested for the Weibull method and the general path model (GPM) method. In both, the integration of the maintenance model improved the accuracy of the prediction by reducing the prediction error. The approach was validated on a heat exchanger test bed. Vileiniskis and Remenyte-Priscott [179] developed a new approach for predicting the quantitative risk of failure. The method consists of creating a Petri-net model to present the current state of the system, the degradation and the future operation and maintenance activities. The model is then run in a Monte Carlo simulation to obtain the statistics of components' performance over a selected horizon. The results are fed into a bow-tie model to estimate the risk of hazardous events. In [160], authors used future power consumption profiles in the prognostics algorithm to predict the end of discharge of a Lithium-polymer battery in a rotatory-wing UAV. While in [46], the authors considered that the variance in future loads are included in the uncertainties.

One can notice a big gap between the two process of the PHM framework. This allows us to conclude that the PHM process as presented by the OSA-CBM was studied as a sequential process for a long time.

By analyzing the papers that treated post-prognostic decision-making and those that aimed to integrate future loads in the RUL prediction, we can easily notice the important interactions between the prognostics and the decision-making modules. In Figure 1.5, the prognostics module influences the decision by the value of the RUL, and the decision-making process modifies the prognostics outcome with the future loads and the selected decisions. This can drive toward the necessity of integrating the prognostics and decision processes into a closed loop. One proven advantage of such an approach is the improved accuracy of the RUL prediction.

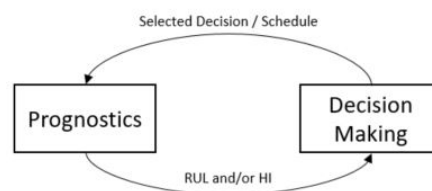


Figure 1.5: The interactions between the prognostics and the decision-making modules

Naturally, decisions are build to be applied to the system. This application could alter the system's health evolution. For example, Frost *et al.* [62] used the health state of wind turbine blades in an adaptive controller to extend the operating time of the system. When the evolution of the system degradation and/or health state is different from the one used to build the decision, this could make the decisions obsolete. Therefore, to best fulfill the objective of the framework, the decision process must be adapted to include the current prognostic information. However, none of the literature works have discussed the need of re-applying the PHM process nor studied the frequency at which the process will be re-initiated.

In the same context, decision application could cause the system to operate in new unknown conditions. In this case, the obtained data from the health state evolution could be new to the prognostic algorithm. Thus, it would be important to have a prognostic module that is updated to the real conditions of the system. As it has been proven, degradation data are highly related to the operation conditions of the system. For example, in the data generated by PRONOSTIA [137] for rolling bearing, the speed of rotation has a major influence on the degradation profile. Therefore, using data from historic degradation can be a good starting point for the prognostic process. But if such a prognostic process is still in use under new operating conditions, it will lead to questionable outcomes. Therefore, there is a need to incrementally update the process. However, this point is also omitted in most literature work on prognostic algorithms.

### **1.3.2/ Toward decision-enhanced PHM**

#### **1.3.2.1/ Issues to be addressed**

Although in recent years, a great number of post-prognostic decision-making methods have been proposed for different application areas, the progress to build an effective and efficient approach is still limited. On one hand, usually, the use of prognostic information in these approaches is limited to a basic definition of constraints or just as a launcher of the decision process. Moreover, the uncertainties presented in the prognostic algorithm's output are ignored, and a deterministic value is used in the decision process. On the other hand, generally, prognostic algorithms do not take into consideration the future loads into the estimation phase. Frequently, the future loads are either replaced with the mean of conditions or just supposed to be included in the uncertainties. Obviously in this classic way of applying PHM, we can note that the processes are executed in a sequential way with a simplified interactions. In such situation, we can safely assume the PHM framework as an open-loop process. Moreover, works that integrated prognostics and decision-making presented some lack of information. Some points that we qualify as important have been omitted from the works. Mainly, the dynamic of estimating the RUL and health indicators, dynamic of the overall framework, and the reaction of the framework face to new system's conditions. According to these discussions, we summarize the key issues we intend to solve as follows.

- How to emphasize the relationship between the prognostic and decision-making modules?
- How often should the PHM process be launched?
- How long ahead should the decision-making module plan?
- How to better/fully use prognostic information in the decision process?
- How to clarify the applicability of the PHM framework?
- How to make the PHM framework robust to the system's conditions changes?

The issues highlighted above confirm the need to enhance decisions in the prognostics and health management framework. Therefore, the main assumptions, objective and contributions of this thesis are presented in the following sections.



### 1.3.2.2/ Assumptions

The works presented in this thesis aim to upgrade the existing prognostics and health management framework. However, the proposed contributions are valid for a certain parameters defined by the following assumptions.

- The data from the system sensors can be processed and allows to obtain suitable health indicators.
- The system health state and degradation are supposed to be observable to a certain extent. Meaning, that from sensors acquired data, one can estimate with a measurable confidence the actual health state or degradation level of the system.
- The degradation of the system is mastered to some extent. In other words, one can model the degradation through physics-based models, stochastic models, data-driven models or hybrid models
- The matter of adequate prognostic methods is not addressed in this paper, under the assumption that adequate methods exist for short and long term prognosis.

### 1.3.2.3/ Objective and contributions

To answer the aforementioned challenges, the purpose of this work is to provide a new adaptation to the PHM framework to upgrade it to a closed-loop process. The proposed framework allows the construction of decisions while integrating their effects on the system's health state evolution in the remaining useful life estimation process. The interactions between prognostics and decision-making are modeled by building decisions iteratively over the duration of the considered horizon. Also, the data used as a reference for the prognostic module are refined to include new variation in the system behavior to guarantee a consistent prognostic method towards systems' changes. The Focus of the adaptation of the framework is on the decision-making process to improve the integration and the quality of the prognostic information. Therefore, in this thesis, we focus more on post-prognostic decisions, but some elements of thoughts are proposed for the prognostic part. According to that, the main contributions of this thesis are as follows:

1. State on prognostic and health management challenges (Chapter 1).
2. Review of post-prognostic decision-making (Chapter 1).
3. Propose a new post-prognostic decision framework (Chapter 2).
4. Initialize the new framework on two study cases (Chapter 3 and Chapter 4).
5. When to use PHM methodology for a single multi-purpose machine (Chapter 3).
6. The influence of decision horizon on decision methods (Chapter 4).

## 1.4/ Summary

This chapter presents a summary of the PHM challenges. A thorough survey on post-prognostics decision-making and its growing interest as a strategy to exploit prognostic

information to maintain and manage life cycle of critical machinery. According to that, a detailed classification of these works is presented. The issues and omits of the previous works have been pointed out. Following that, open challenges of the decision process are defined and the requirements are discussed. Two major challenges of post-prognostic decision-making are: (i) the interdependency between the prognostics and decision-making modules and (ii) the study of parameters that influence the execution of the PHM process and their implementation on real world applications. The objective of this thesis is to propose upgrades to the existing PHM framework by emphasizing the post-prognostic decision process and the prognostic-decision interactions. All the majors parameters for the proposed post-prognostics decision framework (such as decision horizon, decision actions, ...) and their influence are described. The next chapter describes the modification performed on the existing PHM framework to enhance the decision-making process. The following chapters contain the initialization of the post-prognostic decision framework on two study cases.

## Chapter 2

# Decision Enhanced PHM Framework

### Contents

---

<b>2.1</b>	<b>Systems and applications in PHM context</b>	<b>40</b>
2.1.1	Systems	40
2.1.2	Applications	43
<b>2.2</b>	<b>Main Concepts</b>	<b>44</b>
2.2.1	Elementary actions	44
2.2.2	Local decisions	47
2.2.3	Global decisions	50
2.2.4	Estimators and Decisions Builders	52
<b>2.3</b>	<b>PHM framework adaptation</b>	<b>54</b>
2.3.1	Decision building loop	55
2.3.2	Decision applying loop	57
2.3.3	Information/data loop	58
2.3.4	Overview of the Proposed Framework	60
<b>2.4</b>	<b>Metrics and parameters</b>	<b>61</b>
<b>2.5</b>	<b>Summary</b>	<b>61</b>

---

The PHM framework was developed to organize the methodology of studying the degradation and health of a given system, estimating its remaining useful life and/or the evolution of its health conditions, and managing its operations and maintenance activities. Based on the presented PHM framework, the decision-making process is based on the estimated predictive information. Thus, decisions are highly dependent on the future information. Besides, the estimated values are based on certain assumptions of the future conditions and loads. Therefore, the decisions that are scheduled for the future would influence the estimations. Thus, the prognostics and decision-making modules are highly inter-dependent. However, when reading the literature works on post-prognostic decision-making these interdependency is omitted. Moreover, several omissions are present in the post-prognostic decision-making process. Most important issues are: (i) the interdependency between the prognostics and decision-making modules and (ii) the frequency of execution of the PHM process. This chapter presents the modifications applied to the PHM framework to emphasize the decision-making and prognostics interactions. Some essential concepts to the framework re-adaptation are defined following an organization point of systems and applications in the PHM context. The underlying idea is that, prognostics should take into account future decisions and operational conditions on one hand. On the other hand, decision-making process should further integrates prognostics information. For this purpose, three loops are introduced to the existing PHM framework.

## 2.1/ Systems and applications in PHM context

Prognostics and Health Management is a general framework and applying in a certain case, demands a thorough study of the system and the scope of the application. Since the great variety of application in the PHM context, the definition of a generic approach requires a good formalization of applications. Therefore, in order to better define the re-adapted framework for Prognostics and Health Management, we first have to formalize the definition of systems and applications in a PHM context and their characteristics and specifications.

### 2.1.1/ Systems

In this section, we will present the definition of a system in the context of PHM based on the most common assumptions used in literature. A system in PHM is expressed in form of a set of descriptors, where each descriptor is defined by a pair an attribute and its value. The most common attributes and their corresponding values in literature are :

- **Type** Each system has a particular nature that depends on the field it has been designed to be used in. The nature of systems can be classified into; manufacturing, distribution, energy, transportation, aerospace, autonomous vehicles, medical and bio-medical.
- **Typology** Systems can be composed of one machine/actuator or many machines. Thus, the typology of a system will define its structure, whether it is a single machine system or a system of systems. In the case of multiple machines systems, there is a need to define the typology that exists between the different machines. The machines have two possible dependencies either economic (i.e. grouping machines

in applying actions could cost less than acting on each machine alone) or structural (i.e. some machines are dependent on previous ones like for example in a production line where machines are structured in a serial way). In other words, machines can be economically dependent or structurally dependent in a serial, parallel or mixed typologies. To summarize, a system can be composed of (1) a single machine or (2) multiple machines. In the later case, the machines can be (a.) economically dependent, (b.) structurally dependent in a (i) serial, (ii) parallel, or (iii) mixed typology, or (c.) both dependencies are considered. Typology also defines if the machines are geographically distributed or they are placed in the same facility.

- **Granularity level** This characteristic is defining the composition of the machine. Whether the failure of the machine is caused by a single component, or many components can cause the failure of a machine. The granularity level can be different from one machine to another in the case of multiple machines systems. A machine can be simple if its degradation or RUL is determined by a single component, or complex if the health of the machine depends on multiple components state.
- **Dependencies** In the case of complex multi-components system, there might exist some dependencies between the components. Nicolai and Dekker in [141] defined the possible three types of dependencies between components; (i) Structural dependence is when a component stops due to maintenance or failure all the dependent components stops. (ii) Economic dependence signifies that maintaining dependent components together will cost less than maintaining them separately. (iii) Stochastic dependence is when a component fails/degrades causes the failure/degradation of all dependent components. One can also finds cases where the components present a different combination of the three types of dependencies.
- **Complexity level** One can finds a lot of definition for the complexity level of a system in literature, but no consensual definition has yet been proposed. The use of the term "complex system" in literature may indicate a system with multiple components that presents a different kind of dependencies [173], [76] or the models used for the systems behavior and evolution are non-linear and complex [60] or the combination of both multiple component and non-linear evolution with multiple pathways [192]. One can finally conclude that the definition of the a complexity metric is a difficult operation, but we can conclude that the complexity of the system depends on its typology, granularity level, dependencies and degradation parameters.
- **Criticality** It characterizes how critical the system is. The criticality of a system is determined on a design level of the system. It is influenced by the redundancy of critical machines/components, the effects that might be caused with a failure on the systems environment and users safety and the safety coefficient of the system.
- **Observability and metrics** Observability here is defined as in the control theory. If the systems internal states can be measured or determined from its external outputs. A system can be fully, partially or non-observable. Metrics design the outputs of the system and/or the sensors data that monitors the degradation of the system and its performance.
- **Degradation parameters** It concerns the systems degradation model. In this field, we describe: (i) The systems degradation forms or type, example a crack or wear, (ii) The causes of degradation that can be classified into two factors; environmental causes that describes the machine environment i.e. the external sources like

temperature, humidity, etc... and operational causes that describes the operating mode of the system i.e. the internal sources like rotation speed, the load, etc... (iii) The degradation model whether it is a physics-based model for example the use of Paris' law for modeling crack growth, a data-driven model for example the use of neural networks in the prediction of aircraft actuators RUL [23], or a hybrid model for example the use of process gamma as a degradation model where its parameters are found from previous degradation test. (iv) The consequences of the degradation, which describes the level of the loss of operational functions.

- **Overall degradation dynamics** This descriptor defines the speed of the machines degradation. It is influenced by reliability information like the mean time between failures of each of its components, the necessary time for the system to move from one health state to another, or the lifetime of the components.
- **Action variables** This field is used to describe the possible variables that can be used to manage the systems state. These variables can influence the evolution of the degradation of the system and they are used to define the operating mode of the system for example the definition of the rotation speed, or the weight of the loads to move.

**2.1.1.0.1/ Example: CNC Machine** Nowadays, machining equipment such as the computer numerical control (CNC) machine tool or CNC machining centers are increasingly required with the growing market demand in term of machined surfaces and products assortment. These machines require a high level of reliability. Therefore, we will present here the example of a CNC machine system in the context of PHM. CNC machines are made of a set of mechanical subsystems that can face different kind of degradation. A decrease in the quality of the machined surface (the output of the CNC machine) can be caused by different source of degradation. For the sake of simplicity, in this example we will consider that the CNC machine degradation can be caused by three main components; the spindle, the rolling bearing of the spindle and the cutting tool. These components presents different kinds of dependency, but in this case we will only consider structural dependency to keep the example simple. Ding et He in [43], defined that the failure of the cutting tool is a stochastic process caused by the cumulative wear, that depends on the machining conditions, such as the cutting speed, the feed rate... Soualhi et al. [164], Dong et Zhang [47] used the same assumptions about the causes of failure of the bearing and the spindle and the factors on which the degradation depends. Therefore, we will consider that the degradation of each of these components can be modeled by a gamma process degradation model, where the proposed model will be influenced by different factors, environmental factors like the temperature where the machine is installed, the humidity and factors related to the operating mode like the cutting speed, the feed rate, and the characteristics of the product (material, geometry, depth of the cut, ...). Previous studies on the degradation of the considered components shows that it is difficult to observe the exact state of the components, this implies that the CNC machine system is considered as a partially observable system. Moreover, the metrics that allows to observe the state of the machine are the motors electrical measures for the degradation of the bearing and the spindle, vibration measures for all parts degradation and acoustic emission and video microscopy for the cutting tool wear. As to observe the performance of the machine, we will monitor its production rate, the maintenance time and its utilization.

### 2.1.2/ Applications

Similar to the previous subsection, we will present how works in literature detailed the application. Like systems, applications can be defined as a set of pair an attribute and its value. The most commonly used pairs are the following :

- **System** In this field, we will define the system used in the application. The system here will be defined as in subsection 2.1.1.
- **Objective** This attribute describes the main objective of integrating PHM in the context of the presented application. The mission of the application will be mathematically formulated into the objective function of the decision making problem. In literature, one can find a high variability of missions but one can classify them into three categories : (i) *Availability oriented*; where the purpose of the PHM process is to maximize the reliability and the availability of the system. Example the maintenance optimization of railway systems including rolling stock, rail track and rail infrastructure. (ii) *Cost oriented*; where the decisions are made in a PHM framework in the intention of minimizing operational and maintenance costs. Example the work of Camci [27] on the optimization of the maintenance cost of a manufacturing system. (iii) *Assignment oriented*: where the main objective is to secure the success of the targeted mission or an acceptable part of it. Example the framework proposed by Zhang et al. in [196] to plan the road of an autonomous vehicle in a way to succeed in reaching the target based on the battery State-of-Charge.
- **Performance indicators** Performance indicators are metrics related to the application mission. They allows us to evaluate the objective function of the decision making process, compute the deviation from the aimed value, and thus give us an idea about which kind of actions to take to reduce the deviation and consequently fulfill the applications' mission.
- **Decisions type** This attribute describes the possible actions to take in order to fulfill the purpose of the application. Decisions can be of: (a) *maintenance nature*: by setting the part of the system to maintain and at which date, (b) *operational nature*: this category can be subdivided into three subcategories : (i) *Control decisions* where we can modify the set-point or the reference value of some actuators or even modify the parameters of the controller. (ii) *Task assignment* by assigning which part of the system to achieve which tasks this includes also the production scheduling. (iii) *Supply chain management* by defining the logistic movements of spare parts, raw materials, work-in-process and final products. (c) *Mixed decisions* where we can find different combination of the previous two types of decisions. Also this attribute define the elementary actions of the possible decision in relation with the action parameters defined in the system.
- **Decision horizon** The decision horizon is defined as the period of time on which the decisions will be made. This feature is used to define the type of the horizon and its duration. Decision horizon can be of two types: (i) *Rolling horizons* Decisions here are made over a certain period of time with a fixed duration. Then with the evolution of the system, the decision horizon is shifted and decision are built on the new interval of time. Such a type of horizon implies that the decision making process is periodically executed. (ii) *Fixed horizons*: The process of decision making, in this case, is only executed once or is re-executed if the application requires it at



the end of the first decision horizon. In context of PHM, this implies that the PHM process will be executed only once, and decisions will be made based on a single RUL estimation. Thus with such type of horizon we will lose the dynamic aspect of the RUL estimation.

In a classic PHM application, the decision horizon value is set by the application expert or by the traditional functioning of the system. In our framework, we will take the value provided by the systems' owner as a start point then we will try to optimize the duration of the decision horizon.

- **Discretization Period:** The decision horizon is divided into smaller time units. These time units are called periods. They can be a day, a week or a month depending on the application specifications and constraints.

**2.1.2.0.1/ Example: Numerical Machining Workshop** In this example, we are going to optimize jointly the maintenance and production schedule of an industrial CNC machining system. The applications' mission is to find the best compromise between producing with different cutting speeds and maintaining the machine in order to maximize the workshops' benefit. Therefore we are going to monitor the cost of operations and maintenance interventions, plus we will compute the utilization of the machine.

## 2.2/ Main Concepts

The aim of this work is to enhance the process of decision-making in the prognostics and health management context. The decision-making has been defined as the act or the process of making choices after thinking of several possibilities [181]. Therefore, one can wonder how do we define a possibility or a decision in the PHM context. In this section, we introduce some essential concepts for the development of the enhanced-decision framework.

### 2.2.1/ Elementary actions

In a PHM context while stating the decision problem, there is a need to define the possible actions that can be applied. In a general context, the possible actions form the possibilities from which we will make a choice. These actions are called, in this context, **elementary actions**. One can define an elementary action  $\alpha_i$  as a countable set of descriptors  $d_{i,j}$  (attribute  $a_j$ , value  $v_{i,j}$ ). Therefore, one can write an elementary action as:

$$\alpha_i = \{d_{i,j} = (a_j, v_{i,j})\}_{j=1}^{Card(\alpha_i)} \quad (2.1)$$

For every application, there is a countable set of elementary actions. Therefore the following notation is used:

- $\mathcal{A}$ : The countable set of elementary actions for a specific application.
- $Card(\mathcal{A})$ : The number of elementary actions.



$\mathcal{A}$  is defined as:

$$\mathcal{A} = \{\alpha_i\} \text{ with } i \in \{1, \dots, \text{Card}(\mathcal{A})\} \quad (2.2)$$

Let us consider the case of the CNC-machine workshop. In the example, we intended to study mixed decisions i.e. maintenance and production scheduling. Therefore, the elementary actions, in this case, can either be a maintenance actions or a production task.

Let us start by detailing the example of a maintenance elementary action. An elementary action of maintenance has a targeted part of the system. For example, the replacement of the gear box. This specific action has a predefined duration and a required set of skills i.e. the technician need to be a mechanic. In other cases, like checking the accuracy of the machine requires a control technician. The maintenance elementary action of the gear box's replacement can be summarized in the (descriptor, value) style as in Table 2.1.

Table 2.1: Example of a maintenance elementary action

Descriptor	Value
Type of action	Maintenance
Nature of action	Replacement
Concerned system/component	Gear Box
Spare Part Ref.	*ref. of the spare part
Required technician	Mechanic
Duration	2 hours

Elementary actions can also be operational. In this given example of the CNC machine, we can see the production task of a certain order as an operational elementary action. The definition of the production task would require the order in question, some of the order's characteristics e.g. if it is urgent and the deadline date, the nature of the raw material, some specifications about the machining process like speed and depth of cut, etc. An example of the production task definition in the style of (descriptor, value) is given in Table 2.2. This is an illustrative example and the set of used descriptors is not exhaustive.

Table 2.2: Example of a production elementary action

Descriptor	Value
Type of action	Production
Order	Order no. 15
Urgent	No
Deadline	Due in 10 days
Quantity	150 pieces
Raw Material	Steel
Depth of cut	1.5
Feed rate	0.5
Rotation Speed	1500 rpm
Duration	4 hours

Therefore, one can conclude that attributes can be different from one activity to another and also from one application to another. Given the examples above, we can give some examples of attributes. These attributes are divided into two categories according to the type of action. The attributes for maintenance are defined as follow:

- **The type of action:** This field is used to specify the type of action. In this case it is a maintenance activity.
- **The nature of the action:** In maintenance context several actions are possible like replacement, refill, inspection, and repair. It is important to specify the nature of the action intended by the elementary action.
- **The concerned system, or component:** Here, one defines the part of the system concerned by the proposed action. In the case of multiple levels of granularity, this descriptor can have also multiple level. One can imagine a descriptor for the concerned machine, one for each level of subsystem, and one for the lowest level i.e. the component reference.
- **The requirements of the action:** This attribute must define the different requirements of the described action. It could be divided into several attributes to accurately describe the action. In the given example, this attribute regroups the *spare part ref.* and the *required technician* attributes.
- **Duration:** The duration of the action needs to be defined or estimated if no exact value can be known.

We give also some examples of the attributes that can be used to define production elementary actions. These examples are as follow:

- **The type of action:** In this case it is a production task.
- **The concerned system:** In case of multi-machines application, this attribute is quite handy to define which machine is scheduled for the task.
- **The task:** In the case of mission assignment, production scheduling, or supply chain management, the task to execute, or the job the schedule for production is defined here.
- **The task's requirements:** This represents a set of attributes that describes some of the tasks requirements for example, the quantity to produce, or the details of the mission. It can also describe the urgency of the task and/or its constraints. In the given example, the *urgent*, *deadline*, *quantity*, *raw material* attributes are part of this task requirement subset.
- **The control loop parameters:** In the case of automatic control, this field is used to define the values of the control loop parameters which could be the new set points or the controller parameters. For example in [104], this attribute corresponds to the new values of the Q and R matrix of the used LQR controller.
- **The new operational parameters:** In case there is a need to change some parameters of the system like its speed, one can modify the set-points of the controllers and therefore the new reference values are defined here. In general, the new operational parameters are basically used to describe how the task will be performed. In the given example, we can say that the *depth of cut*, *feed rate*, and *rotation speed* are all attributes that describe the new operational parameters. They describe the operating profile.

- **Duration:** The duration of the action needs to be defined or estimated if no exact value can be known.

One can notice that some attributes are common no matter what the type of action is. In general, the elementary action can either be an operational decision as in automatic control, scheduled production order, assigned mission, supply chain management action, or maintenance of a component, subsystem or the whole system. Therefore, the type of the action is always required. The elementary actions are going to be used to build a schedule for the application. Since almost all schedules are time limited, the duration of an elementary action is an important attribute of the action and should be defined for all type of actions.

Moreover, the definition of the suitable attributes is an important step in the process of implementing the framework. This task is quite dependent of the system and the considered application. A good definition of the attributes results in a good definition of the elementary actions. Based on these actions, decisions are made. But before making decisions, these actions are transformed into local decisions. Let us start by defining a local decision.

### 2.2.2/ Local decisions

The local decision represents the elementary action execution on the system. In other words, an elementary action is transformed into a local decision by estimating its outputs in terms of the system's health state, time evolution, and the application's objectives. A Local decisions  $e_i$  is defined with a countable set of descriptors  $d_{i,j}$  (attribute  $a_j$ , value  $v_{i,j}$ ). Local decisions are a transformation of elementary actions. Therefore, they inherit the descriptors of the elementary action. Moreover, other descriptors are needed to define the transformation (scheduling) of the actions. In general, the added descriptors can be classified into three categories: (i) schedule-related in which the time and resources constraints are defined, (ii) health-related descriptors that contains prognostics information about the systems health state evolution, local prognosis, reliability attributes, etc, and (iii) application-related descriptors in which financial information are defined such as costs and benefits also some optimization objective related attributes could also be defined such as the tasks success probability.

So a local decision  $e_i$  can be written as:

$$e_i = \{d_{i,j} = (a_j, v_{i,j})\} \text{ with } j \in \{1, 2, \dots, \text{Card}(e_i)\} \quad (2.3)$$

We suppose that for any studied application there is a finite number of the local decisions, thus we note:

- $\mathcal{E}$ : The countable set of local decisions.
- $\text{Card}(\mathcal{E})$ : The number of the local decisions in set  $\mathcal{E}$ .

Therefore,  $\mathcal{E}$  is defined as:

$$\mathcal{E} = \{e_i\} \text{ with } i \in \{1, 2, \dots, \text{Card}(\mathcal{E})\} \quad (2.4)$$

For most of the applications, the set of feasible local decision is time-sensitive. The set of feasible local decisions at instant  $t$  is defined as a subset of the set of possible decisions  $\mathcal{E}$ . The feasibility of local decisions is defined by the time constraints and the local prognosis of the elementary action they represent. This point is more detailed in section 2.2.4. One can note:

- $\mathcal{F}(t)$ : The countable set of feasible local decisions at instant  $t$ .
- $Card(\mathcal{F}(t))$ : The number of feasible local decisions in set  $\mathcal{F}(t)$

Therefore,  $\mathcal{F}(t)$  is defined by:

$$\mathcal{F}(t) = \{e_i\} \text{ with } i \in \{1, 2, \dots, Card(\mathcal{F}(t))\} \quad (2.5)$$

With:

$$\mathcal{F}(t) \subset \mathcal{E} \quad (2.6)$$

Let's go back to the case of the CNC-machine workshop. In the previous section, we gave the example of two elementary actions: a maintenance activity and a production task. Now, let us suppose that these actions are transformed into local decisions at instant  $t$ . In the following, we present an example of maintenance and production local decisions.

For simplicity purposes, we consider that the CNC machine is composed of only two components. Moreover, maintenance activities of replacement nature are considered to be perfect. In other words, replaced components are supposed to be in a new health state with a degradation level equal to zero. In this example, we consider the replacement of the gear box. The related elementary action is defined in the previous section. Here, we consider the scheduling of this action and therefore its transformation to a local decision. This transformation requires the definition of the start time, the allocation of the required resources, the estimation of some health-related indicators, and the computation of application-related indicators. The maintenance local decision of the gear box's replacement can be summarized in the (descriptor, value) style as in Table 2.3.

Table 2.3: Example of a maintenance local decision

Descriptor	Value
Type of action	Maintenance
Nature of action	Replacement
Concerned system/component	Gear Box
Spare Part Ref.	*ref. of the spare part
Required technician	Mechanic
Duration	2 hours
Start time	$t$
End time	$t+2$
Maintenance Technician	Operator 1
Initial Health State	(0.2, 0.9)
Estimated Health State	(0.2, 0)
Cost	200

Always in the same context of the CNC machine application, we present an example of a production local decision. In this example, we define some examples of descriptors for the

local decision derived from the previously presented production task. Therefore, one can observe that the local decision has inherited the descriptors of the elementary action. We also add descriptors that are related to the schedule. This includes the start and end time as time constraints and the batch of raw material to be used and the responsible production operator as required resources. For the health-related descriptors, we specify the initial health state of the machine before starting the operation, the estimated health state at the end of the operation, and the estimated degradation evolution with its confidence interval. The cost of production is specified among the estimated benefits obtained by selling the produced good minus the costs. The probability of achieving the operation without failure is also estimated as the success probability. These attributes and their illustrative values are summarized in Table 2.4.

Table 2.4: Example of a production local decision

Descriptor	Value
Type of action	Production
Order	Order no. 15
Urgent	No
Deadline	Due in 10 days
Quantity	150 pieces
Raw Material	Steel
Depth of cut	1.5
Feed rate	0.5
Rotation Speed	1500 rpm
Duration	4 hours
Start time	$t$
End time	$t+4$
Raw Material Batch	Batch Steel 2
Production Operator	Operator 1
Initial Health State	(0.2, 0.9)
Estimated Health State	(0.35, 0.96)
Estimated Degradation Evolution	$(0.15 \pm 0.02, 0.06 \pm 0.015)$
Cost	1700 u.m.
Estimated Benefits	1000 u.m.
Success Probability	0.95

To recap, the local decision is the result of a virtual scheduling of an elementary action. In this framework, we search for all possible elementary action and estimate the outcome of scheduling each one of them at instant  $t$ . This allows us to obtain the set  $\mathcal{F}(t)$  of all possible local decisions. Each of these local decisions presents an elementary action. Therefore, the descriptors of the elementary action are also used by the local decision. Other descriptors are added to clearly define the local decision. They can be classified into three categories. For each of these categories we present some examples of possible descriptors. As noted for the elementary actions, the descriptors of the local decision can differ according to the system studied and the intended application. Therefore, the choice of the descriptors is an important task of the implementation of the framework. Some non-exhaustive descriptors are given and grouped by category:

- **Schedule related descriptors:**

- *Start time or date*: To define the schedule of the actions, we need to define the start time or date of the decision.
- *Allocated Resources*: Some actions can require a special set of resources to be executed. Therefore, when these actions are transformed into decisions the required resources or the available resources are assigned to the local decision. The resources could be (i) material, like special tools for maintenance activity, or spare parts, or (ii) human, like a technician, maintenance teams, and production operators.
- **Health descriptors**: Here, we are going to define two attributes to describe the evolution of the system health state with the integration of the present decision. These attributes are:
  - *The initial health state indicators*: This can be defined by a set of health indicators of the studied system before the execution of the local decision.
  - *The estimated final health state indicators*: This is the set of the estimated final values of the health indicators after the execution of the local decision.
  - *Reliability level*: This describes the reliability of the system when executing the specified local decision.
  - *RUL values, and so on*.
- **Application related descriptors**: In this section of the local decision definition, we will define the attributes of the decision in the context of the proposed application. These attributes are used in the computation of the performance indicators of the application. For example, in this section we can find:
  - *Cost*: The estimated cost of the execution of the action.
  - *Benefits*: The likely benefits from the execution of the action.
  - *Mission success rate*: The predicted success rate of the mission if the action defined is integrated.
  - *and so on*.

At each instant  $t$ , we are going to create the set of possibilities for the decision-making process. This is done by creating the set of feasible local decisions  $\mathcal{F}(t)$ . Now, the decision-making process has to choose a local decision and schedule it. Since most decision-making processes are executed over a long duration (i.e. decision horizon), this act of selecting one local decision or a sequence of local decisions is done multiple times. And the selected decisions are scheduled sequentially. Therefore, we need to define the structure of the sequence of chosen local decisions. We call this structure global decision. This concept is defined in the next section.

### 2.2.3/ Global decisions

A global decision is a sequence of local decisions over a duration  $H$  called decision horizon. Therefore, a global decision  $g_u(t)$  can be defined with a countable set of descriptors  $D_{u,j}$  (attribute  $A_j$ , value  $V_{u,j}$ ). The descriptors of the global decision can be classified into three categories:

- **Composition Descriptors:** In this set of descriptors, the composition of the global decision is defined. Thus it contains two attributes:
  - *The Sequence of local decisions:* This attribute contains the sequence of local decisions. For example the schedule of jobs and maintenance actions.
  - *The Duration of the Horizon:* This attribute will define the duration of the decision horizon of the global decision.
- **Health Descriptors:** Similar to the local decisions, global decision defines the estimated evolution of the system's health state. For this some of the most commonly used attributes are presented:
  - *Initial Health State Indicators:* This contains the set of the initial health indicators before applying the global decision.
  - *Final Health State Indicators:* This is the set of the estimated final health indicators after applying the global decision.
  - *Reliability Level:* The evolution of the reliability of the system while applying a global decision, could be a relevant indicator of the system health.
  - ...
- **Application Related Descriptors:** In this category, one defines the attributes of the decision that are oriented to the application objective. These attributes are the performance indicators of the application plus other descriptors that can be used in the computation of the performance indicators. Some examples of the most common indicators are:
  - *The cost of the global decision*
  - *The predicted gain of the global decision*
  - *The Occupation (%) of the system*
  - ...

Therefore,  $g_u(t)$  is defined as:

$$g_u(t) = \{D_{u,j} = (A_j, V_{u,j})\} \text{ with } j \in \{1, \dots, \text{Card}(g_u(t))\} \quad (2.7)$$

To preserve the consistency of the definitions, the following notations related to the definition of the global decision are proposed. Global decisions are related to their time of construction, thus one can note:

- $\mathcal{G}(t)$ : The set of global decisions at time  $t$ .
- $g_u(t)$ : The  $u^{th}$  global decision at time  $t$  of  $\mathcal{G}(t)$ .
- $\text{Card}(\mathcal{G}(t))$ : The number of global decisions in  $\mathcal{G}(t)$ .

Therefore,  $\mathcal{G}(t)$  is defined as:

$$\mathcal{G}(t) = \{g_u(t)\} \text{ with } u \in \{1, 2, \dots, \text{Card}(\mathcal{G}(t))\} \quad (2.8)$$

In some cases, the number of possible actions can be tremendous causing a considerable number of feasible combinations and consequently a big set of global decisions. This

can be considered as an optimization problem. Therefore, a suitable optimization method from the literature should be used to construct a smaller set of global decisions by finding the most interesting sequences of elementary actions. The choice of such an optimization method is based on the application specifications in terms of constraints, real-time performances, the quality of the desired solution and so on. Such a point should be discussed during the implementation phase.

While the other descriptors may vary from one application to another, the composition descriptors are common for all types of applications since they are related to the proposed framework. Therefore descriptors  $D_{u,1}$  and  $D_{u,2}$  are defined as:

$$D_{u,1} = (\text{Sequence of local decisions}, V_{u,1}) \quad (2.9)$$

where:

$$V_{u,1} = (e_i)_1^{m_{g_u(t)}} \quad (2.10)$$

and:  $m_{g_u(t)}$ : The number of local decisions that build global decision  $g_u(t)$

$$D_{u,2} = (\text{Decision Horizon}, V_{u,2}) \quad (2.11)$$

where:

$$V_{u,2} = H(t) \quad (2.12)$$

$H(t)$  is the duration of the decision horizon at time  $t$ .

At this point, we have defined the concepts of the structure of possibilities and decisions. We also gave some hints about how these structures are defined and obtained. However, which layers of the PHM framework are responsible for these activities. More precisely in which module the elementary actions are transformed into local decisions, in which module the set of feasible local decisions  $\mathcal{F}(t)$  is built, and which module is responsible for choosing local decisions, scheduling them, and building the global decision.

## 2.2.4/ Estimators and Decisions Builders

The proposed method to highlight the prognostic decision interactions consists of building elementary actions into sequences of decisions while evaluating the outcomes of these local decisions in terms of the system's health and the objective of the studied application. In this context, two new modules are proposed to be added to the phases of analysis and decision support as shown in Figure 2.1.

A decision builder is joined to the decision support phase. The decision builder module can interact with the prognostic phase using an estimator module that is designed to forecast the degradation level of the system. The two modules are shown in Figure 2.1 and described as:

- **Decision builder:** This module is responsible for building the global decisions thanks to the set of available elementary actions  $\mathcal{A}$ . This module is part of the decision support phase of the PHM process. It has four main functions; (i) finding the possible elementary actions, and create the set of feasible local decisions  $\mathcal{F}(t)$  (ii) creating the set of possible sequences of local decisions  $S(t)$  at time  $t$  (iii) evaluating the elements of  $S(t)$  (iv) selecting and scheduling the suitable sequence of local decisions  $s^*$  from  $S(t)$ . The decision builder is the first element of decision-making in



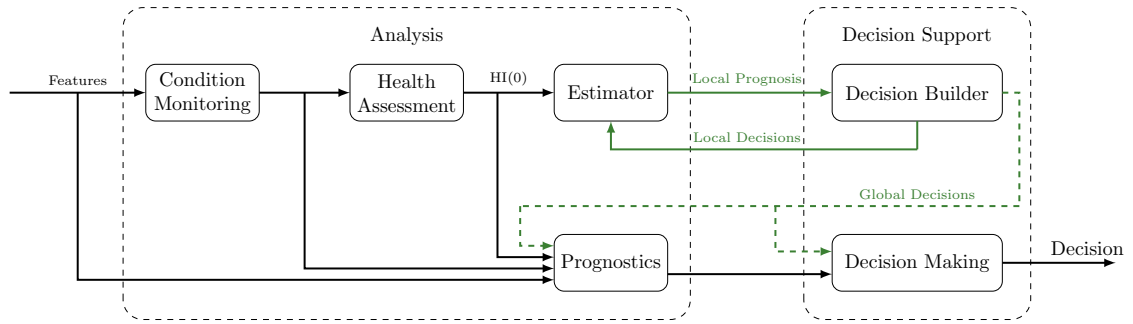


Figure 2.1: Block diagram of the decision building process

which it will assemble a set of global decision  $\mathcal{G}(t)$  at time  $t$ . Since post-prognostic decision-making is considered as an optimization problem, decision builders can be classified as an optimization method. Various optimization methods can be modified to become a decision builder, such as the search algorithms  $D^*$ ,  $A^*$ , ..., heuristics like the ant colony optimization, particle swarm optimization, and so on. The selection of the suitable decision building method is application related and should be addressed during the implementation phase.

- **Estimator:** This is a function that aims to forecast the degradation of the system under the new local decision. It evaluates the outcome in terms of health state (i.e. degradation, reliability, ...) of the studied system for each possible elementary action at time  $t$ . It is also equipped with a virtual system, that contains the future state of the system once a local decision is selected by the decision builder. This module is part of the analysis phase of the PHM process. Figure 2.2 shows an example of the output of the estimator. Estimator could be defined as a short term prognostic algorithm that can integrate future loads of the system. The choice of the used algorithm in this process can be subject to some constraints like the execution time and/or precision. The estimator has to evaluate the outcome of a large number of elementary actions and for some time-sensitive application, it could be more beneficial to use a swift prediction algorithm to reduce the estimation time. For applications that are not time-sensitive but that require a high level of precision, the estimator could be a replicate of the prognostic algorithm. Therefore, the estimator could be any model-based or data-driven prognostic algorithm. One can use various types of methods as estimators, for example, auto-regressive methods, linear regression-based methods, particle filters, neural networks, etc... The novelty here is to include the future local decisions in the estimation of the system future state. However, we limit the choice of the estimator by the conditions that it should be a short-term prognostic approach and it should take future decisions as an input.

The feasibility of local decisions and their sequences is a two steps process:

- In the first step, the decision builder considers all possible local decisions at an instant  $t$ . These possible local decisions are, then, tested against their time constraint. For example, let us consider the local decision of scheduling a production job at  $t = 100$  u.t. The job in question has a processing time of 50 u.t. meaning that its completion date will be equal to 150 u.t. But the job has a due date equal to 140 u.t. Thus it is

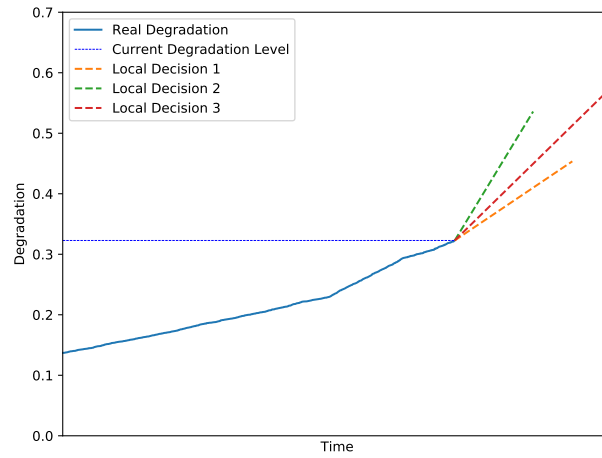


Figure 2.2: Example of the estimator output

judged as unfeasible. The time aspect of local decisions feasibility is presented in figure 2.3. Where,  $e_1$  is time-unfeasible and  $e_2$  is time-feasible.

- The second step is dependent on the results of the local prognosis provided by the estimator. For example, if we consider another production job that is feasible from a time point of view. The job could be considered as unfeasible if the outcome of its local prognosis exceeds the failure threshold of the system. The health aspect of local decisions feasibility is presented in figure 2.4. Where,  $e_1$  is health-unfeasible and  $e_2$  is health-feasible.

A sequence of local decisions is judged to be feasible if all the local decisions it contains are feasible in both time and health aspects.

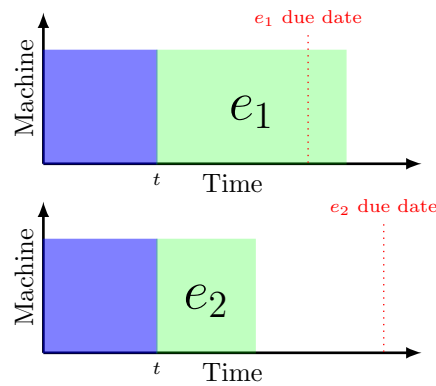


Figure 2.3: Time Feasibility of Local Decisions

## 2.3/ PHM framework adaptation

In the previous section, we presented the main concept needed to understand the modification that we made on the classic PHM framework. We have presented the added

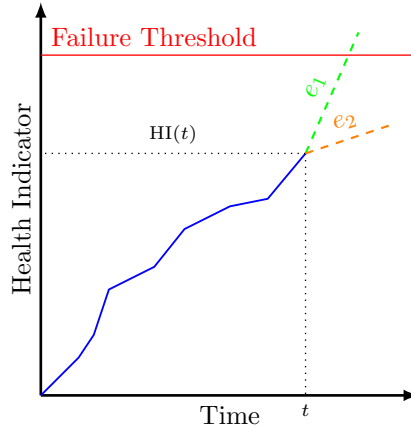


Figure 2.4: Health Feasibility of Local Decisions

modules and specified their responsibilities in defining, verifying, estimating, assessing, choosing, scheduling, and building the actions and decision structures. In this section, we explain the process of these activities and specify each of the proposed closed loops and their implications.

### 2.3.1/ Decision building loop

In the classic PHM context, decision-making and prognostics have always been studied separately and performed in a sequential manner. In this modification of the framework, we propose an iterative method for the decision-making process that allows a better integration of the prognostic information. For a given application, the decision builder searches for all possible elementary actions and transform them into local decisions. Local prognostics are done on these local decisions. While considering the new prognostic information provided by the estimator, the decision builder builds the global decision iteratively.

Figure 2.5 presents the sequential diagram of the decision building procedure. At the beginning of a PHM Stage  $k$ , the current state of the system is used to update the state of the estimator. A step of construction is initiated and the information is transmitted to the decision builder. The latter considers the set of elementary actions  $\mathcal{A}$  and create the set of candidate local decision sequences  $\mathcal{S}(t)$ . The set is feed to the estimator to obtain the short term prediction of the systems' state under each of these sequences. Once all the local prognostics are performed the set of possible sequences is updated with the final estimated state of the system and returned to the decision builder. The latter selects one of the sequences  $s^*$  based on the performance evaluation. The selected sequence  $s^*$  is given to the estimator to re-update its state. The estimator, then, checks the remaining time to the end of the decision horizon. If the duration of the decision horizon is reached, the building process is completed. Else, another step of construction is launched, until the end of the decision horizon is reached.

Once the decision-making process has received the set of global decisions  $\mathcal{G}(t)$ . The process can evaluate the evolution of the health indicators of each global decision by running it through the prognostic process then select the suitable global decision  $g^s(t)$  to be applied on the system or it can go directly to the selection process.

Although the construction of one global decision is quite a sequential process, the building of the set of global decisions  $\mathcal{G}(t)$  can be done concurrently. The possibility of constructing several global decisions at the same time depends on the capacity of the chosen algorithms for the decision builder and the estimator. For example, if one chooses the ant colony optimization as decision builder and linear regression as an estimator, the construction of  $Card(\mathcal{G}(t))$  global decision can be done all at once.

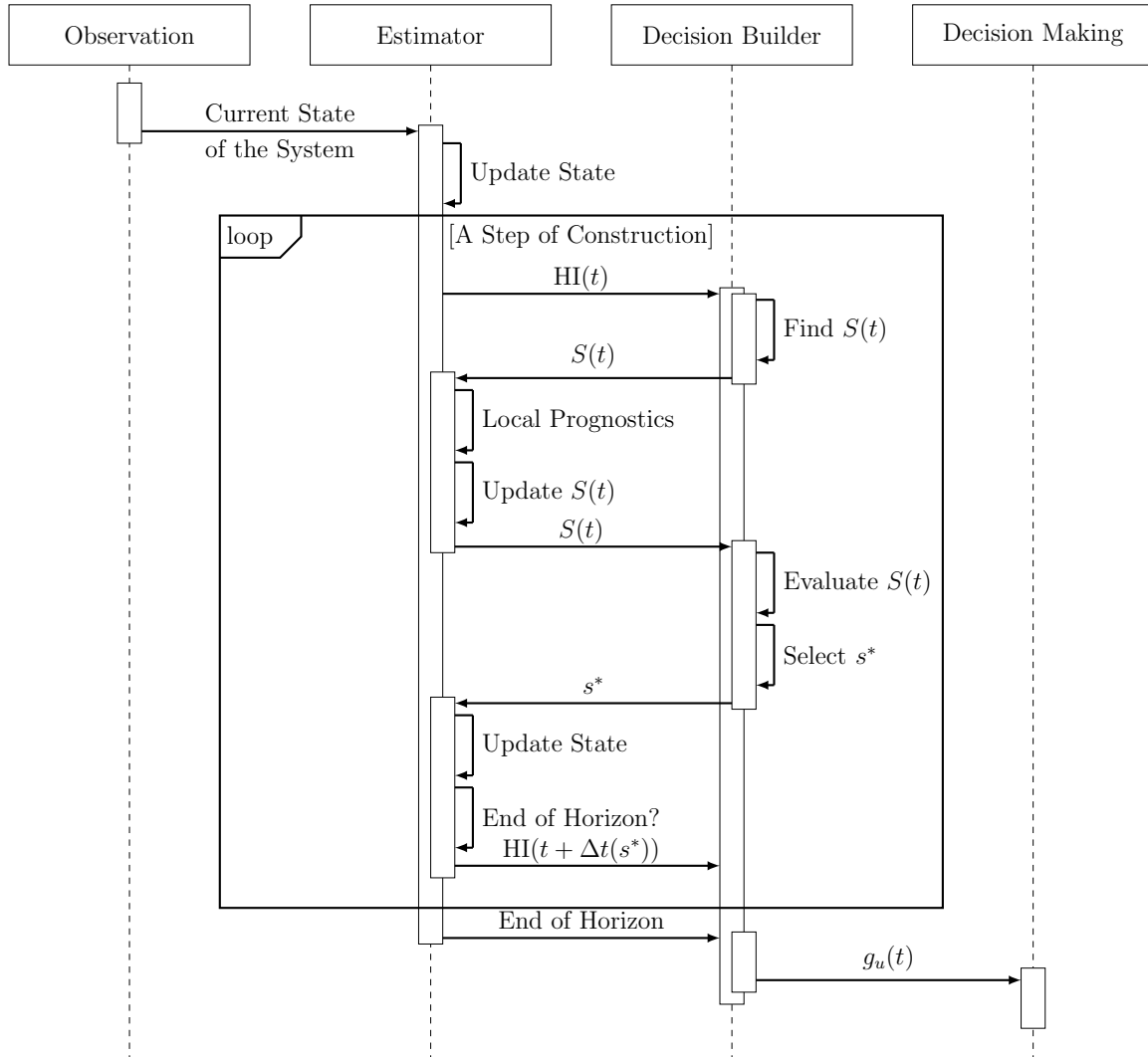


Figure 2.5: The Decision Building Sequential Diagram

By implementing this decision building loop, we propose a method to reduce the effects of two of the analysis phase issues:

- The prognostics accuracy variation regarding the duration of predictions. The short term prognostics are known for being more accurate than long term prognostics. The proposed decision building loop is based on these short term (local) prognostics computations. Therefore, we estimate that the built global decision present a more accurate estimations of the degradation evolution.
- This decision building loop is also based on the interaction between decision-making

and prognostics. Instead of taking decision based of a long term prognostics, the effects of each local decision is estimated and this information is used in the selection process. Estimating the outcome of a decision is different from estimating the RUL or the degradation evolution under average conditions and unknown future loads. Thus, this loop allows to reduce the uncertainties caused by the unknown operational conditions (i.e. the operational profile and the future loads).

### 2.3.2/ Decision applying loop

As said earlier, decisions are built in the PHM context to prevent a system from failing or to mitigate the effects of failure if avoiding it is no longer an option. Thus, when a global decision is selected, it will be applied in totality or partially on the system. The application of decisions (partial or global) leads to a real evolution of the health state of the system that can differ from the estimated one. Moreover, to capture the effect of this feedback, a re-execution of the prognostic and decision process is needed. Therefore, the proposed framework is set to be executed more than once, at specified moments called stages.

At a specific stage  $k$  (when  $t = T_k$ ), the system undergoes a normal PHM process, from data acquisition to health indicators assessment. Once the current state of the system is identified, the decision building process is triggered and the construction of global decisions over the specified horizon begins. When the decisions are built, the decision-making module selects a suitable decision for the next horizon. The selected decision  $g_k^s = g^s(T_k)$  is applied to the system until  $t = T_{k+1}$  the time of the next stage  $k + 1$ . At this new stage  $k + 1$ , the decision horizon is shifted, and the system undergoes once more the same PHM process. This process can be done over and over again as much as needed. This dynamic is represented in Figure 2.6 by the red feedback wire. An example of the decision process at stages  $k$  and  $k + 1$  is shown in Figure 2.7.

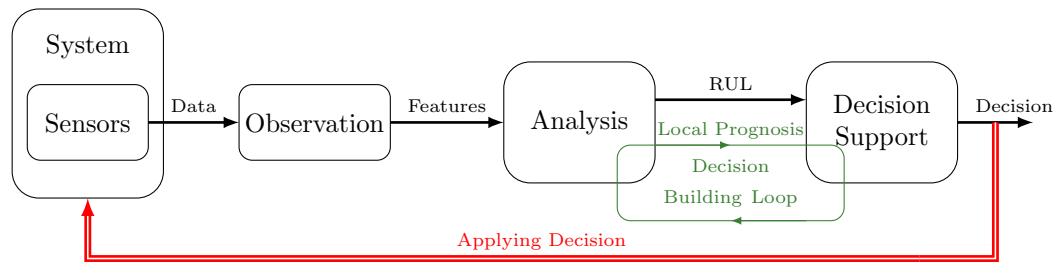


Figure 2.6: The Decision Applying Loop

The applied part of the selected global decision  $g_k^s$  between two consecutive stages  $k$  and  $k + 1$  is called **partial decision** of stage  $k$  (denoted  $p_k$ ). This partial decision is also represented by the set of descriptors  $(D_{u,j})$  like the global decision, although, the values  $(V_{u,j})$  of the attributes  $(A_j)$ , here, are the result of the application of the global decision on the system. Thus these values  $V_{u,j}$  represent the real values of the state of health evolution, the cost, and the duration.

The partial decision duration or the duration between two stages of decision is an important parameter of the proposed framework. If this duration is reduced to be equal to the duration of a local decision, then the proposed framework is similar to the activities of real-time

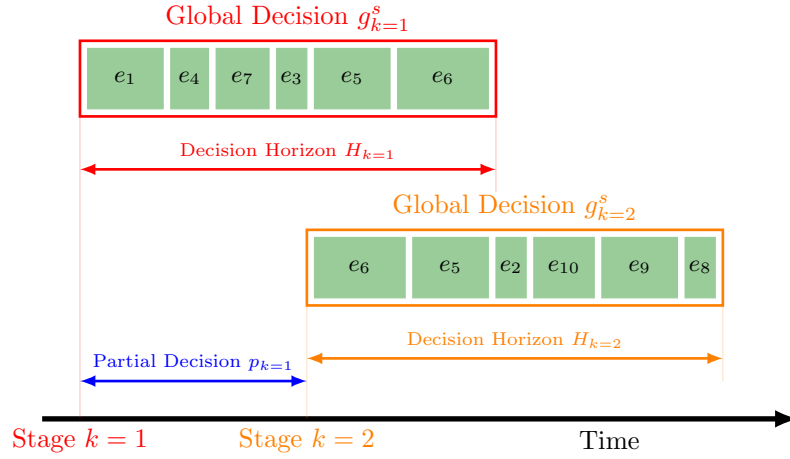


Figure 2.7: Illustration of the results of the decision applying loop

decision-making like in the case proposed by Zhang in [196], or the automatic control decisions detailed by Pereira *et al.* in [148], Nguyen *et al.* [139] or by Vieira *et al.* in [178] in which the dynamic of the RUL estimation is high. On the Opposite, if the duration of the partial decision is extended to be equal to the decision horizon duration, here the proposed framework is similar to the works that integrates a rolling horizon like the work of Van Horenbeek and Pintelon [173], plus if we suppose that the PHM process will be executed only once, the framework will be treating the post-prognostic decision-making like the one presented by Herr *et al.* in [79] or [81].

The decision horizon and the duration of the partial decision are two application-related parameters. We do not define a formula for these two parameters here since they highly depend on the application. When applying this framework, one has to study their suitable values to optimize the decision process. Such a study of the decision horizon duration is the subject of chapter 4 of this manuscript. In chapter 4, the effects of the decision horizon duration on the resolution of a joint maintenance and operational scheduling problem are studied.

No matter how accurate the prognostics algorithms are, there always be a difference between the estimated evolution of the system's health and its real evolution under the chosen decisions. One can only hope to reduce the difference between these evaluations and in this context make better-informed decisions. In this context, besides making decisions based on an iterative building loop, we propose to re-execute the prognostics and the decision-making process to capture the feedback from applying the decisions. This decision applying loop can help reduce the effects of uncertainties caused by the environment of the system. Such sources of uncertainties are often hard to model and to take into consideration directly. By re-executing the PHM process from time to time, decisions are build in a more accurate way and the effects of such uncertainties are reduced.

### 2.3.3/ Information/data loop

When the global decision is built, the evolution of the system health indicators is estimated under the selected local decisions. On the other hand, when the partial decision is applied

```

graph LR
    Features --> CM[Condition Monitoring]
    CM --> HA[Health Assessment]
    HA -- HI(0) --> Est[Estimator]
    HA --> Pro[Prognostics]
    Est --> FE[Fine Tune Estimator]
    FE --> Est
    CM -. Real Evolution .-> Sum((+ -))
    Sum -. Estimated Evolution .-> FE
    Est -- Local Prognosis --> DB[Decision Builder]
    DB -- Local Decisions --> Est
    Pro -- Global Decisions --> DB
    DB -- Global Decisions --> DM[Decision Making]
    Pro --> DM
    DM --> Decision
    subgraph Analysis
        CM
        HA
        Est
        Pro
    end
    subgraph Decision_Support [Decision Support]
        DB
        DM
    end

```

Figure 2.8: Information Loop

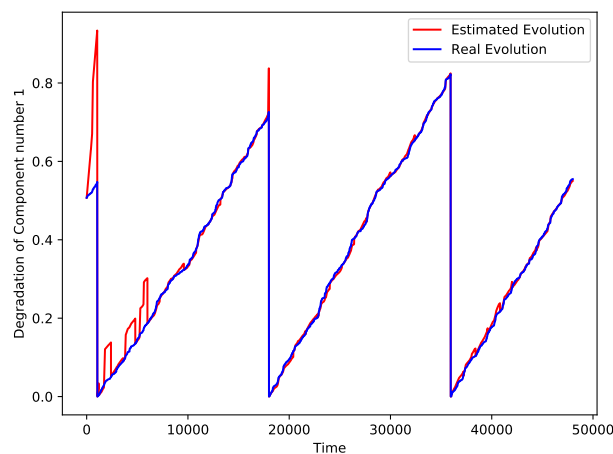


Figure 2.9: Degradation Evolution of Component Number 1

The information loop is considered as a foundation for methods that aims at fine-tuning the parameters of the estimator. While the role of the information loop is to gather the real evolution of the system state and compare it to the estimated one. The resulting error can be directly used to adjust the parameters of the estimator. One can find several methods for the estimator that benefits from the implementation of the information loop such as artificial intelligent methods, regression methods, case-based reasoning, and so on. For

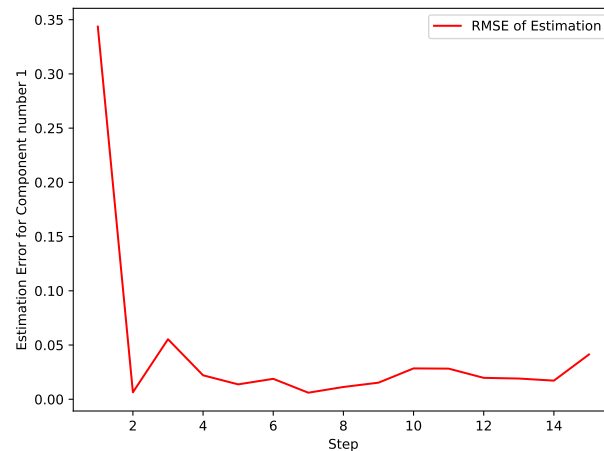


Figure 2.10: Evolution of RMSE of Component Number 1

instance, the newly available degradation data can be added to the set of training data for artificial intelligence-based estimators (like neural networks). Another example is the use of a case-based reasoning approach as an estimator. In this case, one can imagine the importance of the feedback information provided by the information loop, in the phase of revision to improve the estimator's parameters and/on in the phase of memorizing to save new relevant cases. In the case of regression methods, the new data can be used to re-adapt the parameters or even the type of regression model.

The integration of such a loop into a PHM process keeps the systems model up to date with the real system. Thus for a complex system with high variability of operating conditions or system behavior, such an information loop has an important in monitoring the system and in allowing the adaptation of the used methods to the newly available conditions.

When a system is subject to sources of uncertainties that could hardly be modeled in the degradation evolution, the information loop helps to capture the effect of these uncertainties and refine the system's degradation model.

### 2.3.4/ Overview of the Proposed Framework

To summarize this subsection, a new PHM process framework is proposed in which improvements are made on the classic PHM process by integrating three closed-loops processes (Figure 2.11). This new framework aims at dynamically estimating the RUL of the system while integrating the feedback of the decision on the system through the decision applying loop. The presented framework emphasizes the relationship between decision-making and prognostic modules by building decisions and estimating their outcomes via the decision building loop. Moreover, the information loop can be used to optimize the estimator's parameters and to analyze its performance.

To guarantee the efficiency of this framework, one should choose the suitable methods to use in the different modules. Some of the possible methods have been discussed earlier in the definition of the different modules. Yet, the choice of methods needs to be based on the treated problem and the requirement of the application in terms of execution time, accuracy and other constraints metrics.



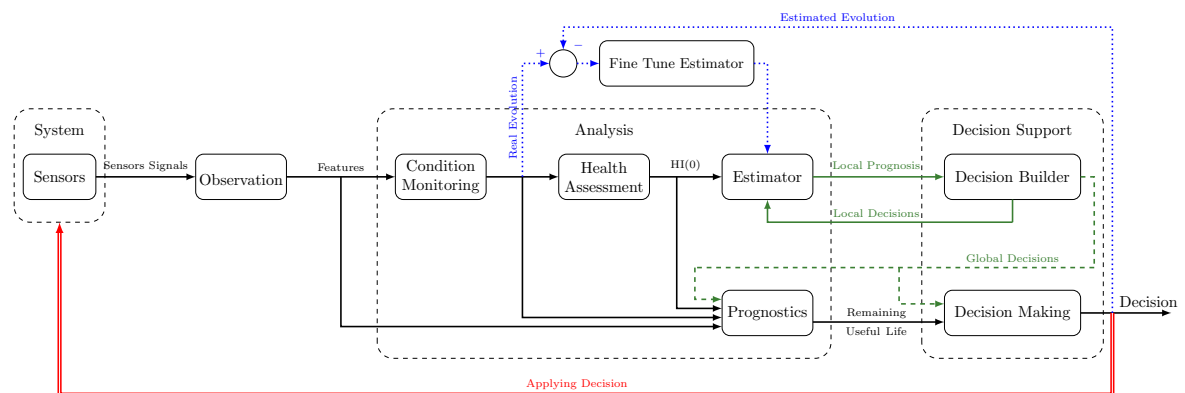


Figure 2.11: The Closed Loops PHM Process

## 2.4/ Metrics and parameters

When applying the proposed PHM framework, the following points should be taken into consideration:

- A good knowledge of the system's degradation modes, critical components, and effects of the failures is important to define the scope of the study. We recommend a minimum experience when dealing with the machine. A good starting point could be by performing the failure modes, effects, and criticality analysis on the targeted system.
- A good definition of the system and intended application are mandatory to define decision horizon, elementary actions, and local and global decisions.

Once the application is fully defined, the users should follow the specification when choosing which method to use in each of the processes. Most importantly, they should consider the reacting point of the PHM framework (i.e. the duration from acquiring new data to provide global decisions).

Same parameters of the PHM adapted framework require further investigations to be set. For example, the decision horizon value and the partial decision duration.

## 2.5/ Summary

This chapter discusses the required definition for applying the PHM framework. Moreover, some adaptation to the existing PHM framework are proposed to enhance the decision-making process. Three loops are proposed to improve the prognostic accuracy by eliminating uncertainty sources that are caused by unknown future loads and conditions, improve the performance of the decision-making process by dynamically incorporating prognostic information, and improve the re-activity of the overall PHM framework to new operating conditions. In the following chapters, two case studies of instantiating and implementing the proposed post-prognostic decision framework are detailed. The first application deals with a single multi-purpose machine. While the second application

studies a fleet of rolling stock units. Furthermore, the utility of applying the proposed framework, the duration of the decision horizon, and the frequency of PHM framework execution are discussed.

## Chapter 3

# Single Machine Production and Maintenance Optimization in PHM Context

### Contents

---

<b>3.1</b>	<b>Decision problem definition</b>	<b>65</b>
3.1.1	Related works and motivation	65
3.1.2	Scope of the study	67
3.1.2.1	Hypothesis	68
3.1.2.2	Constraints	69
3.1.3	System description	69
3.1.3.1	Machine structure	69
3.1.3.2	Degradation model	69
3.1.4	Problem description	72
3.1.4.1	Production Scheduling Problem	73
3.1.4.2	Maintenance Problem	74
3.1.4.3	Joint Problem	74
<b>3.2</b>	<b>Initialization of the proposed framework</b>	<b>75</b>
3.2.1	System definition	76
3.2.2	Application definition	77
3.2.3	Elementary Actions	79
3.2.4	Local Decisions	80
3.2.5	Global Decisions	81
3.2.6	Loops settings	82
<b>3.3</b>	<b>Implementation</b>	<b>83</b>
3.3.1	Sequencing problem	83
3.3.2	Ant Colony Optimization	84
3.3.3	ACO's prognostics-enhancement	85
<b>3.4</b>	<b>Numeric application</b>	<b>87</b>
3.4.1	ACO parameters' tuning	87
3.4.1.1	ACO Parameters	88
3.4.1.2	Decision Horizon	90
3.4.1.3	ACO Convergence	92
3.4.2	Adapting conventional CBM methods	92
3.4.2.1	Moore with condition-based maintenance	93
3.4.2.2	Genetic algorithm with condition-based maintenance	94
3.4.3	PHM+ vs. CBM	95
3.4.4	Investigating the use of PHM+	99

<b>3.5 Discussion</b>	<b>102</b>
3.5.1 Limits	102
3.5.2 Results Summary	103
<b>3.6 Summary</b>	<b>104</b>

---

In the manufacturing context, maintenance planning and production scheduling has been extensively studied separately in the literature. The growing requirements of the market is driving industries to optimize these processes simultaneously. With the emergence of industry 4.0, companies have a huge amount of data related to the operations of their assets. To take full advantage of these data, industries are looking to build models that can estimate the future conditions of the systems and use this new information in the decision-making process. In other words, they tend to apply the prognostics and health management methodology. However, in the industrial context, the production process presents great variability (e.g. type of product, type of material, etc...). Therefore, the classic prognostic model presents larger uncertainties caused by the unknown future loads. Hence, to fully benefit from the integration of the PHM methodology, industries are trying to master the prognostic-decision inter-dependency. This chapter presents a first application of the proposed PHM framework. To validate the approach, a multi-component single multi-purpose machine system is considered. The application is inspired from computer numerical control (CNC) machine in which machines are capable of producing several types of products with different production profiles (i.e. rotation speed, feed rate, etc). The chapter is divided into 4 major sections. First, we present the problem in a classical format i.e. in the optimization problem formulation. Second, we initiate the proposed framework to this problem formulation. We start by defining the system and the application. Then, we detail the elementary actions, local decision, and the structure of the global decisions. The section is concluded with the set ups of the needed loops. In section 3, we present the implementation of the framework. An ant colony optimization (ACO) algorithm is used as the decision method. The decision building loop is implemented within the ants' process of finding the shortest path between a food source and the colony. Section 4 presents the ACO parameters settings and its convergence performance study. To conclude on the performance of the proposed approach, we present the adaptation of some conventional methods that are used to solve the studied problem. Then, the solutions generated by the ACO are assessed and compared to those obtained by the conventional methods. Moreover, the use of the PHM framework is investigated. Finally, some discussion of the proposed methods, the obtained results, and future paths is done at the end of this chapter.

### 3.1/ Decision problem definition

In this section, we define the context of the application. We start by giving some related works that motivated this study. Then, we describe the system and the application.

#### 3.1.1/ Related works and motivation

Nowadays, the industry sector is experiencing an economic pressure due to the highly competitive environment driven by the market requirements. Which motivated the manufacturers to optimize their process in terms of time, cost and quality. In this purpose, companies are focusing on production and maintenance activities improvement. Therefore, one can find a lot of research works on these two major activities [1, 19, 44, 72, 96] of the industry sector.

The maintenance planning and production scheduling are usually in conflict since they

perform on the same resources. Although, undoubtedly, the fusion of these two activities will increase the agility of the manufacturing systems, they have always been optimized separately. One can still find a lot of works in literature that aims at optimizing one activity regardless of the other, yet in the last decade, one can easily notice the emergence of the optimization of maintenance and production scheduling simultaneously. Kaabi et al. [92], developed four heuristics to minimize the total tardiness of jobs on a single machine subject to systematic preventive maintenance. Benbouzid et al. [12], presented a sequential strategy and an integrated one to schedule maintenance and production activities in a flow shop. Few years later, Khelifati and Benbouzid-Sitayeb [94] proposed a multi-objective genetic algorithm to solve the same problem in a job-shop workshop. Najid et al. [157], optimized the joint scheduling of production and systematic maintenance for a multiple objects lot-sizing problem using a mixed integer linear programming. Fitouhi and Nourelfath [58], integrated noncyclical preventive maintenance scheduling and production planning for multi-state machines in order to minimize the total cost through a simulated annealing algorithm.

The implementation of PHM techniques is the scope of several research papers in this industrial context. These works, that are detailed in the first chapter, could be classified according the studied system's type and the considered workshop typology. We can find works that focused on single machine with single purpose [32, 145], single multi-purpose machine [100], multi-machines [13], job shop [59], and flow shop [99, 175].

Pan et al. presented in [145], a mathematical programming formulation to minimize the maximum tardiness of a single machine by simultaneously scheduling production and predictive maintenance based on it's RUL. Varnier and Zerhouni in [175], solved the joint scheduling problem for a flow-shop workshop where the machines have two production modes that influences their degradation by optimizing the aggregated sum of makespan and maintenance delays. Fitouri et al. [59], proposed a heuristic to solve the problem of job shop production and predictive maintenance scheduling for a job shop based on prognostic information. The proposed approach aims at minimizing the makespan and the total cost of maintenance. Ladj et al. [100] solved the integrated production and maintenance scheduling for a multi-functional single machine, in order to minimize the total maintenance cost, where each job is characterized by a degradation level. Later, Ladj et al., in [99], presented a hybrid of variable neighbor search and fuzzy logic to solve predictive maintenance and production scheduling for a permutation flow shop where each of the considered machines has its deterioration level, remaining useful life and a future degradation value when processing each kind of job. Liu et al. [120], used the health states and the RUL of a single machine to jointly schedule production activities and maintenance actions in the aim of minimizing the total cost. Bencheikh et al., presented in [13], a process to solve the joint problem for a multi-machine workshop using multi-agent systems. The agents are used to model the machines and the tasks. As the agents negotiate the schedule of production and maintenance based of the health states of the machines. Cheng et al. [32], presented a methodology to solve the problem of scheduling production and maintenance on a single machine while including the quality management of the produced parts.

In the previous studies, either a unique estimation of the RUL or the health indicators of the machines was performed and the obtained values is compared against a threshold to schedule maintenance regardless of the tasks being processed. Or the production tasks have a predetermined value of the amount of RUL they are consuming. However, the degradation of the machine under a new task depends of the initial degradation of

the machine and the operational profile used to perform the task. Besides initiating the proposed PHM framework, this chapter aims to solve the problem of jointly scheduling maintenance and production tasks on a single multi-purpose machine. Unlike the previous works, implementing the new PHM framework would allow a dynamic way of constructing decision and would also emphasize the use of prognostics information in the decision process. In other words, the proposed decision method is based on several estimations of health indicators instead of a single one.

With the emergence of new methodologies like condition-based maintenance and prognostics and health management, it has become difficult to choose what method should be applied in each scenario. Almost any work in condition-based maintenance or prognostics and health management context proposes to compare its approach to other classic maintenance approaches most commonly systematic periodic maintenance. The authors in the works of Camci [27] or Langeron *et al.* [103], prove that the use of prognostics and health management is more beneficial than other policies. Therefore, one can wonder if this statement is true no matter what type of component or machine.

Very few works have compared their results to the different maintenance policies to select the suitable one. For example, Van Horenbeek and Pintelon in [173] proposed a prognostic-based predictive maintenance policy, the results of their method are compared to several maintenance policies such as condition-based maintenance, and classic age-based policies. They also studied the influence of some parameters like the dependencies between the components and the prognostic horizon on the policies' performances. Hence, providing readers with a methodology to select a suitable maintenance policy. However, the authors have only done this study for a particular machine configuration. In other words, one can conclude on the choice of the optimal policy only if a similar machine configuration is present. Therefore, the question of which policy is more suitable for other configurations remains unanswered. Yet, the work of Van Horenbeek and Pintelon, like many other works, has not included the production scheduling and how it can influence the performance of the maintenance policy. Basically, the works on CBM or PHM maintenance scheduling implicitly assume that the operating conditions of the machines and their future loads are constant such as the works of Camci [27], Shi and Zeng [158], and Langeron *et al.* [103].

Hence, the second focus of this chapter is to study the appropriate maintenance method according to the system properties and the problem characteristics.

To summarize, the main contributions of the chapter are as follows:

- initiate and implement the proposed PHM framework,
- use this implementation to solve the joint maintenance and production problem on a single multi-purpose machine by emphasizing the decision-prognostics interactions, and
- study the utility of using prognostics information in the decision-making process according to the system properties and problem characteristics.

### 3.1.2/ Scope of the study

The intended application consists of managing the production and maintenance services for a single multi-purpose machine. The machine is capable of producing in several

operating profiles (i.e. speed, feed rate, etc..). The system is assumed to be composed of several predictive components. The idea is to schedule a set  $\mathcal{P}(i)$  of production tasks and plan the machine's maintenance intervention when needed. To mark out the scope of this study, we defined a set of hypothesis and constraints as described in the following sections.

### 3.1.2.1/ Hypothesis

The following hypothesis are used to define the scope of the study:

- H1 The machine can produce several products with different operating profiles.
- H2 All the components of the machine are considered as critical. Meaning that if any of the components fails the machine fails. In other words, the machine is composed of multiple serial components.
- H3 Dependencies between the components are limited to structural dependencies. The stochastic and economic dependencies are not considered in this application.
- H4 No redundancies in the component level is considered.
- H5 No preemption between tasks is allowed. Meaning that when the machine start a production task it has to finish it before starting another task.
- H6 Only maintenance tasks could be interrupted.
- H7 Only replacements are considered in the maintenance actions. The quality of a replacement is assumed to be perfect. Meaning, when a component is maintained (i.e. replaced) the degradation level is reset to zero.
- H8 The degradation or the health indicators of the predictive components is supposed to be observable and could be modeled through a physic-based, data-based, or hybrid-based models.
- H9 The degradation of the components is influenced by the operating conditions of the task to be achieved and by the type of the product.
- H10 The degradation of the components evolve only when the machine is performing a production task.
- H11 The degradation of any component could be modeled through a deterministic mathematical function.
- H12 No uncertainties are considered in this application.
- H13 Prognostics algorithms could be developed and implemented to estimate the evolution of the degradation or the health indicators of predictive components given the future tasks to be performed.



### 3.1.2.2/ Constraints

The following constraints are necessary to ensure the coherence of the application:

- C1 At instant  $t$ , the machine can have one of three status: (i) in production if a task is in process, (ii) at maintenance if it is scheduled to maintenance, or (iii) idle if it is capable of producing but no task is possible.
- C2 At any instant  $t$ , the machine can achieve only one production task.
- C3 At any instant  $t$ , the maintenance team could replace only one component.
- C4 The scheduling of a production task is not possible if any of the machine's components would have its degradation level exceeds the failure threshold.
- C5 Production tasks with completion time that exceeds the due date would generate lateness penalties.
- C6 Production tasks cannot be scheduled if their completion time exceeds the order's deadline.

### 3.1.3/ System description

This study focuses on a single multi-purpose machine that is composed of several components. The structure of the system, and the degradation model used are presented in the following paragraphs.

#### 3.1.3.1/ Machine structure

The machine used in this study is inspired from the computer numerical control (CNC) machines. These machines are known for their ability to operate on different types of products with different configurations (cutting speed, feed rate, and various tools). Therefore, we consider a single machine capable of producing  $\Pi$  types of products with  $J$  production (operating) profiles. We assume that the machine is multi-component i.e. it is composed of  $K$  components. These components are interdependent. We only considered structural dependencies between the components [141]. In a way, that if one component fails the whole machine fails. Therefore, the machine could be modeled as a sequence of serial components.

#### 3.1.3.2/ Degradation model

When dealing with multi-purpose machines, most works considered that the machine's degradation model is influenced by the type of operation and/or the operating profile [80, 100]. However, in the works of Herr *et al.* [80] and Ladj *et al.* [100], the degradation model of the machine or its components is considered linear and the sequence of the job does not influence the evolution of the degradation. In this study, we assume that the degradation model of the machine's components satisfies the following hypotheses:

- the deterioration is monotonically increasing,

- when the component is new its degradation level is supposed to be null,
- the component is subject to wear only when the system is operating,
- the type of the job influences the degradation evolution,
- the operation profile influences the degradation evolution,
- the degradation model is not linear, and
- the order of the job sequencing influences the degradation evolution.

Goebel investigated the milling tool wear in regular, entry, and exist cuts [68]. Goebel and Agogino published a data set of the milling tool wear under different conditions [66]. The authors varied the depth of cut, the feed speed, and the material of the pieces to mill. Then, they measured the wear of the tool with the help of a microscope. We used the obtained data set to validate the aforementioned hypotheses. First, we plot the obtained points for a given test case and we tried to fit linear and exponential regressions to the data points (Figure 3.1). For this purpose, we denote  $H(t_i)$  the measured tool's wear at instant  $t_i$  where  $t_i$  is the time of the  $i^{th}$  point and  $\hat{H}(t_i)$  the estimated tool's wear at the same point. We use  $\hat{H}(t_i) = a \times t_i + b$  for the linear model and  $\hat{H}(t_i) = a \times (e^{b \times t_i} - 1)$  for the exponential model. The obtained results of the models fitting are presented in Table 3.1. The models performance is compared by computing the root mean squared error (RMSE) defined in Equation 3.1, where  $N\_points$  is the number of points,  $H(t_i)$  is the real value of wear, and  $\hat{H}(t_i)$  is the value estimated. Table 3.1 shows that the performance of the exponential model is better than the linear when there is more points.

$$RMSE = \sqrt{\frac{1}{N\_points} \times \sum_{i=1}^{N\_points} (H(t_i) - \hat{H}(t_i))^2} \quad (3.1)$$

Table 3.1: Performances of regressions

Material	Cut's depth	Feed	N. of points	Linear RMSE	Exponential RMSE
Iron	1.5	0.5	22	0,075986	0,076959
		0.25	25	0,037007	0,034767
	0.75	0.25	32	0,047554	0,036717
		0.5	17	0,022999	0,027205
Steel	1.5	0.5	9	0,068475	0,081039
		0.25	12	0,098303	0,093748
	0.75	0.25	20	0,088613	0,031611
		0.5	7	0,024453	0,039548
Mean				0,057924	0,052699

The derived exponential model from each combination of material, cut's depth, and feed are then compared between each other to demonstrate the influence of each parameter. Figures 3.2 and 3.3 represent the exponential regressions models compared to each other. These figures validate the proposed hypothesis.

Therefore, in this chapter, we propose that the degradation model of the CNC machine's component is described by a exponential function (Equation 3.2). The proposed degradation model is influenced by the type of product in process (that includes material and

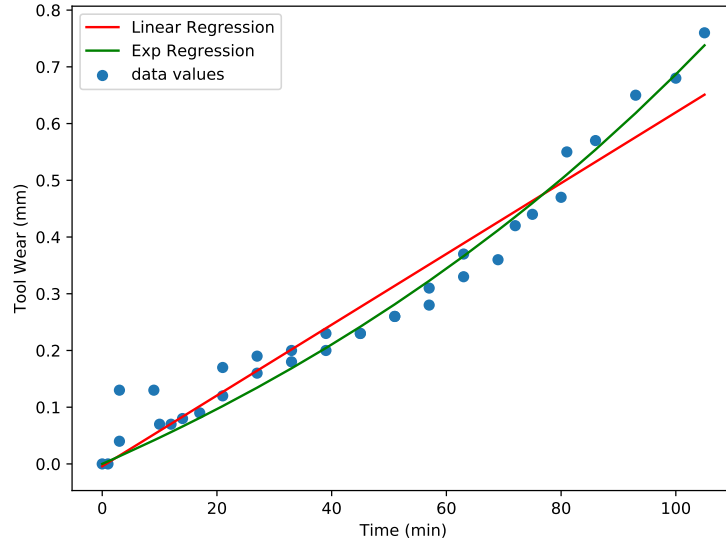


Figure 3.1: Example of regressions for an iron with cut depth of 0.75 and a feed of 0.25 based on the dataset of Goebel and Agogino [66]

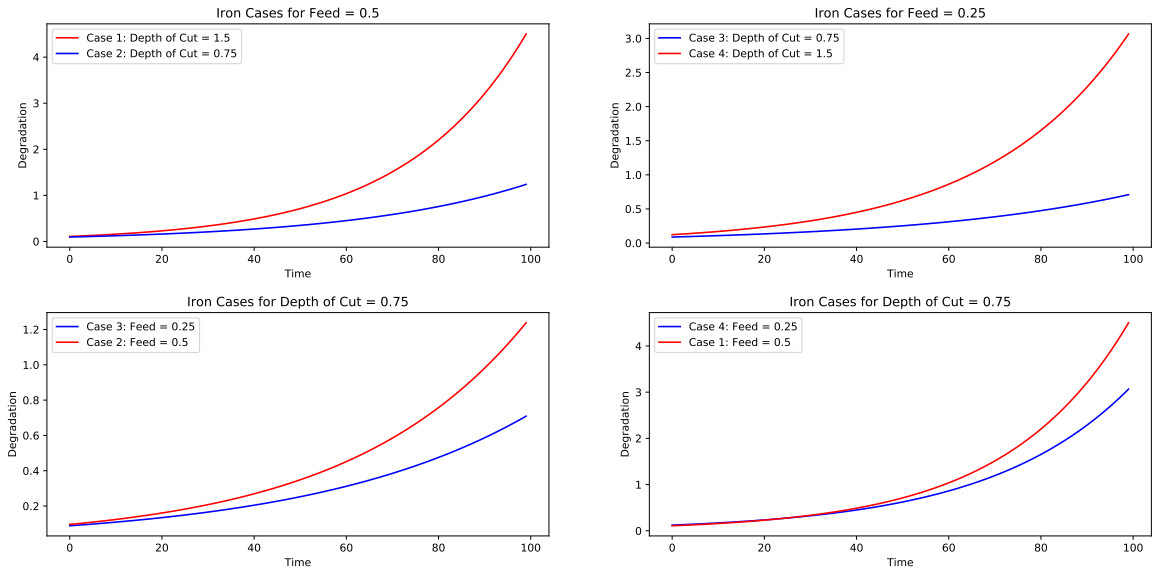


Figure 3.2: Exponential models for iron products

cut depth) and the operating profile.  $S_\pi$  and  $s_j$  represent respectively the influence of the product type  $\pi$  and the effect of the production profile  $j$  on the degradation of a considered component. For the rest of this work, we use the severity term to describe this influence. Therefore, we qualify  $S_\pi$  and  $s_j$  as the severity of producing product type  $\pi$  and producing using profile  $j$  respectively. Each component  $k$  has its degradation model  $H_k$  defined by parameters  $a_k$  and  $b_k$ . Since the component are supposed to be deteriorating only during the machine operations, we choose to model the degradation as a function of produced products  $x$  instead of time where  $x = \frac{t}{\Delta t_{\pi,j}}$  with  $\Delta t_{\pi,j}$  is the duration required to produce a unit of product  $n$  with profile  $j$ .

$$H_k(x, j, \pi) = a_k \times (e^{(b_k \times S_j \times s_\pi \times x)} - 1) \quad (3.2)$$

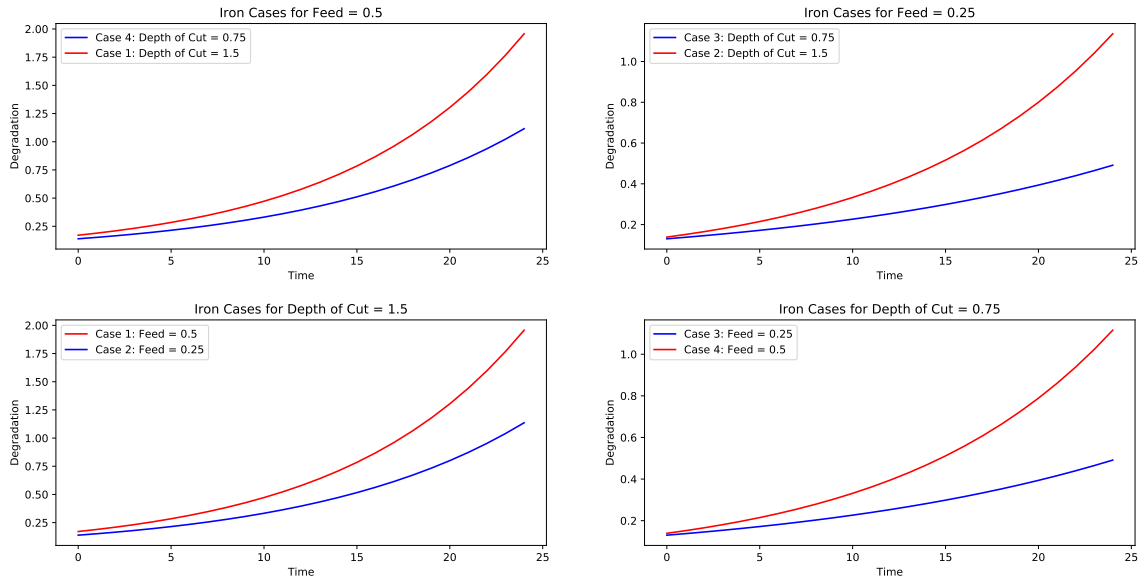


Figure 3.3: Exponential models for steel products

In this first application the degradation model of the machine is supposed to be deterministic. No uncertainties are considered in this case. The component is supposed to be in good health while its health indicator  $H_k$  is below a failure threshold noted  $\Delta_k$ . When  $H_k \geq \Delta_k$ , component  $k$  fails and hence causing the machine to fail.

### 3.1.4/ Problem description

In this section, we present the optimization problem studied in this application. The studied problem consists of scheduling maintenance and production for a single machine over the duration of the decision horizon noted  $DH$ . This decision horizon is often divided into  $I$  time periods (e.g., days, hours, or weeks). Hence, we write  $DH = I \times \Delta T$  where  $\Delta T$  is the duration of a time period (e.g. duration of one day). The notations used to define this problem are presented in table 3.2.

#### Notation

 $K$  $k$  $H_k$  $a_k, b_k$  $\Delta_k$  $DH$  $I$  $i$  $\Delta T$  $Sch_i$  $C_i$  $G_i$  $Maint_i$  $Prod_i$ 

#### Signification

Number of predictive components

Index of component  $k$ Health indicator (degradation) of component  $k$ Degradation parameter of component  $k$ Failure threshold of component  $k$ 

Duration of decision horizon

Number of periods in a decision horizon

Index of period

Duration of a period

Schedule of period  $i$ Cost of period  $i$ Estimated gain of period  $i$ Time spent in maintenance during period  $i$ Time spent in production during period  $i$

$Idl_i$	Time spent while machine is idle during period $i$
$\mathcal{P}(i)$	Set of production orders of period $i$
$p$	Index of order
$O_{i,p}$	Production order $p$ of period $i$
$\pi_p$	Product type of order $p$
$Q_p$	Quantity to produce in order $p$
$r_p$	Release date of order $p$
$d_p$	Due date of order $p$
$D_p$	Deadline of order $p$
$p_{p,j}$	Processing time of order $p$ with profile $j$
$c_{p,j}$	Completion time of order $p$ with profile $j$
$Cp(p, j)$	Cost of production of order $p$ with profile $j$
$Gp(p)$	Estimated gain of order $p$
$U_p$	Lateness indicator of order $p$
$cost_{\pi_p,j}$	Cost of producing one product of $\pi_p$ with profile $j$
$\mathcal{PM}$	Set of components to replace
$M_k$	Cost of replacing component $k$
$CM_k$	Failure indicator of component $k$
$P_{CM}$	Penalty on corrective maintenance
$C_m$	Cost of maintaining $\mathcal{PM}$

### 3.1.4.1/ Production Scheduling Problem

In this paragraph, we present the mathematics formulation for the production scheduling problem. To be considered as valid, a production schedule should satisfy the following constraints:

- no preemption between jobs is allowed,
- the machine can only produce one job at the time, and
- the sum of processing time of all scheduled jobs should not exceed the period duration.

During each period  $i$ ,  $\mathcal{P}(i)$  production orders  $O_{i,p}$  have to be processed. Each order  $p$  is characterized by a product type  $\pi_p$ , a quantity of products  $Q_p$ , a release date  $r_p$ , a due date  $d_p$ , and a deadline  $D_p$ . The production problem consists of finding the best schedule  $Sch_i^*$  of the  $O_{i,p}$  for the  $i^{th}$  period and the profile used for this job. Each order  $k$  has a processing time noted  $p_{p,j}$  under each profile  $j$ , and a completion date  $c_{p,j}$ . For any scheduled job, if its completion date  $c_{p,j}$  exceeds its due date then a penalty would be added to the cost. We suppose that the penalty value is fixed no matter how long the delay is. Any order that exceeds its deadline is considered as a lost opportunity. The scheduled jobs have a production cost  $Cp(p, j)$  (3.3) and generate a gain from selling the products  $Gp(p)$  (3.4). Each production order  $p$  scheduled with a profile  $j$  causes the machines components to deteriorate with a rate noted  $\delta_{p,j,l}$  for all component  $k$  of the machine. This degradation rate is also related to the initial condition of the machine before executing the scheduled order.

$$Cp(p, j) = Q_p \times Cost_{\pi_p,j} + U_p \times Q_p \times LP_{\pi_p} \quad (3.3)$$

$$Gp(p) = Q_p \times P_{\pi_p} \quad (3.4)$$

With  $Cost_{\pi_p, j}$  is the cost of producing one unit of product  $\pi_p$  with profile  $j$ ,  $U_p = 1$  if the job  $p$  is late otherwise  $U_p = 0$ ,  $LP_{\pi_p}$  is the tardiness penalty for product type  $\pi_p$ , and  $P_{\pi_p}$  is the sale price of product type  $\pi_p$ .

### 3.1.4.2/ Maintenance Problem

During the operating time of the machine, its components are subject to wear and tear. When a component degradation level reaches a failure threshold, the machine is no longer able to fulfill the required service. To avoid failure and to maximize the machine's availability and reliability, maintenance interventions are scheduled. We suppose that the replacements are the only possible maintenance intervention. Moreover, we assume that the maintenance quality is perfect, meaning that if one component is scheduled for maintenance it regains an "as good as new" condition after maintenance. We, also, assume that regrouping components does not reduce the intervention time nor the cost. Each component  $k$  of the machine has a replacement cost denoted  $M_k$ . When a component is scheduled for maintenance, a maintenance cost is generated for this component. The maintenance cost is defined in (3.5) as the sum of the replacement cost and a penalty if it is maintain early. In general, industrial companies want to maximize the use of their assets. The early maintenance of a component means that some duration of the component's life is wasted. Therefore, the proposed penalty for early maintenance depends on the difference between the current degradation level and the failure threshold. Meaning that the lowest this difference is the lowest the cost of the penalty. This is the main objective behind using PHM methodology: to use the assets to their full useful life while avoiding their failure. However if the component degradation level exceeds a threshold ( $H_k \geq \Delta_k$ ), the component  $k$  fails  $CM_k = 1$  involving a corrective maintenance activity with a cost  $P_{CM}$ .

$$C_m(l) = M_k + P_{rul} \times \max(\Delta_k - H_k, 0) + P_{CM} \times CM_k \quad (3.5)$$

As the machine has several components, the cost of a maintenance replacement action of replacing a set of components noted  $\mathcal{PM}$  is defined in (3.6). The maintenance activity does not have a direct effect on the estimated gain. Therefore, the gain of a maintenance decision  $G_m$  is assumed to be equal to zero.

$$C_m = \sum_{k \in \mathcal{PM}} M_k + P_{rul} \times \max(\Delta_k - H_k, 0) + P_{CM} \times CM_k \quad (3.6)$$

### 3.1.4.3/ Joint Problem

The objective behind this application is to find a suitable compromise between the production and the maintenance activities. This joint problem aims to establish a settlement between two services maintenance and production in a way that maximizes the benefits over the decision horizon duration  $DH$ . At each period  $i$  of the decision horizon, the joint schedule  $Sch_i$  of maintenance and production activities is built out of the local decisions of production and maintenance. The objective function can be described as:

$$\max \text{Benefits}(DH) = \max \sum_{i=0}^I B_i = \max \sum_{i=1}^I G_i - C_i \quad (3.7)$$

With  $B_i$  the benefits of the  $i^{th}$  period and  $C_i$  and  $G_i$  are respectively the cost and the gain of the  $i^{th}$  period as described in equation (3.8) and (3.9).

$$C_i = \sum_{(p,j) \in Sch_i} C_p(p,j) + \sum_{k \in PM_y} C_m(k) + L_{op} \times Idl_i \quad (3.8)$$

$$G_i = \sum_{(p,j) \in Sch_i} G_p(p) \quad (3.9)$$

With:  $L_{op}$  is a penalty on the time interval  $Idl_i$  during which the machine is capable of producing but it is idle for the lack of a production order.  $Idl_i$  is defined as in (3.10), with  $Prod_i$  and  $Maint_i$  are respectively the time where the machine is used to produce and the time spent in maintenance in the  $i^{th}$  period.

$$Idl_i = \Delta T - Prod_i - Maint_i \quad (3.10)$$

In general, the company wants to solve the joint problem and to maximize its benefits in the long term. To simulate the long term, we propose to solve this problem over a simulation horizon. This horizon, denoted  $SH$ , is divided into  $N$  steps of decision-making (i.e. joint problem resolution) over the decision horizon  $DH$ . The number of steps, noted  $N$ , is defined in a way that verifies  $SH = N * I * \Delta T$ . Therefore, we defined the cumulative benefits as the sum of the benefits of the  $N$  decision-making steps (see Eq. 3.11. This is illustrated in figure 3.4.

$$Cumulative\ Benefits = \sum_N Benefits(DH) \quad (3.11)$$

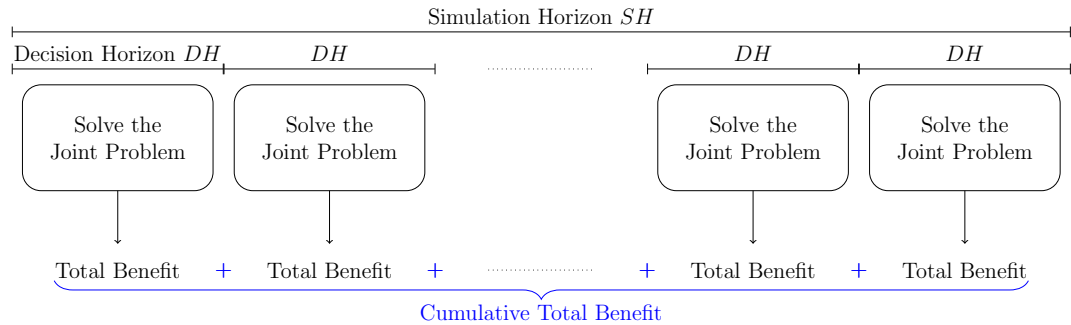


Figure 3.4: Illustration of Cumulative Benefit

## 3.2/ Initialization of the proposed framework

In this section, we put the presented framework in the second chapter into application. We define the different components of the framework and identify its mechanisms. To adapt the proposed framework for this case study, we start by identifying the values of the descriptors of the system and the application. Then, we define the elementary actions, local decisions, and the structure of the global decision. Finally, we set up the needed loops for this application.

### 3.2.1/ System definition

In this section, we present the system as defined by the proposed framework. Most of the attributes values are derived from the optimization problem statement. Some of the attributes are given empirically just to satisfy the needs for the example. We define the system's descriptors as follow:

- **System's type:** manufacturing
- **System's typology:** a single machine
- **Granularity:** In this example, we consider a single granularity level of the system. In other words, the system consists of several components that has the same hierarchy level. Therefore, the system is considered as multi-components.
- **Dependencies:** The  $K$  components of this machine are assumed to be only structurally dependent. The structure dependency is considered to be serial. Thus, we can see the machine a sequence of  $K$  components. The dependency is defined in a way that if one component fails the whole system fails.
- **Criticality:** This attribute is filled empirically. In the example, we are treating a single machine composed of  $K$  serial components. We assumed that each component is essential to the operations of the machine i.e. if one component fails the machine fails. Since no redundancies are considered for the components, we can assume that the components has a high critical level. With no redundant components or machines, if the one component fails, the whole system fails. However, the failure of the CNC machine does not imply environmental threats or user safety issues. It is supposed to have only effects on the products' quality. Maybe, the loss of production orders and some extra time and money for corrective maintenance. Even though, we try to eliminate failures, they can still be authorized. Let us assume that the machine has a medium critical level. A level that describes if one components fails the operations of the workshop will be suspended causing some financial issues but no serious threats to the environment or the users' safety.
- **Observability:** The degradation level of the components is supposed to be observable. This means that we can directly measure the degradation level accurately using microscopes or the degradation level could be deduced from measurable signals like vibrations, motor electrical measures, temperature values, and video microscopy.
- **Degradation parameters:** From the definition of the degradation model in the problem statement we can set the attributes for the degradation parameters. Such attributes should be specific for each component of the machine. In this example, since we did not specify which components are considered and to not lose the generic aspect of the application, we define these attributes in a general manner.
  - *Degradation form:* Some considered components are supposed to be subject of wear and fatigue e.g. rolling bearing. Some other components are supposed to be subject to wear and crack growth like for example gears of the gear box. The cutting tool is supposed to be subject to wear.
  - *Degradation causes:* Degradation evolution is supposed to be related to the operations of the system. When the system is idle the degradation level is



supposed to be constant. Therefore, the degradation evolution is operationally caused. The characteristics of this evolution is assumed to be influenced by operational conditions like the cutting speed, rotation speed, feed rate, type of material, type of product (i.e. depth and profile of the cut).

- *Degradation model*: For this system, we propose a deterministic degradation model. The model is described by an exponential function. The form of this function is obtained by fitting data. Therefore, the used degradation model is assumed to be data-based model. However, since we do not actually have real data for the system and the database given by Goebel and Agogino does not include a large number of data points to allow an accurate estimation of the parameters. Therefore, the parameters of the components are given empirically.
- *Degradation consequences*: The CNC machine is not considered critical. Therefore, we assume that the degradation of the components causes a loss in the quality of the products. Therefore, the failure of some components is not defined when they break but by reaching a certain level of quality loss in the product. To simplify this, we translated this level of quality loss into a threshold of component degradation. Once the degradation level of a component exceeds the threshold we suppose that the produced products do not satisfy the quality control requirements. Thus, this is considered as a failure of the system caused by the failing component.
- **Overall degradation dynamics**: since the components are structurally dependent, then the machine overall degradation follows the component with the highest degradation level.
- **Action variables**: The machine operations can be controlled through two variables: (i) the type of product (i.e. its' material, the depth and profile of cut) and (ii) the production profile used for the operation (this includes the cutting speed, the feed rate, and the rotation speed).
- **Complexity level**: The definition of the complexity level of the system is a difficult task and no consensus method is yet to be proposed. Therefore, we are giving an empirical value for this attribute. We consider that the system has a low complexity level. This is justified by the following reasons:
  - The system is composed of a single machine
  - The machine is composed of a single level of granularity composed of several components.
  - components' dependencies are only structural and present a single type of structural dependencies in which all the components are considered serial.
  - The degradation level is deterministic and can be directly observed or deduced accurately from physical measures.

### 3.2.2/ Application definition

Therefore, the application can be summarized as:

- **System**: As we discussed in the second chapter the application definition should include the detailed definition of the system. Therefore, for this attribute, the definition of the system done in the previous section is kept.

- **Objective:** this is a cost oriented application in which we are trying to maximize the benefits over a time duration (i.e. simulation horizon). The objective function is defined in Eq. 3.7 in subsection 3.1.4.3.
- **Performance indicators:** To follow and assess the efficiency of the methods we propose several indicators that are monitored in the decision-making process i.e. the optimization process. In general, the company wants to solve the joint problem and to maximize its benefits in the long term. To simulate the long term, we propose to solve this problem over a simulation horizon. The obtained solution are compared and assessed upon these indicators that are computed over the simulation horizon:
  - *Total Benefit:* This is the estimated profit of the workshop over the simulation horizon. It is obtained by computing the difference between the total cost over the simulation horizon and the total estimated gain over the simulation horizon.
  - *Total Cost:* During each decision horizon, we compute the cost of the global decision. Then, the sum over the simulation horizon of all the costs of global decisions produce the total cost.
  - *Total Gain:* Global decision are characterized by the potential gain of its scheduled decisions. This is obtained by estimating the price of selling the produced order. The sum over the simulation horizon of all the gains of the global decisions define the total gain.
  - *The proportion of the total time spent in production:* This present the ratio of time in which the machine is producing the orders.
  - *The proportion of the total time spent in maintenance:* This present the ratio of time in which the machine is not available for production due to maintenance activities.
  - *The proportion of the total time spent when the machine is idle:* This present the ratio of time in which the machine is operational but is not producing due to lack of production orders or the remaining time to the end of period does not allow the scheduling of a production action.
  - *Total number of maintenance activity per component:*
  - *Total number of corrective maintenance activities per component:*
  - *Total number of corrective maintenance activities:* This is the sum of all corrective maintenance intervention done during the simulation horizon. It corresponds to the sum of the corrective maintenance activities during each global decision.
  - *The average lost degradation level per component:* When performing predictive maintenance, there is a difference between the threshold of failure and the actual level of degradation of the component in question. This difference is used in Eq. 3.5 to compute the lost RUL penalty. At each maintenance activity, this difference is computed and saved. Then, the average difference for each component per maintenance activities is computed. This allows to determine how well the maintenance activities are scheduled. A big value of this indicator shows a relative premature maintenance scheduling. A null value indicate that the schedule fails to anticipate the failure. The aim of PHM is to minimize this difference while still being able to avoid failure and then corrective maintenance.
  - *The proportion of done orders:* This present the ratio of the orders that are produced from all available orders over the simulation horizon.

- *The proportion of missed orders*: This present the ratio of the orders that have not been scheduled before their deadline from all available orders over the simulation horizon.
- *The quantity produced*: This is the sum of all produced goods during the simulation horizon.
- **Decisions nature**: we are studying a joint problem between production and maintenance. Thus, the decisions in this application are mixed. The two nature of the actions can be defined as:
  - Action of maintenance type: selecting a component  $k$  to be maintained
  - Action of operational type: scheduling order  $p$  with production profile  $j$ . It is important to note that the operational action is a two level action. This is caused by including the two action variables. By changing one of these variables we obtain another action.
- **Simulation horizon**: In real life, there is no simulation horizon since the application is supposed to be running infinity. However, since this application is a proof of concept we defined a simulation horizon upon which we validate the approach and assess its performance over a long enough run.
- **Decision horizon**: we consider a rolling decision horizon.
- **Discretization Period**: The decision horizon is divided into smaller periods of time we call discretization period. In this application the discretization period has a one day duration. We assumed that production operation do not allow preemption. Therefore, it is not allowed to start a production operation today and resume it tomorrow. Thus, this constraint define the shortest duration of time as one day period.

### 3.2.3/ Elementary Actions

The studied application consists of a joint problem of maintenance and production scheduling. Therefore, two types of elementary actions exists: (i) operational elementary action and (ii) maintenance elementary action.

Let's start by defining the operational elementary action. In the definition of the application, the operational elementary action is defined as the production of a certain order  $k$  with production profile  $j$ . We propose to detail the attributes used for defining the operational elementary action as in table 3.3.

The maintenance elementary action consists of choosing which component to maintain. There could be  $K$  maintenance elementary actions. Each define the maintenance intervention of a component. There could also be elementary actions that regroup components. However, in this application, we assume that regrouping components does not reduce the intervention time nor the cost. Therefore, the maintenance of two components can be seen as two elementary actions that are transformed to two successive local decisions. The definition of a maintenance elementary action is given in table 3.4.

Table 3.3: Definition of a production elementary action

Descriptor	Value
Type of action	Production
Order	$p$
Due Date	$d_p$
Deadline	$D_p$
Quantity	$Q_p$
Product Type	$n_p$
Profile	$j$
Duration	$p_{p,j}$

Table 3.4: Definition of a maintenance elementary action

Descriptor	Value
Type of action	Maintenance
Nature of action	Replacement
Concerned component	$k$
Duration	$Duration_k$

### 3.2.4/ Local Decisions

At instant  $t$ , we propose to find all the possible local decisions. We consider the set of the elementary action and we estimate the outcome from scheduling these actions at this instant  $t$ . We have two types of elementary actions and consequently, we have two types of local decisions.

Let's start by defining the attributes for a typical maintenance local decision. In this application, maintenance decisions are considered for each component separately since no gain neither in the duration of the invention nor in its cost is obtained through regrouping components for maintenance. The defined local decision is presented in table 3.5.

Table 3.5: Structure of a maintenance local decision

Descriptor	Value
Type of action	Maintenance
Nature of action	Replacement
Concerned component	$k$
Duration	$Duration_k$
Start time	$t$
End time	$t + Duration_k$
Initial Health State	$(H_1, \dots, H_{k-1}, H_k, H_{k+1}, \dots, H_K)$
Estimated Health State	$(H_1, \dots, H_{k-1}, H_k = 0, H_{k+1}, \dots, H_K)$
Cost	$M_k + P_{rul} * \max(\Delta_k - H_k, 0)$
Gain	0

Considering the definition of production elementary action, the local decision's derived from this action is defined in table 3.6.

Table 3.6: Structure of a production local decision

Descriptor	Value
Type of action	Production
Order	$p$
Due Date	$d_p$
Deadline	$D_p$
Quantity	$Q_p$
Product Type	$n_p$
Profile	$j$
Duration	$p_{p,j}$
Start time	$t$
End time	$c_{p,j}(t) = t + p_{p,j}$
Initial Health State	$(H_1, H_2, \dots, H_k, \dots, H_{K-1}, H_K)$
Estimated Degradation Evolution	$(\delta_{p,j,1}, \delta_{p,j,2}, \dots, \delta_{p,j,k}, \dots, \delta_{p,j,K-1}, \delta_{p,j,K})$
Estimated Health State	$(H_1 + \delta_{p,j,1}, \dots, H_k + \delta_{p,j,k}, \dots, H_K + \delta_{p,j,K})$
Lateness ( $U_p(t)$ )	$U_p(t) = 1 \text{ if } (c_{p,j}(t) > d_p) \text{ else } U_p(t) = 0$
Cost ( $Cp(p, j, t)$ )	$Q_p * C_{n_p,j} + U_p(t) * Q_p * LP_{n_p}$
Gain ( $Gp(p, t)$ )	$Q_p * P_{n_p}$

### 3.2.5/ Global Decisions

A global decision, in this application, is defined as the schedule of production operations and maintenance intervention during a decision horizon. As described in the previous chapter, the global decision can also be defined as a set of (attributes, values). To carry on with the implementation of the proposed framework, we define in table 4.8 the structure of the global decision used through this application. For the descriptors, we do present the used formula to compute the values instead of an example of values.

Table 3.7: Structure of a Global Decision

Descriptor	Value
Schedule	$Sch_i$
Decision Horizon	$DH$
Initial Health State	$(H_1(0), H_2(0), \dots, H_{L-1}(0), H_L(0))$
Estimated Health State	$(H_1(0) + \sum_{e_x \in Sch_i} \delta(e_x, 1), \dots, H_K + \sum_{e_x \in Sch_i} \delta(e_x, K))$
Estimated Health evolution	$(\sum_{e_x \in Sch_i} \delta(e_x, 1), \dots, \sum_{e_x \in Sch_i} \delta(e_x, K))$
Global Cost	$C_i$ (Eq. 3.8)
Global Gain	$G_i$ (Eq. 3.9)
Global Benefit	$B_i = G_i - C_i$
Time in production	$Prod_i = \sum_{(p,j) \in Sch_i} p_{p,j}$
Time in maintenance	$Maint_i = \sum_{k \in \mathcal{PM}_i} Duration_k$
Lost time	$Idl_i$ (Eq. 3.10)
Number of Corrective Maintenance	$\sum_{DH} \sum_{k=1}^K CM_k$
Done Orders	$\sum_{p \in P(i)} (p \in Sch_i)$
Missed Orders	$\sum_{p \in P(i)} ((p \notin Sch_i) \text{ and } (D_p \leq DH_i))$
Quantity Produced	$QP_i = \sum_{(p,j) \in Sch_i} Q_p$

### 3.2.6/ Loops settings

In the previous chapter, we presented the proposed improvement of the existing PHM process. To enhance the use of prognostics information in the decision-making process and vice-versa, three loops have been introduced:

- An **estimator** module that predicts the outcome of local decisions on the system's health indicators. This prognostics information is used by the decision builder in the process of selecting a suitable local decision and scheduling it. Then, it searches for new possible local decisions and convey them to the estimator for local prognostics. Iteratively, decisions are built. Hence, the loop is called decision building loop.
- Global decision (i.e. the output of the decision building loop) are selected, approved, and applied on the system by the decision-making module. The application of these schedules is a natural process that requires modeling in the PHM framework. This is the decision application loop.
- Between the estimated evolution of the health indicators and their real evolution, a difference could be found due to uncertainties sources that are not accounted for. This difference is captured through comparing the two evolutions. The resulted errors of local prognostics are used to improve and fine tune the estimator's parameters. This loop is based on the system's health information. Hence the name of information loop.

This application constitutes a first test of validation of the proposed modifications. Therefore, we propose a simplified example in which not all the loops of the framework are implemented. The focus of this chapter is to study and validate the implementation of the decision building loop.

In this application, we use a deterministic degradation model (see Eq. 3.2) and we suppose that the model's parameters are known. The degradation of the components are modeled through an exponential function. Therefore, we can use the model as an estimator to provide the degradation level after achieving a certain task. This study is also based on the assumption that no uncertainty sources are considered. We assume that the system is placed in a highly controlled environment.

Since the model is deterministic and no uncertainties are considered, the estimated level of degradation is the actual level that is obtained through applying the decision. Hence, the estimator's outputs are valid and accurate. Thus, there is no need to implement the information loop since the estimated and the real values are identical. Consequently, the loops of the proposed PHM+ framework are reduced to two loops: (i) the decision building loop (in green) (ii) and the decision applying loop (in red). The implemented loops are presented in figure 3.7. The information loop and its components and the prognostics module are presented in light gray because they are not considered in this study.

The three-loop framework is introduced as a potential solution to manage and reduce some of the uncertainties sources especially the one related with future loads and conditions of the system. The horizon of the predictions (prognostic horizon) has a direct influence on the prognostics performances. The further in the future, we are trying to predict, the more likely the context of the estimations will change and uncertainty will increase. Therefore, for large prognostic horizon we can expect less accuracy of the results. This issue is addressed by the decision building process. The loop propose a small local prognostics

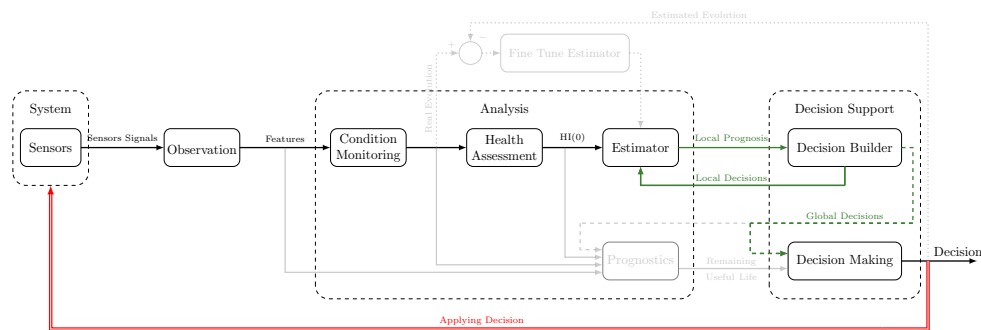


Figure 3.5: Implemented Loops

based on the future loads instead of general long term prognostics. Even though, this is expected to reduce the uncertainties and improve the accuracy of the decision-making process. Thus, the prognostics horizon is shorten to a local prognostics. However, for a global decision local prognostics are done sequentially based on the previous estimation. Therefore, the more local decision constructed in a global one, the more likely predictions become inaccurate. The process is influenced by the duration of the decision horizon. We introduced in the framework the partial decision variable which represents the applied part of the global decision on the system before the decision is questioned. This partial decision is mainly based on the difference between the estimated and the observed evolution of the system's health state.

Moreover, the use of a deterministic degradation model and the absence of uncertainties allow us to estimate the exact outputs of the decisions. Therefore, the decision horizon does not present any influence on the health indicators estimations nor on the estimators performances. Hence, the partial decision parameter of the framework does not influence the performance of the system. In other word, building decision over a long duration and applying it partially or building it on a short duration has the same outcome since we control the outcomes of the decisions. Hence, we will consider a partial decision that is equal to the global one i.e. decisions are applied in totality in this case. Therefore, the decision applying loop is executed at the end of each decision horizon.

### 3.3/ Implementation

The previous sections of this chapter presented the studied problem in an optimization way and the initialization of the framework. The required loops for the implementation are highlighted. The next step of the decision enhancement process is the choice and set up of the decision-making method.

#### 3.3.1/ Sequencing problem

The objective of this approach is to find a suitable compromise between producing and maintaining the system by integrating the health information in the decision process. It can be seen as finding the sequence of production tasks that fits at most the duration of the decision horizon (i.e. reducing the idleness of the machine), optimizes the machine

health conditions (i.e. reduces the waste in terms of component's remaining life and avoids failures), and maximizes the benefits of the workshop over the decision horizon and over the simulation horizon.

The sequencing problem can be seen as a special travel salesman problem (**TSP**). The travel salesman problem was originally described by Lawler [107] as the search for the optimal path that allows to visit several cities each one time while optimizing a travel cost. In our case, production orders and maintenance activities are considered the cities. Each city can be visited at most one time except for the maintenance activities. Therefore, in the same schedule during a decision horizon we can find several maintenance activities for the same component.

During period  $i$ , each task (production order)  $k$  has several paths. Each path correspond to a production profile. Therefore, each path has a duration (i.e. processing time of the destined task with that specific profile), cost of production, and benefits. The paths to a maintenance activities are the specification of which component to maintain.

The paths between tasks are not always possible. The possibility of taking a path is subject to time constraints (Equations 3.12 and 3.13) and health constraint (Equation 3.14). Therefore, at each movement the map of possible paths should be recalculated. The construction of the map of possible paths is the main objective of the estimator in the decision building loop.

$$\text{at instant } t; t + p_{p,j} \leq DH \quad (3.12)$$

$$\text{at instant } t; t + p_{p,j} \leq D_p \quad (3.13)$$

$$\text{at instant } t; H_k(t) + \delta_{p,j,k} \leq \Delta_k, \forall k \quad (3.14)$$

The classic TSP problem and several of its derivatives have been widely solved in the literature [150, 152]. The used methods include exact resolution [9], tabu searches [10], genetic algorithms [106], ant colony optimization [49], etc. We started looking for inspirational resolution methods that could solve our problem while implementing the proposed enhancement for the PHM process (i.e. the decision building process, the prognostics information, and the decision-prognostics interactions). From the potential methods, one algorithm stands out: the ant colony optimization technique.

### 3.3.2/ Ant Colony Optimization

The ant colony optimization (ACO) is a well used probabilistic method for computational problems that goes from scheduling/sequencing jobs [130], finding good paths in a graph [11] to designing systems [125]. The algorithm proposed by Dorigo and Di Caro in [48] is based on the biological communication system of ants. When looking for food, ants wonder randomly. Once they find a source of food, they return to the colony while depositing a secreted or excreted chemical factor called pheromones. While wondering randomly, some ants would come across a pheromone trail. These ants would still choose their next step randomly, but the deposited pheromone quantity influences this choice by increasing the probability of the ant following the pheromone trail.

Since the colony is composed of thousands of ants, the search for food is a parallel process. Each ant of the colony could find a source of food and while going back and



deposing the pheromones the paths of different ants could cross and merge for small of long periods. Moreover, when several ants find the same source of food they take the same path back to the colony and thus, depositing larger quantity of pheromones. However, with time passing by the quantity of pheromones start to reduce due to evaporation. So if a path is not taken very often eventually it will be forgotten. On the other hand, a path that is taken with high frequency will eventually attract most if not all ants. This explains the long lanes of ants, one can observe almost everywhere. This biological concept was captured and introduced in an artificial pheromone system.

The studied joint maintenance and production problem is similar to the ants food search. One can imagine the food source as the benefits of the workshop after traveling (i.e. executing) the path (i.e. the schedule) constructed by the ant. In this process, at each step the ant choose a part of the path (i.e. a local decision) to go until reaching a food source (which could be seen as reaching the end of the decision horizon). This aspect of ants wondering randomly discovering the space and their iterative paths' construction process inspired us to choose the ant colony optimization algorithm, enhance its prognostics, and implement it as a decision-making method for our problem. The steps of prognostics enhancement and method implementation are explained in the next paragraph.

### 3.3.3/ ACO's prognostics-enhancement

We implement the ACO artificial pheromone system for path finding to solve our problem by enhancing the prognostics and the decision-making mechanisms. Ants initially wonder randomly in the space. At each step, they stop, assess their surroundings, then choose a path randomly. Hence, the ant is already a sort of decision builder in which decisions are made in a random way. When an ant stumbles upon an obstacle it tends to avoid it and circle around it. It is as if the ant has the capability to estimate the feasibility of a path. Therefore, we enhance this behavior of assessing paths feasibility through equipping an estimator in each ant. Thus, the estimator and the decision builder are implemented in the ants. And the decision building loop is then modeled through the behavior of the ants in choosing their ways. To summarize the ACO is implemented as shown in figure 3.6.

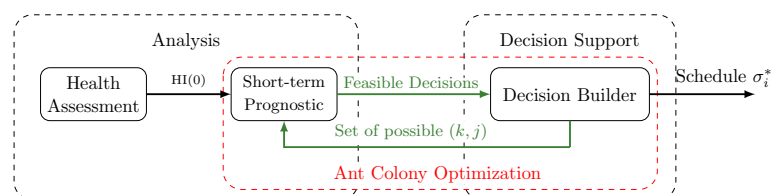


Figure 3.6: Integrating Decision Building Loop in ACO

First, we equip each ant with an estimator. In this application, the estimator is an algorithm that integrated the degradation model of the components plus the time and health constraints. Moreover, the ant, itself, is the decision builder component of the decision building loop. Its objective consists in choosing a sequence of paths between certain tasks (and possible the maintenance workshop) that maximizes the benefits. The chosen paths is rewarded with new quantities of pheromones that are proportional to the benefit of the overall solution.

A cycle of the decision building loop for one ant is presented in figure 3.7. The ant's estimator is updated with the conditions of the machine. Then, it communicate the

information to the decision builder. Now, the ant is aware of the current conditions of the situation (i.e. the current health indicators of the machine and the instant of time). Please note that the term current refers to the specified instant of time relative to the ant and not the current instant of the true system. The ant assess its surroundings (i.e. look through the list of available production orders at that instant). Then, it combines each available order  $p$  from the list with each possible profile  $j$ . Only decisions of scheduling  $(p, j)$  that respects the time constraint of the order  $p$  (i.e. Eq. 3.15) or that still fits in the schedule (i.e. Eq. 3.16) are added to the set of possible schedules. The local decisions  $(p, j)$  are now under construction and they are lacking the health information. Therefore, the list of possible operational local decisions is passed to the estimator. The estimator predicts the outcome of each of the local decisions by performing short-term prognostics. Then, the obtained local decisions (now fully constructed) are assessed against the health constraints and only the feasible local decisions (i.e. none of the final degradation level of the components exceeds the failure threshold) are kept in the feasible set. This behavior is similar to when the ant discards the path with obstacle. The list of feasible decisions is passed to the decision builder in which the maintenance decisions are added to the set and each decision of the set is evaluated. The probabilities of selection are computed based on the cost and the expected benefits of each feasible decisions from the list. Then, a local decision is selected randomly using the roulette technique. This decision is scheduled in the ant's global decision and passed to the estimator to update its machine conditions (i.e. the machine's health indicators and the instant of time). Finally, the estimator checks if the end of the decision horizon is reached. If there is still time another cycle of decision building is initiated. Otherwise, if the end of the decision horizon is reached, or the decision builder could not find any possible elementary action that can fit in the remaining time, then the decision building process comes to an end. The ant then finishes the construction of the global decision by evaluating its different attributes (i.e. cost, gain, benefits, etc).

The advantage of the ACO is that the exploration of other solution is guaranteed with a random factor. The ants chooses their next path according to a selection probability. Each path is characterized with a selection probability that is calculated according to an estimation of its benefits and the quantity of pheromones.

Each ant of the colony is launched independently from other ants. At the end of each cycle the pheromone quantities on the paths is updated according the global decision (schedule) of each ant. Therefore, if the same path is chosen by several ants its pheromone quantity is updated with bigger quantity than other paths. At each cycle, we also keep the best global decision obtained by the ants. With several iterations the ACO algorithm converges to a single global decision with the paths having higher pheromone quantity. Usually this global decision is the best global decision found by all ants during the several iterations.

Once we find the best path, this sequence of decision is applied to the machine. So we obtain the actual degradation level of the components and the cost and benefits over the decision horizon. The list of tasks is updated with the task that are not selected in the previous decision horizon and that are still possible to schedule in the next period. To that list, we add the new tasks that are newly ready. And another cycle starts.

To summarize, each ant goes through several decision building cycles to obtain a global decision. This is done several times (the number of ACO iteration). Once the best global decision (path) is obtained, the decision applying loop is executed. The prognostic-decision ACO algorithm is detailed in Algorithm 1.

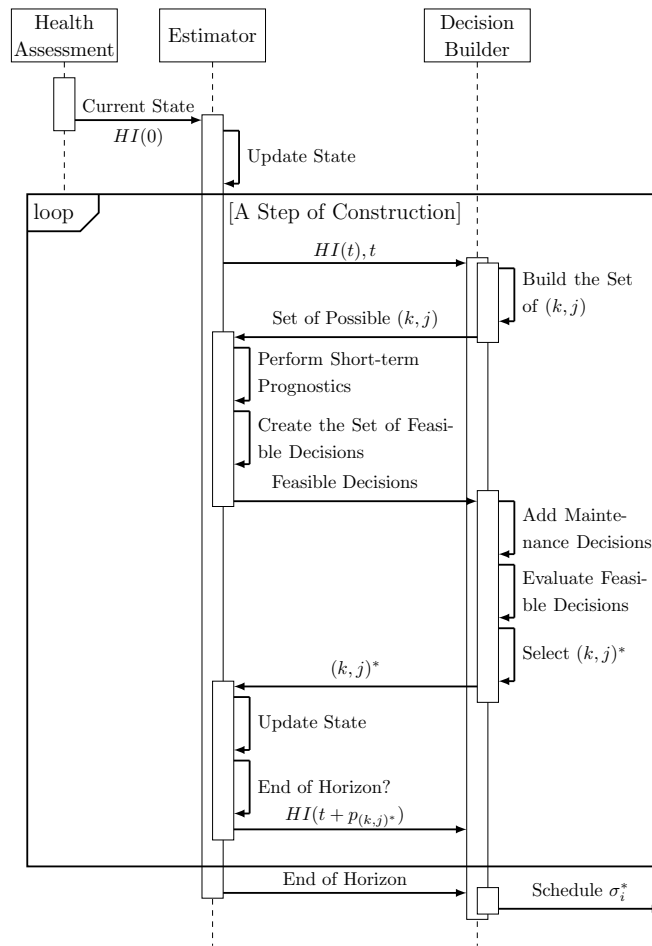


Figure 3.7: ACO's Decision Building Sequential Diagram

### 3.4/ Numeric application

In this section, we present the validation of the proposed algorithms by applying them on a case study. We start by defining the different parameters of the study case. Then results are presented in two sections: (i) we compare the performance of the PHM+ method to common CBM methods and (ii) we study the use of PHM methodology in the single machine context.

#### 3.4.1/ ACO parameters' tuning

The ant colony optimization is known as a tricky method to configure. The algorithm presents several components that influence its convergence speed. These parameters are:

- the number of ants that constitute the colony,
- how many cycles these ants are sent out to search for the suitable solution,
- the evaporation rate of the deposited pheromones, and

**Algorithm 1:** ACO with Local Prognostics for Production and Maintenance Planning

---

```

1 Tasks  $\leftarrow$  [ ];
2 for  $i \in SH$  do
3   Schedule  $\leftarrow$  [ ];
4   Best_Global_Decisions  $\leftarrow$  [ ];
5   Tasks  $\leftarrow$  Tasks + Ready_Orders( $i$ );
6   for  $cycle < Cycles\_Limit$  do
7     for  $ant \in Colony$  do
8       ant_virtual_machine  $\leftarrow$  copy( $[H_1(0), H_2(0), H_3(0)]$ );
9       ant_tasks  $\leftarrow$  copy(Tasks);
10      ant_schedule  $\leftarrow$  [ ];
11       $t \leftarrow 0$ ;
12      while  $t < DH_i$  do
13        Possible_Local_Decisions  $\leftarrow$  Find_Decisions( $t$ , ant_virtual_machine,
14          ant_tasks) ;
15        Evaluate Possible_Local_Decisions; Compute Selection Probability of
16        Possible_Local_Decisions;
17        ant_schedule  $\leftarrow$  ant_schedule +
18          Roulette_Selection(Possible_Local_Decisions);
19        Update  $t$ , ant_tasks, ant_virtual_machine;
20      Find Best Schedule  $\forall ant \in Colony$ ;
21      Add Best Schedule to Best_Global_Decisions;
22      Update Pheromone Quantities ;
23      cycle++;
24    Schedule  $\leftarrow$  Best of Best_Global_Decisions;
25    Apply Schedule;
26    Update Tasks;
27     $i++$ ;

```

---

- the quantity of pheromones an ant can deposit while going back to the colony (i.e. once the schedule is built).

The most important variables are the number of ants and the number of cycles. These two variables define the compromise between the exploration of the solution space and the convergence of the algorithm. Moreover, these variables have a direct impact on the execution time of the algorithm and a good compromise should be set. In the next paragraph, we present a study of the number of ants and of cycles to be used in our problem. We also study the convergence of the algorithm.

### 3.4.1.1/ ACO Parameters

The number of ants and the number of the ants' cycles are two major parameters in the ant colony optimization algorithm. These two parameters are responsible for the solution space exploration and the convergence of the algorithm. We propose to study the influence of these two parameters on the benefits of one period of the simulation horizon. For this study, we take the decision horizon (i.e. the period) duration equal to one day. We are

interested in knowing how much time does it take the algorithm to finish the optimization for one period with the chosen parameters (i.e. the number of ants and cycles). The execution time as function of both parameters is represented in figure 3.8.

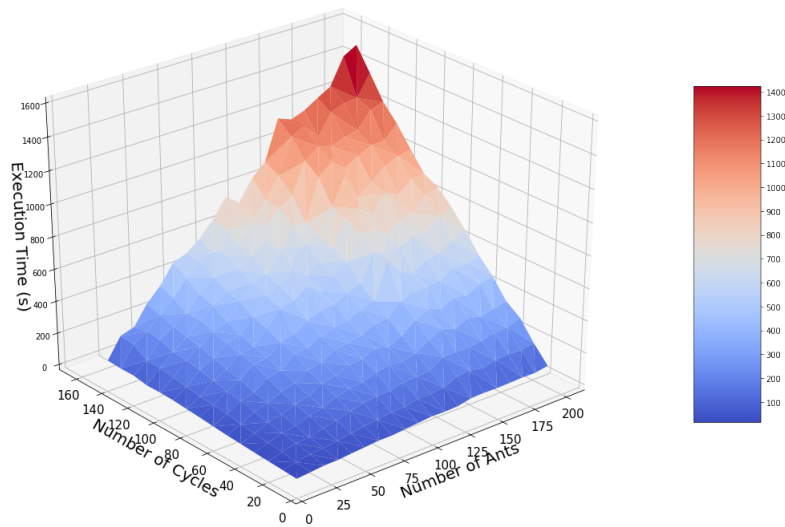


Figure 3.8: Execution Time =  $f(\text{Number of Ants}, \text{Number of Cycles})$

We can note that the execution time seems to grow linearly as the number of ants and/or the number of ants' cycles increase.

We present, in figure 3.9, the evolution of the benefits of the obtained solution over one period as a function of the number of ants and their cycles.

We can note that the benefits of the period increases as the number of the ants and/or the number of cycles increase. This results is expected because by increasing the number of ants and/or cycles we increase the search in the solution space which increases the chance of an ant finding a better schedule than what has been already found. We can also note that starting from 90 ants and 90 cycles, the benefits of one period start to form a horizontal plan, and the variation of the obtained benefits seems to be minimal. It's like starting from the point (90,90) the ant colony optimization stabilizes.

Based on the execution time performance and the obtained benefits over one period, we propose to set the ACO parameters as follows:

- Number of ants = 100 ant
- Number of cycles = 100 cycles

We choose these values because they allow for the algorithm to execute in a duration around 600 seconds for a benefits that is close to the best value. Setting values below the chosen ones could result in a solution with a benefit that is too much below the optimal one. On the other hand, choosing values that are above 100 would raise the execution time.

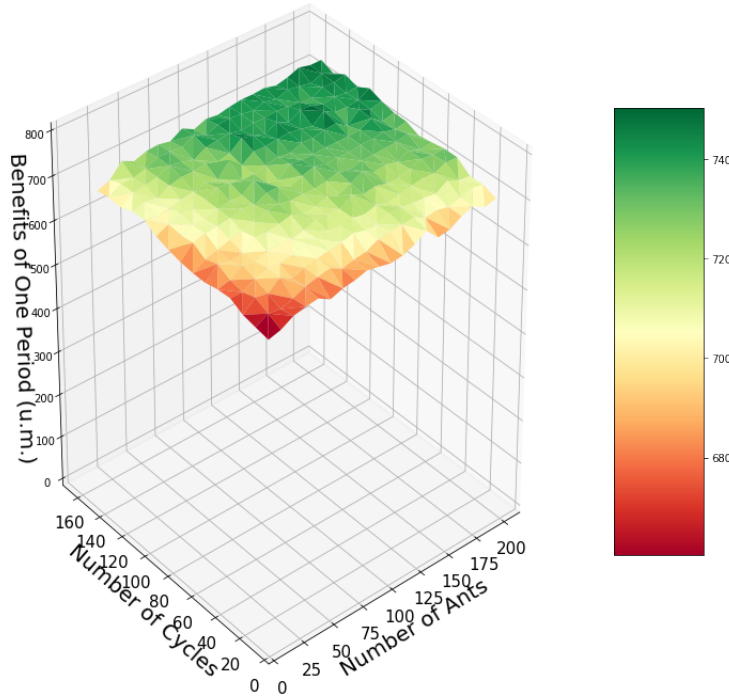


Figure 3.9: Benefits  $B_i = f(\text{Number of Ants}, \text{Number of Cycles})$

### 3.4.1.2/ Decision Horizon

In Section 2 of this chapter, we presented the configuration of the loops that are used in this first application. We also set the value of the partial decision to be equal to the decision horizon. However, the decision horizon value was not discussed. Indeed, we believe that the decision horizon value should be studied for each problem. There is no magic value that would be effective for every problem. The value (duration) of the decision horizon is dependent of several factors such as the chosen decision method (algorithm), the characteristics of the systems (e.g. number of machines, number of components, degradation speed of components, etc), and the characteristics of the problem (e.g. assumptions). In this first approach, we only focus on defining the decision horizon while considering the chosen decision algorithm and the assumptions of the problem. The influence of the systems characteristics are the subject of study in the next chapter.

We start the investigation of the influence of the decision horizon on the performances of the decision algorithm by defining the evaluation process. We propose to solve the joint problem for a duration of 60 days that we refer to as the simulation horizon. This simulation horizon is divided into smaller time duration i.e. the decision horizon. For each decision horizon, we solve the problem of the joint maintenance and production using the proposed ACO method. We obtain a benefits value for each decision horizon that we sum to obtain the cumulative benefits over the simulation horizon as defined in Eq. 3.11.

We propose that the value of the decision horizon is a divider of the simulation horizon. We execute the ACO for each case 10 times. Then, we evaluate the decision horizon upon the average value of the time of execution, the cumulative total benefit, and the proportions of the operational profile used.

Figure 3.10 presents the evolution of average execution time as a function of the duration of the decision horizon. We can see that the time of the execution increases almost exponentially for an increasing duration of the decision horizon.

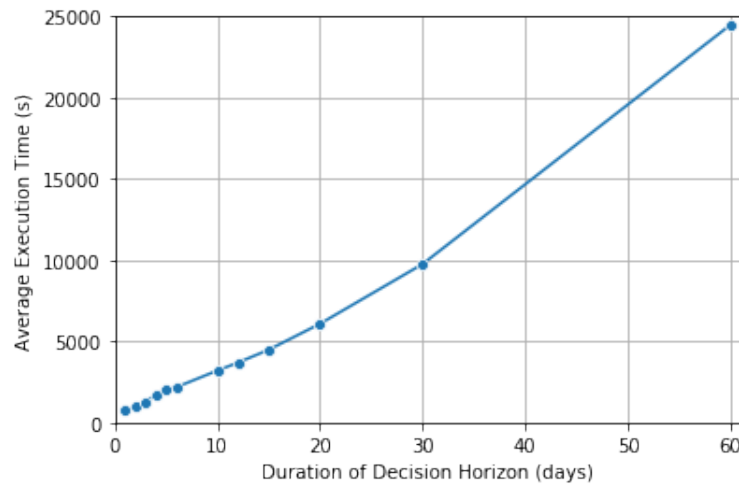


Figure 3.10: Average Execution Time =  $f(\text{Duration of Decision Horizon})$

Figure 3.11 presents the evolution of the average cumulative total benefits as a function of the duration of the decision horizon. We can note that this total cumulative benefits decreases as the number of days in the decision horizon increases. This result could also be related to the choice of the number of ant and the number of cycles. When the number of days in the decision horizon increases, the number of possibilities of orders increases exponentially, and the ACO seems to struggle with the increasing number of possibilities which in translated in a loss of performance.

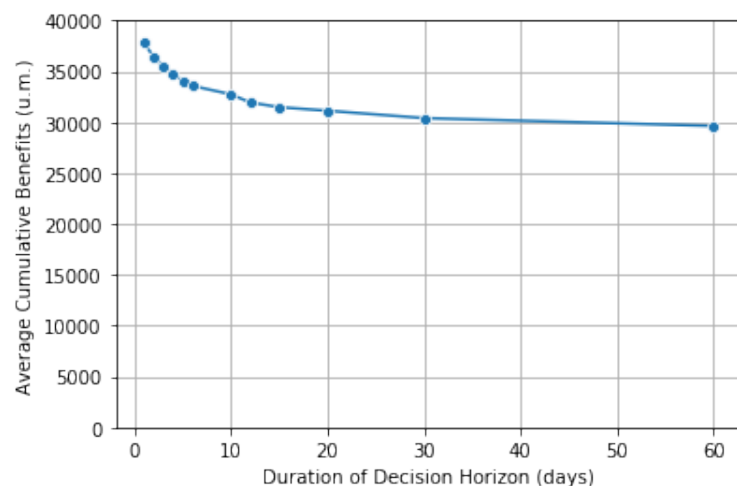


Figure 3.11: Average Cumulative Benefit =  $f(\text{Duration of Decision Horizon})$

Figure 3.12 presents the evolution of average % of how many orders are scheduled with each speed as a function of the duration of the decision horizon. We can see that the percentages of the speeds are initially different but they all tend to converge to 33% which corresponds to a uniformly distributed orders on the possible profiles. This result support the previous one. The ACO tends to loose its performance as the decision horizon

duration increases. This decrease in the performance is mainly related to the exponentially increasing number of possibilities. This allows us to conclude that the ACO is only suitable for problem with low number of possibilities.

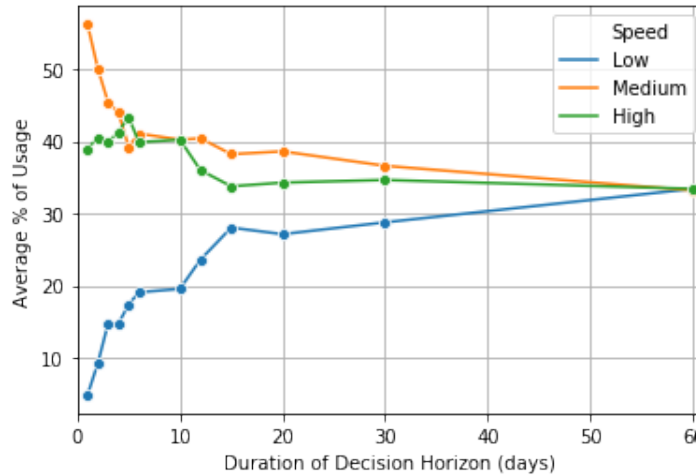


Figure 3.12: Average Execution Time =  $f(\text{Duration of Decision Horizon})$

The problem, in its definition, impose that the production orders could not be interrupted and resumed later (no preemption is allowed). Thus, at the beginning of each period, we have a new schedule that is quite independent from the previous one. Moreover, the performance of the ACO seems to be at the top for a decision horizon equal to a single period duration. Therefore, for this application, we set the decision horizon duration for the decision method (i.e. ACO) equal to the duration of a single period.

### 3.4.1.3/ ACO Convergence

The final step in the setting up of the ant colony optimization is to validate the chosen set of parameters. The ACO is based on a probabilistic selection of feasible local decisions. From an implementation point of view, the generation of random number is influence by the random seed of the computer. Therefore, we launched a 10 executions of the same period with the same system health conditions and set of production orders. We, then compared the obtained results with the maximum obtained benefits. The evolution of the benefits of the solutions as a function of the cycle number is represented in figure 3.13.

The differences in percentage of obtained results are presented in figure 3.14.

We can see that in the worst case scenario the obtained solution is only 1.6% away from the best one. We conclude that this level of variation is acceptable for our problem. And we validate the proposed settings.

### 3.4.2/ Adapting conventional CBM methods

The proposed prognostics-based method is compared to classic methods in which the maintenance is planned with condition-based maintenance framework and the production is scheduled with either a deterministic heuristic (Moore's heuristic) or with genetic algorithm.



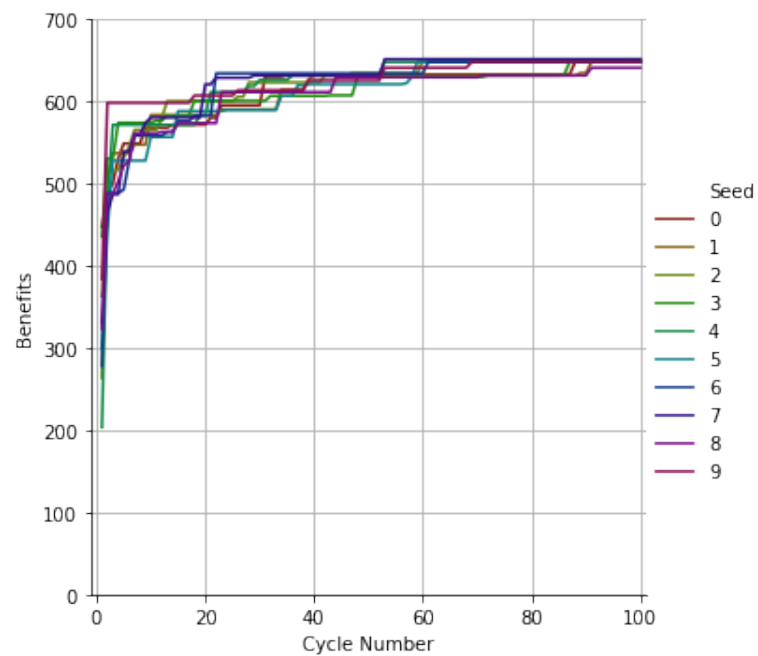


Figure 3.13: Evolution of the Benefits of the Best Schedule

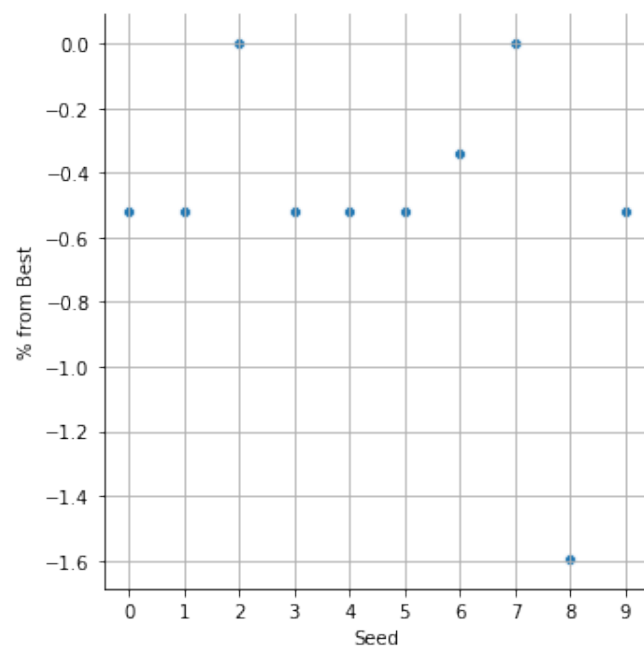


Figure 3.14: ACO Convergence Validation

#### 3.4.2.1/ Moore with condition-based maintenance

Thus we need to minimize the number of tardy jobs. In this context, Moore [135] developed an algorithm that minimizes the number of tardy tasks in the case of a single machine with non-preemptive tasks.

Since our objective is to manage the system life and health, it is important to add mainte-

nance decisions. That's why we combined Moore algorithm with a classic CBM algorithm. Where at the start of each period we check if there is a necessity to schedule a maintenance action for the system or not. The need for a maintenance action is determined by comparing the actual degradation level of each component with a threshold. If one or more degradation level exceed the threshold for maintenance a maintenance action is scheduled at the beginning of the period then the rest of the duration is scheduled with Moore algorithm to obtain the global decision, otherwise we directly schedule production. If the system fails before the next health inspection a corrective maintenance action takes place. Corrective maintenance costs more than scheduled maintenance and takes more time. Throughout this chapter, we refer to this method as **M-CBM**.

### 3.4.2.2/ Genetic algorithm with condition-based maintenance

To solve the joint problem, we combined condition-based maintenance with a genetic algorithm. At the start of each period, if the period corresponds to the inspection date, then an inspection of the components degradation level takes place. Components in which the degradation level exceeds the CBM threshold are scheduled for maintenance at the start of the period. Then, the remaining time is scheduled for production using the genetic algorithm (GA). In this paper, this method is referred to as **GA-CBM**. To implement GA, several components should be considered:

- the genetic representation of the solution,
- the fitness function,
- the method to generate initial population,
- the genetic operators (mutation and crossover), and
- the survival rules.

Even though GA does not guarantee the global optimum solution, it is a commonly used method in cases of combinatorial, high instances or non-linear optimization problems [38]. The required components of the GA implementation are presented in this subsection.

**3.4.2.2.1/ Solution representation** The algorithm search for a good combination of orders and the used profile to produce them. Then, we propose a coding consisting of a 2D matrix composed of two lines of integers. The first line contains the sequence of orders, while the second line contains the production profile to be used. Then each allele is composed of the order index  $k$  and a production profile  $j$ .

**3.4.2.2.2/ Fitness function** Individuals in the GA are evaluated to measure their fitness toward an objective function. Here, the fitness evaluation uses the benefit function defined by Equation 3.7. The evaluation is obtained through simulating the outcome of the chromosome (i.e., we simulate the application of the individual's schedule to the machine and we evaluate the outcome cost).

**3.4.2.2.3/ Initial population** GA requires an initial generation (or set) of valid solutions. Here, a solution needs to verify some constraints to be considered as valid. These constraints consist of having a schedule with a duration that does not exceed the duration of a period and in which each order has a completion time lower than its deadline (see Equations 3.15 and 3.16). The individuals of the initial population are built randomly by combining orders and production profiles then scheduling them in random order. Invalid individuals are modified to guarantee that the constraints are respected.

$$\sum_{(p,j) \in Sch_i} p_{p,j} \leq DH_i \quad \forall i \in SH \quad (3.15)$$

$$c_{p,j} \leq D_p \quad \forall (p,j) \in Sch_i \quad (3.16)$$

**3.4.2.2.4/ Genetic operators** The proposed GA uses two genetic operators: mutations and crossover. For the mutation, we proposed two operators: (i) a single point mutation operator in which a randomly selected allele is mutated by exchanging the current order with another order from the list of unscheduled available orders, and (ii) random two-point mutation operator in which two alleles are selected and are exchanged. In this algorithm, we used a classic order crossover (also noted OX crossover).

**3.4.2.2.5/ Survival rules** New generations are created from the previous generation survivors and genetic operators' offspring. Here, a novel generation is built by; (i)  $X_{Survival}\%$  of the best chromosomes from previous generation, (ii)  $X_{Mutation}\%$  of the mutation's offspring, and (iii)  $X_{Crossover}\%$  of the crossover's offspring. The offspring selection is based on a roulette wheel process using the fitness value for defining the selection probability. Since the objective is to maximize the fitness function, the probability of each individual can be defined as in Equation 3.17.

$$Prob(Indiv_i) = \frac{fitness(Indiv_i)}{\sum_{Indiv \in Offspring} fitness(Indiv)} \quad (3.17)$$

The overall algorithm of the GA-CBM is presented in Algorithm 4.

### 3.4.3/ PHM+ vs. CBM

In this section, we propose to compare the general performance of the proposed decision-enhanced PHM method and the classical combinations of heuristics, respectively meta-heuristics, with condition-based maintenance. For this purpose, we defined a set of performance indicators:

- the total cost over the simulation horizon,
- the total benefits over the simulation horizon (the objective function),
- the proportions of missed and done production orders,
- the quantity of produced goods,

**Algorithm 2:** GA with CBM for Production and Maintenance Planning

---

```

1 for  $i \in SH$  do
2   if  $i == Inspection\_Date$  then
3     for  $\forall component\ k$  do
4       if  $H_k \geq CBM_k$  then
5         Schedule Component  $k$  for Maintenance;
6   Create Initial_Generation ;
7   Input_Generation  $\leftarrow$  Initial_Generation;
8   while  $number\_of\_generations < Generation\_Limit$  do
9     Evaluate Individuals of the Input_Generation;
10    Sort Input_Generation ;
11    Survival of the best  $X_{Survival}\%$  Individuals of the Input_Generation;
12    Mutation_Results  $\leftarrow$  [ ];
13    Crossover_Results  $\leftarrow$  [ ];
14    for  $Individual \in Input\_Generation$  do
15      Generate a Random Numbers  $a, b$  and  $c$ ;
16      if  $a < Simple\_Mutation\_Probability$  then
17        Generate a random number  $pos$  for position;
18        Simple_Mutate Individual at position  $pos$ ;
19        Add the obtained solution to Mutation_Generation;
20      if  $b < Exchange\_Mutation\_Probability$  then
21        Generate two random numbers  $pos1$  and  $pos2$  for positions;
22        Exchange_Mutate Individual at positions  $pos1$  and  $pos2$ ;
23        Add the obtained solution to Mutation_Results;
24      if  $c < Crossover\_Probability$  then
25        Generate one random numbers  $pos$  for position;
26        Select another parent from Initial_Generation;
27        Crossover Individual and the other parent at positions  $pos$ ;
28        Add the obtained solutions to Crossover_Results;
29    Evaluate Individuals of the Mutation_Results;
30    Sort Mutation_Results;
31    Selection of the best  $X_{Mutation}\%$  Individuals of the Mutation_Results;
32    Evaluate Individuals of the Crossover_Results;
33    Sort Crossover_Results;
34    Selection of the best  $X_{Crossover}\%$  Individuals of the Crossover_Results;
35    Input_Generation  $\leftarrow$  Output_Generation;
36    number_of_generations++;
37  i++;

```

---

- the number of corrective maintenance interventions,
- the proportions of time the machine has spent in production (noted  $Prod(\%)$ ), in maintenance (noted  $Maint(\%)$ ), and idle (noted  $Idl(\%)$ ) over the simulation horizon, and
- the execution time of the method (it contains the building of the decisions and its

simulation).

For this purpose, we generated several machine configurations. All these configurations present a three components machine. The components degradation rates and initial conditions differ from one configuration to another. The results are classed by the tasks sizes. Each of these configuration is used for 10 task sets per category of tasks. We considered three categories of tasks:

- large tasks in which the orders quantities are between 100 and 200 product per task,
- medium tasks in which the orders quantities are between 30 and 100 product per task, and
- small tasks in which the orders quantities are between 10 and 50 product per task.

Each of the performance indicators presents a single value for each category of tasks. This value is defined as the mean of values obtained from the simulation of several combinations of machine configuration and the task sets over the simulation horizon. Since Moore algorithm could only function with a single profile of production. The presented results are compared for a single profile of production i.e. normal speed speed.

Table 3.8 presents the obtained results grouped by task's size. We can see that in all cases the Moore-CBM method is the fastest to find the mixed schedule. However, the performances of this method remains relatively bad compared to other method. It causes a higher number of corrective maintenance and low quantity of products.

When observing the total benefits and the percentage of production time, one can notice that the genetic algorithm combined with condition-based maintenance and the decision-enhanced PHM method using ant colony optimization are quite close in the performance. In terms of cost-benefits objective, the performance of the methods changes over the size of tasks. In small task the PHM+ methods out performs the GA-CBM. The gap is smaller when the tasks are of a medium size. Finally the GA-CBM becomes better with bigger tasks. We can see that this difference is mostly caused by the maintenance planning. We can see that the produced quantities for small and medium tasks are quite similar for both methods (with PHM+ producing slightly more). However, the mean number of corrective maintenance is quite different between the methods. In PHM+, no corrective maintenance is needed due to the selection of the only possible tasks with prognostics information (and a deterministic degradation model) while the GA-CBM remains uncertain about the machine capacity to achieve all tasks before reaching the failure threshold. This difference in the corrective maintenance interventions explains the difference in the production, maintenance, and idle proportions for the small and medium tasks. When it comes to large task, the GA-CBM outperforms the ACO in the production scheduling process. This explains the bigger quantity of produced elements. Plus the difference in the number of corrective maintenance is reduced. This explains why the GA-CBM out performs the PHM+ in larger tasks. But the execution time of the PHM+ remains lower than the GA-CBM and the difference becomes even bigger with larger tasks. This is mostly related to how these methods are different in exploring the solutions space and the mechanism of their convergence.

These results, especially when PHM+ is not always the best method, drive us to question about the cases when it is more interesting to invest in using PHM methodology and when a condition-based maintenance is sufficient. When reading works in literature,

Table 3.8: Methods comparison

Task Category	Small			Medium			Large		
	PHM+	GA-CBM	M-CBM	PHM+	GA-CBM	M-CBM	PHM+	GA-CBM	M-CBM
Method	PHM+	GA-CBM	M-CBM	PHM+	GA-CBM	M-CBM	PHM+	GA-CBM	M-CBM
Total Cost	46350	59898	55015	50410	60432	55429	53849	62968	55816
Total Benefits	30115	19713	17340	25680	20757	14304	20036	21936	9890
Done(%)	91	92	93	88	87	87	79	76	74
Missed(%)	9	8	7	12	13	13	21	24	26
Quantity Produced	34360	34259	31782	34167	34028	31277	32880	33842	29660
Corrective Maintenance	0	1,12	1,49	0	0,45	1,79	0	0,03	1,46
Prod(%)	95,7	95,19	91,45	95	94,54	88,31	91,33	94,01	82,95
Maint(%)	3,78	4,48	4,39	3,68	4,44	4,42	3,79	4,46	4,12
Idle(%)	0,52	0,33	4,16	1,32	1,02	7,27	4,88	1,53	12,93
Execution Time (s)	472	881	3	107	817	3	45	706	3

authors always tend to present their method as the ultimate solution and they do not make a comparison with other methods. Very few works have compared their results to the different maintenance policies to select the suitable one. For example, Van Horenbeek and Pintelon in [173] proposed a prognostic-based predictive maintenance policy, the results of their method are compared to several maintenance policies such as condition-based maintenance, and classic age-based policies. They also studied the influence of some parameters like the dependencies between the components and the prognostic horizon on the policies' performances. Hence, providing readers with a methodology to select a suitable maintenance policy. However, the authors have only done this study for a particular machine configuration. In other words, one can conclude on the choice of the optimal policy only if a similar machine configuration is present. Therefore, the question of which policy is more suitable for other configurations remains unanswered.

Moreover, the work of Van Horenbeek and Pintelon, like many other works, has not included the production scheduling and how it can influence the performance of the maintenance policy. The works on CBM or PHM maintenance scheduling implicitly assume that the operating conditions of the machines and their future loads are constant such as the works of Camci [27], Shi and Zeng [158], and Langeron *et al.* [103].

Therefore, in the next section, we propose a study on the use of prognostics information and decision-enhanced PHM in the case of a single multi-components multi-purpose machine.

#### 3.4.4/ Investigating the use of PHM+

To study the selection of the adequate method, we need to compare the two methodologies in several test cases that take into consideration different factors. In this application, the considered factors are:

- The degradation's speed of the component, three categories are considered: (i) rapidly deteriorating, (ii) normally deteriorating, and (iii) slowly deteriorating. The speed of the deterioration is determined by the component's coefficients  $a_k$  and  $b_k$ . These components are used to create four machines as presented in Table 3.9.
- The initial condition of the components. Four sets of initial conditions are considered: (i) all components are new, (ii) components are at the first half of their life with random degradation level, (iii) a mix of components degradation levels one new, one used, and on at the end of its life, and (iv) all components have the same degradation level and they are at the second half of their life. These initial conditions are presented in Table 3.10.
- Three categories of production orders were considered. The categories are defined according to the quantity of the demanded product; small tasks where  $Q \in [10, 50]$ , medium tasks where  $Q \in [30, 100]$  and large tasks where  $Q \in [100, 200]$ .
- Product type: to capture the influence of the product type, we considered two cases: (i) 5 sets of production orders that contain three different type of products, and (ii) 5 sets of orders that have the same unique type of product.
- Production profile: to capture the influence of the production speed, we defined two cases: (i) the machine can only produce with one speed, and (ii) the machine can be used with three different profiles (i.e. three speeds; low, normal, and high).

Table 3.9: Machines configurations

Machines	Components		
	k=1	k=2	k=3
$M_1$	rapid	slow	rapid
$M_2$	slow	normal	rapid
$M_3$	rapid	rapid	rapid
$M_4$	slow	slow	slow

Table 3.10: Initial conditions configurations

Initial Condition	Degradation Level		
	D1	D2	D3
IC1	0	0	0
IC2	0.3	0.4	0.24
IC3	0.8	0.4	0
IC4	0.6	0.6	0.6

For each machine, we consider 4 initial conditions for 3 categories of production orders each presents 10 set of production orders (5 with single product, and 5 with multiple products). This leads to a total of 120 test cases for each machine when using a single speed and 120 test cases when using a three speeds.

Tables B.1 and B.2, in the appendix B, present the obtained results respectively for the single-speed case and the multiple profiles. Each row of these tables presents for a specific machine the mean of benefits over 20 test case (i.e., 5 production orders sets and 4 initial conditions). All the values in these tables are expressed in k u.m. (thousand units of currency). These results are summarized in figure 3.15 in which we present the difference between the benefits obtained using PHM+ and those obtained using CBM. The results are presented by type of product, number of used operational profiles, and the number of machine.

One can easily note that the benefits are more important for slow deteriorating machine ( $M_4$ ) compared to rapid deteriorating machine ( $M_3$ ). The rapid the degradation of the component gets, the more frequently it undergoes maintenance and the higher the maintenance cost gets which causes the benefits to get lower.

One can note that the difference between PHM+ and CBM is always positive for the small tasks. In other words, it is more beneficial to use PHM+ in this case. By comparing the values of the difference between different machines, one can notice that the benefits of using PHM+ reaches its maximum for the machine  $M_3$  which is composed of only rapid deteriorating component. Furthermore, the benefit of using PHM+ has a minimum value for the machine with the slowest deteriorating components. Machines with slow deteriorating components are more easy to control from a health management perspective. Although the variation of future decisions influences the evolution of the degradation, the machine's states do not present a big variation between two inspection dates. Therefore, the use of PHM+ does not produce a big difference.

When the tasks get larger, the machine is spending more time producing the same product with the same operational conditions. Thus, the variation of future decisions is reduced. One can notice that the reduction of future decisions' variation reduces significantly the



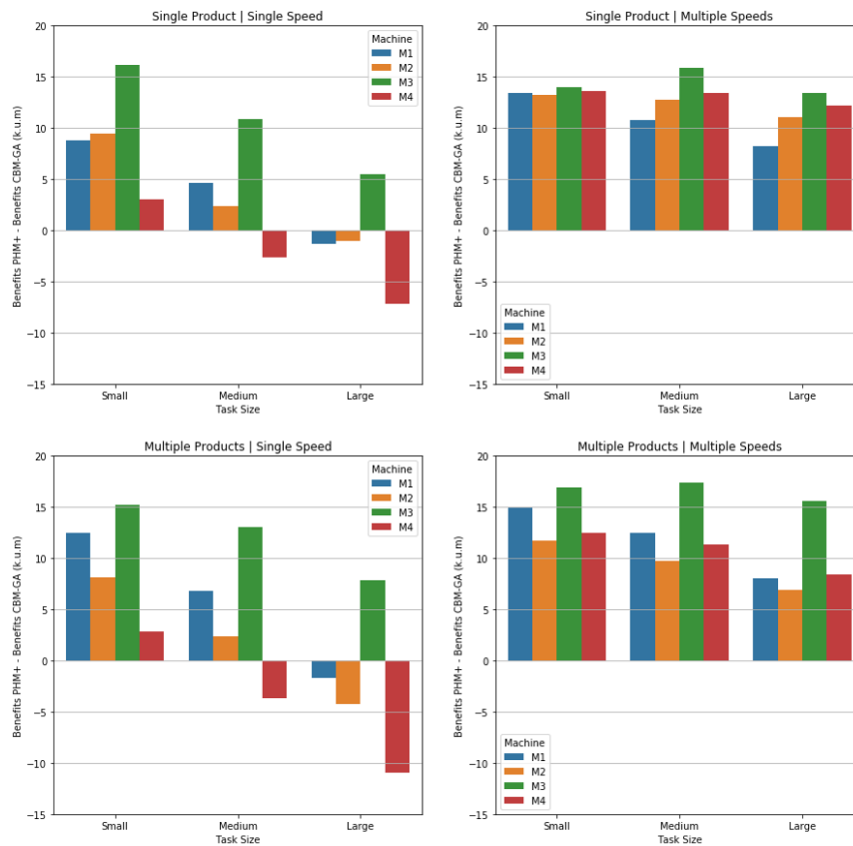


Figure 3.15: Benefits of Using PHM+ vs CBM

benefits of the PHM+ on one hand. On the other hand, the benefits of the CBM are almost stable. Therefore, the difference between the two methods decreases.

The decrease of the PHM+ benefits is more important for slow degrading machines. In the case of medium tasks, one can note that the difference is negative meaning that it is more beneficial to use CBM. The values of the PHM+ benefits continue to decrease when orders get larger. For large tasks, one can notice that only machine M3 has a positive difference. In cases in which the machines have different components' degradation dynamic or only slow deteriorating components, the CBM method has the biggest benefits.

One can conclude that when the machine's overall degradation dynamic becomes slow, the PHM+ benefits decrease. Thus, for slow degradation machines, it is more beneficial to use CBM. Furthermore, When the variation of the future loads decreases the PHM+ benefits decrease and the CBM becomes more practical than PHM+. These observations allows us to conclude on the use of PHM+ or CBM with genetic algorithm in the case of a single speed machine. The obtained classification is presented in Figure 3.16.

Table B.2 presents the difference in benefits between using PHM+ and CBM for a single machine composed of three subsystems capable of functioning with several production profiles. The table presents the results for a single product type and multiple products.

One can notice that for these test cases the benefits of using PHM+ are more important than using CBM even for large tasks. When using several production profiles, the variability in the degradation evolution is higher. This variability cannot be detected with CBM, in this case.

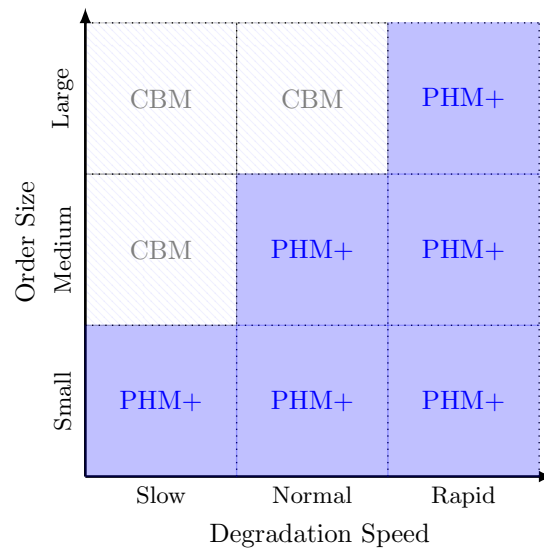


Figure 3.16: Use map of PHM+ vs CBM for a single speed test case

However, one note that the degradation dynamic has the same effect as in the case of a single speed machine. The difference between the two methods is maximized for machines with rapidly deteriorating components. One can also note that the smaller the tasks gets the higher the difference between the two methods. Therefore, the variability of the production orders has the same effect as the production profiles.

To conclude, the higher the variability in the machine's operational conditions the more beneficial it is to use PHM+. The operational conditions are varied using:

- different product types,
- different production orders, and
- different production profiles.

One can also conclude that the machine parameters especially the degradation dynamic of its components influences the method selection for the joint optimization problem.

## 3.5/ Discussion

### 3.5.1/ Limits

Even though the described decision problem is more realistic from previous works in the literature, this formulation is still distant from a real world application. The main limits in this application are as follows:

- The machine in this application is assumed to be multi-component in a simple serial structure. While in real life situations, we find complex scheme for the components dependencies.

- Every component is critical to the machine's functions. However, in real life, some of the trains components are not critical to its operations. And, their failure will only cause user discomfort and hence introduce some cost penalties.
- Moreover, some components are installed in a redundant schemes which allows a more flexible corrective maintenance situation without introducing cost penalties.
- In real life, components are highly dependent one another, we know that the degradation of one component has an influence on the degradation evolution of another component. Such dependencies should be accounted for when dealing with PHM problems.
- In some cases, stopping the machine to perform maintenance on two components can cost less and take less time than maintaining a single component at the time even if it is a premature maintenance. Therefore, one should consider maintenance grouping strategies.
- The use of deterministic degradation models seems good for a first application of the framework to serve as a proof of concept. However, in real life the degradation process is subject to several uncertainties.

Moreover, the framework instantiation presented is also quite limited. The limits of the implementation of the decision-enhanced framework can be summarized as follows:

- In PHM projects, the degradation modeling and the extraction of health indicators are quite challenging and not a straight forward process.
- Also, the development of a short-term prognostics algorithm could be a challenge in itself.
- The proposed approach is based on a prediction of health indicators for each available task. For small problems this could be feasible. However, with large sets of production tasks the complexity and computation efforts would grow exponentially. And, at some point it would be absurd to compute all the combination.
- The results concerning the study of method selection are obtained for a certain degradation model and a specific application parameters (i.e. costs of production, maintenance, etc...). These results are not generic to any machine because of the assumption this application was built on.
- The passage from slowly deteriorating component to a normal degradation speed to a rapid deteriorating component is quite subjective and no threshold are set in this study. This classification is quite relative to the application, the system, their complexity, and how critical they are.

Nevertheless, this research work is a step ahead of post-prognostics decisions maturity in production workshop context.

### 3.5.2/ Results Summary

To summarize, the integration of the decision-enhanced PHM seems to improve the performances of single machine workshops. However, the application is quite sensitive

to the proposed parameters and costs. According to the objective of the workshop, the methods parameters could be readjusted to emphasize on production or maintenance services.

The proposed ant colony algorithm showed promising results when compared to classical methods of resolution for the joint problem. However, in some cases, the performance of the algorithm are a little behind genetic algorithms. This observation raised an important question about when it is more important to use one method or the other.

We tried to answer this question. For the proposed application, we studied different configurations and problems to see when to use prognostics information and when a condition-based maintenance strategy is sufficient. We noted that the more variance and uncertainty we have in the application the better it is to choose prognostics-based approaches. The variances in the operations and the uncertainties are caused by the type of product, the size of the tasks, and the speed of deterioration of the components.

### 3.6/ Summary

This chapter presents the instantiation and implementation of the proposed post-prognostic decision framework on a first case study. The application consists of solving the joint problem of maintenance and production scheduling for a single multi-component multi-purpose machine inspired from computer numerical control (CNC) machine. The focus of the case study is to implement the decision building loop of the proposed framework. For this aim, an ant colony optimization algorithm is used in which each ant is considered as a decision builder module and equipped with its own estimator. This method is compared to classical heuristics and meta-heuristics combined with condition-based maintenance. On average, the proposed implementation of the framework presents better results than the other classical methods featuring CBM. However, one can notice that in some cases the classic CBM method seems to be more beneficial. Hence, a study on in which cases it is more interesting to use prognostics-based method is conducted. The obtained results show that the higher the variability in the machine's operations the more beneficial the proposed approach becomes. Three variability factors were studied: (i) the degradation dynamics, (ii) the size and type of the production orders, and (iii) the possible production profiles. The chapter ended on a discussion of the important results and some perspectives for future investigation. One question could be asked at this stage: with the framework validated for a single machine test case how could one integrate and implement the decision-enhanced PHM in the context of multi-machines? In the next chapter, we answer this question by studying the application of the framework on a fleet of trains.

## Chapter 4

# Railway Assets Mission Assignment & Maintenance Scheduling

### Contents

---

<b>4.1</b>	<b>Decision problem definition . . . . .</b>	<b>107</b>
4.1.1	Related works and Motivation . . . . .	107
4.1.1.1	Related works . . . . .	107
4.1.1.2	Synthesis . . . . .	109
4.1.1.3	Motivation . . . . .	109
4.1.1.4	Contributions . . . . .	110
4.1.2	Scope of the study . . . . .	110
4.1.2.1	Hypothesis . . . . .	111
4.1.2.2	Constraints . . . . .	111
4.1.3	System description . . . . .	112
4.1.3.1	Rolling stock fleet structure . . . . .	112
4.1.3.2	Degradation model . . . . .	112
4.1.4	Problem description . . . . .	113
4.1.4.1	Vehicle scheduling problem . . . . .	114
4.1.4.2	Maintenance problem . . . . .	116
4.1.4.3	Joint Problem . . . . .	117
4.1.5	Decision Horizon Study . . . . .	117
<b>4.2</b>	<b>Instantiation of the proposed framework . . . . .</b>	<b>118</b>
4.2.1	System definition . . . . .	118
4.2.2	Application definition . . . . .	120
4.2.3	Elementary Actions . . . . .	122
4.2.4	Local Decisions . . . . .	123
4.2.5	Global Decisions . . . . .	124
4.2.6	Loops settings . . . . .	125
<b>4.3</b>	<b>Implementation of the decision module . . . . .</b>	<b>127</b>
4.3.1	Genetic Algorithm . . . . .	127
4.3.1.1	Solution Representation . . . . .	128
4.3.1.2	Initial Generation . . . . .	129
4.3.1.3	Genetic Operators Results . . . . .	129
4.3.1.4	Mutation . . . . .	130
4.3.1.5	Crossover . . . . .	131
4.3.1.6	Fitness Evaluation . . . . .	131
4.3.1.7	Construction of New Generation . . . . .	132
4.3.1.8	Overall Algorithm . . . . .	132
4.3.2	Heuristics . . . . .	132

4.3.2.1	H1: . . . . .	133
4.3.2.2	H2: . . . . .	135
<b>4.4</b>	<b>Numeric application . . . . .</b>	<b>137</b>
4.4.1	Case study . . . . .	137
4.4.2	Methods Comparison . . . . .	139
4.4.3	Decision Horizon Duration Study . . . . .	142
4.4.3.1	Genetic Algorithm . . . . .	143
4.4.3.2	Heuristic H2_v1 and H2_v2 . . . . .	144
4.4.3.3	Synthesis . . . . .	145
4.4.4	The effects of the problem's size on the decision horizon . . . . .	146
4.4.5	The effects of the trains' configuration on the decision horizon . . . . .	146
<b>4.5</b>	<b>Discussion . . . . .</b>	<b>148</b>
4.5.1	Limits . . . . .	148
4.5.2	Results Summary . . . . .	150
<b>4.6</b>	<b>Summary . . . . .</b>	<b>151</b>

---

On one hand, maintenance of moving systems differs from traditional maintenance problems, e.g. in a production context, in that the location of these systems over time depends directly on their use and on the missions they have been assigned to. On the other hand, if missions are assigned with respect to the system's health condition, the effectiveness of the missions could be maximized. Researchers have proposed to jointly solve the mission assignment and maintenance planning for rolling stocks, e.g. trains in an iterative decision cycles. In the prognostics and health management context, the assignment of missions to rolling stock units is supposed to take into consideration the predictive information of the systems. Thus, the assignment is dependent on the estimated health indicators and/or the remaining useful life. Moreover, we know that assigning a mission to a system could change the degradation's evolution and hence the prognostics information. This inter-dependency between the prognostics and decision-making module is enhanced for the joint problem. For this purpose, we propose to apply the proposed framework in chapter 2 to solve the joint problem of mission assignment and maintenance scheduling for rolling stocks. The chapter is divided into 4 main sections. The first section is dedicated to define the decision-making problem in the context of rolling stocks management. Second, we apply the proposed framework to this problem formulation. We start by defining the system and the application. Then, we detail the elementary actions, local decision, and the structure of the global decisions. The section is concluded with the set ups of the framework loops. In section 3, we present the developed and used decision methods to implement the different loops of the system in particular the decision-building loop. Section 4 presents a case study in which we compared the performance of the presented methods. Moreover, various factors of the problem definition (e.g. the components characteristics, or the rolling vehicle structure) are modified to investigate their effects on the decision horizon. Finally, some discussion of the proposed approach, the obtained results, and future paths is done at the end of this chapter.

## 4.1/ Decision problem definition

The management of a fleet of systems is a challenging task for the industry, in particular the railway context, in which there is a need to find a compromise between scheduling maintenance and assuring operations. The use of the predictive information in this decision-making process increase the complexity of the task. Railways industries are looking to maximize the usage of the components before replacing them (i.e. reducing the waste in terms of RUL) while guaranteeing a certain level of reliability and avoiding failures. We start by presenting an overview of some related works of mission assignment and maintenance scheduling for rolling stocks. Then, we describe the considered fleet of rolling units and how they are used in this application. Finally, we present the optimization problem statement.

### 4.1.1/ Related works and Motivation

#### 4.1.1.1/ Related works

In the context of applying PHM methodology on rolling stock, several works have studied condition monitoring [131] and prognostics [18, 113]. However, to our knowledge only two papers have addressed the post-prognostics decision-making on rolling stocks. Herr *et*

*al.* [81] presented a linear program to optimize jointly task assignment and the maintenance scheduling of trains. Given predefined train timetables, the algorithm best matches train with the scheduled missions and plan maintenance interventions when needed. The assignment of rolling stock units and the schedule of its maintenance intervention is based on prognostic information to maximize the use of each train in terms of useful life. In this study, the trips of the timetables have different duration and different starting dates. Later, Herr *et al.* [82] fine-tuned the problem definition. In the second paper, the authors considered a set of identical daily trips and the maintenance interventions have been restricted to one intervention per rolling unit during the planning horizon. The problem is then solved for different fleet sizes and horizon duration to demonstrate the influence of these parameters on the performance of the linear programming method. Except for the works of Herr *et al.*, we could not find other works that integrated health information in the management of rolling stock units (trains, trams or metros). Since very few works addressed the PHM decision process on rolling stocks, we extended our literature study to works that considered decision-making on rolling stock regardless of the health information. Two categories of works are found: (i) research that has been conducted on the maintenance planning only and (ii) works that studied the joint problem of mission assignment and maintenance planning. These works are detailed in the following.

The problem of maintenance planning for rolling stocks (e.g. trains or tramways) have been well addressed in the literature. Several works have addressed the problem of scheduling maintenance on rolling units without considered the operations. For instance, Budai *et al.* [22] presented a heuristics to schedule preventive maintenance activities while minimizing the needed time for the interventions. Wang *et al.* [184] used a branch and bound algorithm to solve multiple criteria maintenance strategy selection problem. Cheng *et al.* [33] evaluated and selected a suitable maintenance strategy using an analytic network process technique. The authors, then, estimated the quantities of spare parts for the components and their corresponding replacement intervals.

Moreover, the joint problem of mission/task assignment or vehicle scheduling and maintenance planning has been addressed in the railways context. For example, Giacco *et al.* [64] proposed a mixed-integer linear programming model to solve the joint problem of rolling stock routing and maintenance planning. The authors aimed to minimize the total number of used trains and the number of empty rides and to maximize the distance traveled by each train between two similar maintenance operations. Lai *et al.* [101] used also a mixed-integer programming method to optimize the trips scheduling for a fleet of trains while taking into consideration their needs in terms of maintenance inspections. The presented study is limited to cyclic maintenance inspections of 3-days to 3-months frequencies. Andres *et al.* [6] determined a suitable train routing while scheduling the necessary maintenance interventions using a mixed-integer linear programming model that minimizes the global cost. Lin *et al.* [116] proposed a binary non-linear programming model to solve train mission-assignment and maintenance planning for high-speed trains. The problem consists of assigning well-conditioned trains to each trip while scheduling maintenance on the accumulated mileage or time from the last inspection. Although several frequencies of inspections have been proposed, the authors considered the train as a single component with systematic preventive maintenance requirements. The planning horizon of this study is fixed to one week even though in some cases the duration of the horizon is significantly smaller than the inspection frequency. Zhong *et al.* [200] scheduled rolling stock units based on predefined timetables while considering maintenance restrictions. The authors proposed a two-stage heuristics. At the first stage, a mixed integer programming method



solves the assignment problem while ignoring maintenance constraints. In the second stage, the obtained schedules are checked for feasibility when the maintenance restrictions are added. Mira *et al.* [132] developed an integer linear programming model that integrates preventive maintenance planning within the rolling stock operation schedule. The objective of their work is to find the sequence of trips that minimizes the cost and satisfies the predefined maintenance needs. Both works [200] and [132] considered maintenance interventions as special trips (i.e. trips without passengers) plus trips are assumed to have different starting and ending points.

#### 4.1.1.2/ Synthesis

Most of the published works on integrating maintenance in the rolling stock operation schedules deal with preventive maintenance or systematic cyclic inspections. Although condition monitoring and prognostics and health management technologies are presented in this field, their application and integration in the decision-making on rolling stock either with variable or fixed geographical positions are still missing. To our knowledge, there is a lack of publications on post-prognostics decision-making on rolling stock. Even though the problem described by the works of Herr *et al.* seems fairly realistic, it presents several major assumptions. First, the authors have optimized the joint problem from a maintenance point of view. However, the joint problem is a multi-objective problem since it mixes two disciplines maintenance and operation. Moreover, the authors considered rolling stock units as single-component systems. When in reality, trains are composed of several critical components. Plus, each component presents its health state and its deteriorating speed. Furthermore, the degradation model of the trains is considered deterministic and linear. However, the prognostic algorithm generally provides a distribution of possible future states due to the existence and the propagation of uncertainties. Finally, as mentioned by the authors the generated problems are characterized by a great unnecessary maintenance capacity.

Rolling stock scheduling and maintenance planning are usually done manually by dispatchers most of the time and for a small horizon of 7 to 10 days [200] due to the complexity of the problem caused by several constraints e.g. the number of components considered in the rolling unit, the maintenance sites, and the depot sites. However, when analyzing the previous works, we have noticed that most authors do not mention the duration of the decision horizon they considered while solving the problem. Except for Lai *et al.* [101], the authors solved the problem for a single value of decision horizon duration. Even Lai *et al.*, who solved the problem using several horizons did not study the influence of the horizon duration on the decision method performance.

#### 4.1.1.3/ Motivation

Rolling units are considered among the systems that have a long life span. In addition, maintenance cost and operating cost are important for these applications. Since the rolling systems are used to transport human and/on sensitive merchandise, their failures could have catastrophic consequences and hence they are not allowed. Therefore, the application of PHM methodology to the rolling systems is very promising. Due to their moving nature, rolling units experience different environmental and operational conditions that would affect differently the degradation of the components. Hence, the prognostics

methods should take these changing conditions into account when estimating the evolution of degradation and/or the remaining useful life. To summarize, the use of prognostics information in the management of rolling stocks is a promising field of application. On one hand, the decisions could be improved by integrating the predictive information. On the other hand, the prognostics methods are highly dependent of the future varying conditions of the system. Therefore, we believe that the railways context would benefit from a decision-enhanced PHM framework in which we emphasize the inter-dependencies between the prognostics and the decision-making processes.

#### 4.1.1.4/ Contributions

In this work, we aim to reinforce the problem stated by Herr *et al.* to make it more realistic and to fill the pointed out gaps. Therefore, We integrate prognostic information in the process of decision-making to schedule jointly maintenance and operation for rolling stock systems. The considered problem studies rolling units as multi-components with dynamic degradation models. The first main contribution of this chapter is the instantiation of the proposed framework on the rolling stocks application. The implementation of the framework would provide a method to solve the joint problem of mission assignment and maintenance planning while enhancing the use of the prognostics information and emphasizing the prognostics-decision inter-dependencies.

All the previously mentioned works focused on solving the joint problem regardless of the decision horizon duration. These works omitted the decision horizon duration in the definition of their problem and the used resolution methods. In PHM context, the prognostics algorithm accuracy are supposed to be in function of the prognostics horizon (i.e., how far ahead predictions are made) [83]. Due to uncertainty sources [67], the larger the prognostic horizon is the greater the uncertainties are present in the prediction phase. Thus, the resulted RUL has a higher error rate and is more inaccurate. Since prognostics is performed to provide new important information to the decision-making process to better manage the system life cycle, we wonder if the decrease of the prediction's accuracy over time, implies the decrease of the decision-making process performances. Therefore, we propose to study the decision horizon influence on the resolution of the joint problem as a second main contribution. We aim to find the best decision horizon duration when using the decision-enhanced PHM framework as presented in chapter 2.

#### 4.1.2/ Scope of the study

The intended application consists of managing the operations and maintenance of a fleet  $\mathcal{M}$  of  $M$  rolling units. The rolling vehicles are considered as parallel systems and could be used independently of each other. These systems are assumed to be composed of several predictive and preventive components. The idea is to assign a set  $\mathcal{P}$  of daily missions (i.e. trips) to these units and plan their maintenance intervention when needed. To mark out the scope of this study, we defined a set of hypothesis and constraints as described in the following sections.

#### 4.1.2.1/ Hypothesis

The following hypothesis are used to define the scope of the study:

- H1 All the components, that form a rolling unit, are considered as critical. Meaning that if any of the components fails the correspondent rolling vehicle fails. In other words, any rolling unit is composed of multiple serial components.
- H2 Dependencies between the components of a rolling unit are limited to structural dependencies. The stochastic and economic dependencies are not considered in this application.
- H3 No redundancies in the component level is considered. Even if some components are of the same type, the rolling unit need all these component to be functional.
- H4 Not all rolling units are necessary used to perform the set of missions.
- H5 Only replacements are considered in the maintenance actions. The quality of a replacement is assumed to be perfect. Meaning, when a component is maintained (i.e. replaced) the degradation level is reset to zero.
- H6 The degradation or the health indicators of the predictive components is supposed to be observable and could be modeled through a physic-based, data-based, or hybrid-based models.
- H7 The degradation of the components is influenced by the operating conditions of the task to be achieved.
- H8 The degradation of the components evolve only when the corresponding unit is performing a mission.
- H9 Prognostics algorithms could be developed and implemented to estimate the evolution of the degradation or the health indicators of predictive components given the future tasks to be performed.

#### 4.1.2.2/ Constraints

The following constraints are necessary to ensure the coherence of the application:

- C1 At instant  $t$ , a rolling unit can have one of three status: (i) at mission if a task is assigned to it, (ii) at maintenance if it is scheduled to maintenance, or (iii) at rest if it is not being used nor scheduled for maintenance.
- C2 At any instant  $t$ , only one rolling unit can achieve a mission.
- C3 At any instant  $t$ , a mission is assigned to at most one rolling unit.
- C4 The assignment of a mission to a rolling unit is not possible if any of the unit's components would have its degradation exceeds the failure threshold.
- C5 The maintenance resources are limited in a way that the maintenance workshop could only contain a limited number of trains and the maintenance workforce could only maintain a limited number of components per time period (e.g. per day).

### 4.1.3/ System description

#### 4.1.3.1/ Rolling stock fleet structure

Each rolling stock unit is considered as a serial multi-component system i.e. if one of these components fails the whole system fails. In PHM context, we distinguish two types of components; (i) preventive components that are difficult to monitor, their degradation cannot be modeled or observed, or it would be more strategic to perform systematic cyclic maintenance; and (ii) predictive components that are equipped with sensors, they present big volume of historical data, and their degradation behavior is at least partially observable. For these components, PHM technology is applied. Therefore each rolling stock unit  $m$  is defined as a series of  $K$  predictive components ( $c_k, k = 1, \dots, K$ ) and  $L$  preventive components ( $c_l, l = K + 1, \dots, K + L$ ).

For preventive components, the maintenance dates are defined for a certain value of mileage coverage. If the component exceeds the specified number of miles it could serve, the component is assumed to fail. The end of life of a preventive is then set to the maximum number of mile it could serve. To ensure passenger safety and comfort and avoid failures, we consider that each of the preventive components has a mileage threshold noted  $\Theta_l$  that determines the end of life of the component and above which the component is supposed to fail. Consider, at instant  $t$ ,  $\theta_{m,l}(t)$  being the mileage traveled by component  $l$  of unit  $m$  since its last maintenance operation. Preventive component  $l$  is scheduled for maintenance only when its mileage reaches a predefined threshold  $\Phi_l$  (Eq. 4.1).

$$\theta_{m,l}(i) \geq \Phi_l \quad (4.1)$$

In this application, we consider that the rolling stock units are similar in the type and number of components. However, They are differentiated by their components degradation level estimated by a condition assessment process. A prognostics process assesses the health state and estimates the remaining useful life of predictive components. For example, we consider the heating, ventilation, and air-conditioning (HVAC) subsystem as a component. Moreover, for parts of large numbers (e.g., wheels or doors), they are grouped into components according to some criterion (e.g., by wagon or by side).

#### 4.1.3.2/ Degradation model

The health state of a predictive component  $k$  of unit  $m$  is described through a variable  $H_{m,k} \in [0, 1]$  for ( $1 \leq k \leq K$ ). The degradation of any predictive component is considered to be monotonically increasing over time as a result of an accumulation of small positive independent increments. Moreover, such a degradation process has been widely modeled by stochastic processes in literature. Readers could refer to the survey paper by Van Noortwijk [174]. For simulation purposes, we consider that the degradation of the predictive components could be modeled by a homogeneous Gamma process. The health state of a predictive component  $k$   $\{H_{m,k}(t), t \geq 0\}$  is represented by the homogeneous Gamma process  $\Gamma(\nu_k(t), \mu_k)$  with shape parameter  $\nu_k(t) = \alpha_k * t$  and scale parameter  $\mu_k$  and the following properties:

- $H_{m,k}(t' = 0) = 0$
- $H_{m,k}(t)$  has independent increments

- For  $t > 0$  and a small non-negative time increment  $h > 0$  during which unit  $m$  is serving a mission,  $H_{m,k}(t + h) - H_{m,k}(t)$  follows a gamma distribution  $\Gamma(\nu_k(t + h) - \nu_k(t), \mu_k)$  with shape parameter  $(\nu_k(t + h) - \nu_k(t))$  and scale parameter  $\mu_k$

Components of the same type are considered identical in their degradation dynamics. Thus, they have the same parameter of the gamma distribution  $(\alpha_k, \nu_k)$ . However, they are differentiated by their degradation level obtained from the prognostics.

Component  $k$  of unit  $m$  is considered as good as new when its health state  $H_{m,k} = 0$  and  $H_{m,k} = 1$  indicates that it has reached its end of life and failed. To ensure passenger safety and comfort and avoid failures, components' end of life are considered upon a certain threshold denoted  $\Delta_{m,k} = \Delta_k \in [0, 1[$ . To avoid very early maintenance interventions, we suppose that each type of predictive component could only be considered for maintenance after it exceeds a minimal level of degradation denoted  $\Lambda_{m,k} = \Lambda_k \in [0, \Delta_k[$  (Eq. 4.2). The relation between the variable  $H_{m,k}$ , the thresholds  $\Delta_k$  and  $\Lambda_k$  and the launch of a maintenance operation is illustrated in Fig. 4.1. We, also, assume that the health indicator level could be converted to a number of served mileage.

$$H_{m,k}(i) \geq \Lambda_k \quad (4.2)$$

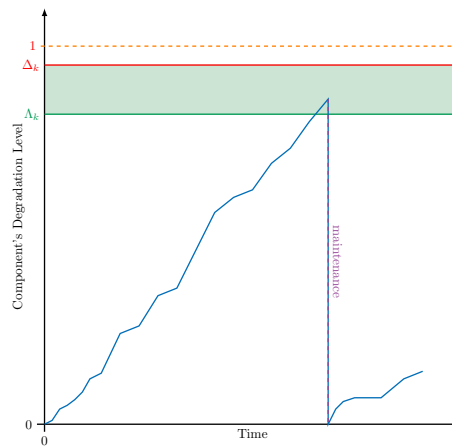


Figure 4.1: Health State Illustration.

#### 4.1.4/ Problem description

We start by defining the problem of jointly assigning missions to railways vehicles and the planning of their maintenance. The vehicle scheduling problem focuses on assigning timetabled tasks to the fleet of available vehicles. However, other constraints need to be considered while creating a vehicle schedule (e.g., vehicle type and maintenance inspection). Before assigning a mission, the prognostics algorithm assesses the rail vehicle's ability to achieve the task. The decision-making algorithm considers this information while solving the vehicle scheduling problem. Predictive maintenance is scheduled to find a compromise between early maintaining the rolling equipment, risking its failure, and missing tasks due to fleet unavailability. The notations used to define this problem are presented in table 4.1.

Notation	Signification
$DH$	Duration of Decision Horizon
$\Delta T$	duration of a period
$I$	Number of Periods in a Decision Horizon
$i$	Index of period
$\mathcal{M}$	Set of rolling units
$M$	Number of rolling units
$m$	Index of rolling unit
$\mathcal{P}$	Set of planned tasks
$P$	Number of planned tasks
$p$	Index of planned task
$d_p$	Computed distance of mission $p$
$s_p$	Severity of mission/task $p$
$K$	Number of predictive components
$k$	Index of predictive component
$\mu'_k$	Scale parameter of component's $k$ degradation model
$\alpha_k$	Shape parameter of component's $k$ degradation model
$H_{m,k}(i)$	Degradation level of component $k$ at instant $t$
$\delta_{p,k}$	Estimated degradation rate of component $k$ during mission $p$
$L$	Number of preventive component
$l$	Index of preventive component
$Sch_i$	Schedule of period $i$
$\beta_{p,m}(i)$	Indicator of assigning mission $p$ to train $m$ during period $i$
$\omega_m(i)$	Indicator if train $m$ undergoes maintenance during period $i$
$\pi_m(i)$	Indicator if train $m$ is at rest during period $i$
$f_m(i)$	Indicator if train $m$ fails during period $i$
$C_i$	Cost of period $i$
$C_{lost}$	Cost of missing a mission
$CR_k$	Replacement cost of predictive component $k$
$CR_l$	Replacement cost of preventive component $l$
$LP_x$	Penalty cost on the lost mileage of component $x$
$C_{m,x}(i)$	Cost of maintenance of component $x$ during period $i$
$C_{cor}$	Cost of corrective maintenance
$Total\_Cost(DH)$	Total cost during a decision horizon

#### 4.1.4.1/ Vehicle scheduling problem

The vehicle scheduling problem addresses the task of assigning vehicles to cover the mission in a timetable. Therefore, given a set  $\mathcal{P}$  of planned tasks (with  $Card(\mathcal{P}) = P$ ) and a set  $\mathcal{M}$  of rolling stock units (with  $Card(\mathcal{M}) = M$ ) in which  $P \leq M$ , the problem consists of finding an assignment between the  $M$  units and the  $P$  missions based on equipment's' health state and their ability to fulfill missions. The scheduling is supposed to be done over a rolling horizon of a duration  $DH = I \times \Delta T$  where  $\Delta T$  is the time unit (a day for instance) and in this case  $I$  is the number of time units. For each time unit, denoted  $i$  with  $1 \leq i \leq I$  the set of missions should be fulfilled by the railway vehicles. Each mission  $p$  corresponds to a planned route which is composed of a set of trips starting and ending

in a depot. Each mission is characterized by a severity coefficient  $s_p$  which depends on the length and conditions of the trip (e.g., number of miles the rolling stock unit should travel in that mission, or if there is hill climbing), the number of stops, and the chosen path (e.g., if it is a high-speed track or normal track). Thus, it influences the degradation of the components by changing the scale parameter of the gamma process which can be expressed as  $\mu_k = s_p \times \mu'_k$  with  $\mu'_k$  a characteristic of the component's type.

A task  $p$  is associated with a predicted degradation rate  $\delta_{p,k} \in [0, 1[$  that can vary from one component to another. This wear rate is provided by a prognostic process. For each component  $k$  of a given rolling stock unit  $m$  at instant  $t = i \times \Delta T$  with a health state  $H_{m,k}(i)$ , its predicted degradation level after executing its assigned mission  $p$  must be lower than its failure threshold  $\Delta_k$ . Eq. 4.3, illustrate this constraint.

$$H_{m,k}(i) + \delta_{p,k} < \Delta_k, \forall k \in \{1, \dots, K\}, \forall i \in \{1, \dots, I\} \quad (4.3)$$

Also, each task  $p$  is characterized by a distance  $d_p$  to be covered by the rolling vehicle  $m$  at which the mission will be assigned to. Therefore, the assignment of mission  $p$  to rolling stock unit  $m$  should also verify that the covered distance will not exceed the threshold for preventive maintenance. This constraint is represented in Eq. 4.4. It is assumed that if the mileage of a preventive component  $l$  exceeds its threshold, the component fails, thus causing the failure of the vehicle.

$$\theta_{m,l}(i) + d_p < \Theta_l, \forall l \in \{K + 1, \dots, K + L\}, \forall i \in \{1, \dots, I\} \quad (4.4)$$

Lets denote the variable  $\beta_{p,m}(i) \in \{0, 1\}$ . Where  $\beta_{p,m} = 1$  if mission  $p$  is assigned to unit  $m$ . Otherwise, it is equal to zero. These variables are subject to the following constraints.

$$\sum_{m=1}^M \beta_{p,m}(i) \leq 1, \forall i \in \{1, \dots, I\}, \forall p \in \{1, \dots, P\} \quad (4.5)$$

$$\sum_{p=1}^P \beta_{p,m}(i) \leq 1, \forall i \in \{1, \dots, I\}, \forall m \in \{1, \dots, M\} \quad (4.6)$$

Constraint 4.5 represents the mission coverage constraint, where each task  $p$  is covered by at most one railway vehicle during any period  $i$  of the decision horizon. Eq. 4.6 says that one rolling stock unit  $m$  can fulfill at most one task during any period  $i$  of the decision horizon. Moreover, unit  $m$ , during any period  $i$ , can either be in maintenance, at rest (i.e., no mission is assigned to it) or assigned to a task  $p$ .

$\omega_m(i) \in \{0, 1\}$  represents the maintenance state of unit  $m$  during period  $i$ . If unit  $m$  is in maintenance then  $\omega_m(i) = 1$ . Otherwise,  $\omega_m(i) = 0$ .

$\pi_m(i) \in \{0, 1\}$  capture if rail vehicle  $m$  is at rest during period  $i$  (i.e.,  $\pi_m(i) = 1$  if  $m$  is neither in maintenance nor in operation. Therefore, the state of rolling stock unit  $m$  during a period  $i$  can be limited with constraint 4.7.

$$\pi_m(i) + \omega_m(i) + \sum_{p=1}^P \beta_{p,m}(i) = 1, \forall i \in \{1, \dots, I\}, \quad (4.7)$$

$$\forall m \in \{1, \dots, M\}$$

Lets denote  $C_{lost,p}$  the penalty cost of missing mission  $p$  during a period. All missions have the same priority. Thus, they all have the same missing penalty  $C_{lost,p} = C_{lost} \forall p \in \{1, \dots, P\}$ .



Moreover, we assume that the mission assignment costs the same no matter the rail vehicle or the mission assigned to it. Therefore, the cost of mission assignment is excluded from this study.

#### 4.1.4.2/ Maintenance problem

Lets note  $\sigma_{m,k}(i) \in \{0, 1\}$  and  $\sigma_{m,l}(i) \in \{0, 1\}$  with  $k \in \{1, \dots, K\}$  and  $l \in \{K+1, \dots, K+L\}$  the variables that determine if component  $k$  respectively  $l$  of rolling stock unit  $m$  are scheduled for maintenance during period  $i$ . If  $\sigma_{x,m}(i) = 1 \forall x \in \{1, \dots, K, K+1, \dots, K+L\}$  means that component  $x$ , whether it is predictive or preventive, is scheduled for maintenance. Otherwise, the component is not maintained during period  $i$ . Each component's type is characterized with a maintenance cost. Thus, lets note  $CR_k$  and  $CR_l$  the replacement cost of predictive component  $k$  and preventive component  $l$  respectively. Early maintenance of any component generates an additional cost to the replacement one. This is the penalty on the lost mileage of the maintained component noted  $LP_x \forall x \in \{1, \dots, K, K+1, \dots, K+L\}$ . For the predictive components, the penalty coefficient take into consideration the conversion from health indicator to mileage. Therefore, at period  $i$ , the maintenance cost of predictive component  $k$  of unit  $m$ , noted  $C_{m,k}(i) \forall k \in \{1, \dots, K\}$ , and preventive component  $l$  of unit  $m$ , noted  $C_{m,l}(i) \forall l \in \{K+1, \dots, K+L\}$  are defined respectively in Eqs. 4.8 and 4.9.

$$C_{m,k}(i) = CR_k + (\Delta_k - H_{m,k}(i)) * LP_k, \forall i \in \{1, \dots, I\}, \quad (4.8)$$

$$\forall m \in \{1, \dots, M\}$$

$$C_{m,l}(i) = CR_l + (\Theta_l - \theta_{m,l}(i)) * LP_l, \forall i \in \{1, \dots, I\}, \quad (4.9)$$

$$\forall m \in \{1, \dots, M\}$$

Lets note  $f_m(i) \in \{0, 1\}$  the variable that describes if unit  $m$  fails during period  $i$ . If during a task one of the components of unit  $m$  (predictive or preventive) fails and causes the system to fail (i.e.,  $f_m(i) = 1$ ), a corrective maintenance cost noted  $C_{cor}$  is generated. Vehicle's failure causes a disturbance of the timetable schedule, a delay for all scheduled missions on the same line, and efforts to move the failed rolling unit to a spare track. Thus, corrective maintenance penalty should take into consideration all these activities and their cost. Therefore, the corrective maintenance cost is defined in a way that missing some missions is less costly than risking the asset failure (Eq. 4.10).

$$C_{cor} \gg C_{lost} \quad (4.10)$$

Maintenance resources are considered limited, i.e., the number of maintenance operators and their shifts are limited. This limitation is presented through the maximum number of components  $ML_{Comp} > 0$  that can be maintained during period  $i$ . Moreover, the maintenance workshop is supposed to have a limited number of tracks. Therefore, only a certain number of railway vehicles noted  $ML_{Equip} > 0$  can be maintained simultaneously during period  $i$ . The maintenance planning process should take into consideration and satisfy these limits. Therefore, lets define the constraints that represent the limit in the number of rolling stock units Eq. 4.11 and the limit in the number of components Eq. 4.12.

$$\sum_{m=1}^M \omega_m(i) \leq ML_{Equip}, \forall i \in \{1, \dots, I\} \quad (4.11)$$



$$\sum_{m=1}^M \left( \sum_{k=1}^K \sigma_{m,k}(i) + \sum_{l=K+1}^{K+L} \sigma_{m,l}(i) \right) \leq ML_{Comp}, \quad \forall i \in \{1, \dots, I\} \quad (4.12)$$

#### 4.1.4.3/ Joint Problem

Classically, the problem of joint maintenance and operation optimization consists of minimizing the global cost by adding maintenance cost and missing task cost over the duration of the decision horizon noted  $DH$ . This decision horizon is divided into  $I$  time periods (e.g., days, hours, or weeks). The total cost over a decision horizon  $DH$  is defined as in Eq. 4.13.

$$\begin{aligned} Total\_Cost(DH) = & \sum_{i=1}^I \left[ \sum_{p=1}^P C_{lost} * \left( 1 - \sum_{m=1}^M \beta_{p,m}(i) \right) \right. \\ & + \sum_{m=1}^M (f_m(i) * C_{cor}) \\ & + \sum_{k=1}^K \sigma_{m,k}(i) * C_{m,k}(i) \\ & \left. + \sum_{l=K+1}^{K+L} \sigma_{m,l}(i) * C_{m,l}(i) \right] \end{aligned} \quad (4.13)$$

The first term in the total cost function is the cost of all missed missions over the period. The second term represents the maintenance cost including corrective maintenance penalty in case of failure, and predictive and preventive components if they are maintained. Thus, the objective function of the joint problem can be written as in Eq. 4.14.

$$\min Total\_Cost(DH) \quad (4.14)$$

Solution of this joint problem is valid if it proposes a schedule of operation and maintenance activities that satisfies all constraints defined in Eqs. 4.5, 4.6, 4.7, 4.11, and 4.12. Moreover, a solution is considered feasible if during all the schedule none of the rolling stock units fails during operations. In other words, none of the components exceeds its failure threshold. Therefore, for a solution to be feasible all the components should verify Eqs. 4.3 and 4.4.

#### 4.1.5/ Decision Horizon Study

In reality, the joint problem is going to be optimized several times during the operations of the assets. And what railways companies are looking for is the optimization of the overall cost in the long term. Apart from applying the proposed framework on a rolling stock case study, a second main contribution of this chapter is to find the decision horizon duration  $DH$  that optimizes the resolution of the joint problem over a simulation horizon  $SH$ . In other words, we are looking for the suitable  $DH$  value that minimizes the cumulative total cost over the simulation horizon  $SH$  obtained by repeating the optimization of the joint problem for  $I$  periods. This simulation horizon is covered by  $N$  steps of decision-making (i.e. joint problem resolution) over the rolling horizon  $DH$ . The number of steps, noted  $N$ ,

is defined in a way that verifies  $SH = N * I * \Delta T$ . Where  $\Delta T$  is the duration of a time period (e.g. duration of one day).

Therefore, the objective function of the decision horizon study problem is defined in Eq. 4.15.

$$\min \sum_N Total\_Cost(DH) \quad (4.15)$$

With  $\sum_N Total\_Cost(DH)$  is defined as the cumulative total cost over the simulation horizon  $SH$ .

To summarize, this concept is presented in Fig. 4.2. The simulation horizon  $SH$  is divided into  $N$  equal parts of duration we call decision horizon  $DH$ . For each  $DH$ , we solve the joint problem to obtain the total cost. The sum of these total costs produces the cumulative total cost. The aim of the chapter is to study the influence of the decision horizon on this cumulative total cost and minimize it by finding the suitable decision horizon.

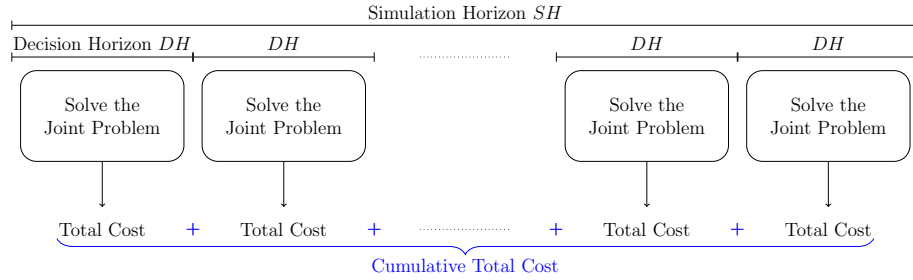


Figure 4.2: Illustration of the Cumulative Total Cost.

## 4.2/ Instantiation of the proposed framework

The presented framework in chapter 2 is instantiated to the problem described in the previous section. Hence, we define the different components of the framework and identify its mechanisms. We start by identifying the values of the descriptors of the system and the application. Then, we define the elementary actions, local decisions, and the structure of the global decision. Finally, we set up the needed loops for this application.

### 4.2.1/ System definition

The proposed framework requires a definition of the system as a list of attributes-values. Most of the attributes values are derived from the optimization problem statement. Some of the attributes are given empirically just to satisfy the needs for the example. We define the system's descriptors as follow:

- **System's type:** railways systems
- **System's typology:** a fleet of  $M$  rolling vehicles i.e. multiple machine in a parallel organization.

- **Granularity:** In this example, we consider a single granularity level of the system. In other words, we compose the system into components that has the same hierarchy level. Therefore, the system is considered as multi-components. Each rolling unit is composed of  $K$  predictive components and  $L$  preventive components.
- **Dependencies:** The  $K + L$  components of this machine are assumed to be only structurally dependent. The structure dependency is considered to be serial. Thus, we can see the machine a sequence of  $K$  predictive and  $L$  preventive components. The dependency is defined in a way that if one component fails the whole system fails.
- **Criticality:** This attribute is filled empirically. We consider the fleet of rolling vehicles as a system with a high level of criticality.

In the example, we considered a fleet of rolling stock units in which each unit is composed of  $K + L$  serial components. We assumed that each component is essential to the operations of the train i.e. if one component fails the machine fails. Since no redundancies are considered for the components, we can assume that the components has a high critical level. With no redundant components or machines, if one component fails, the whole system fails. Moreover, the failure of a rolling unit (e.g. train) can imply a major user safety threat. The rolling units are supposed to transport travelers during their missions and some of the components' failures could cause only the rolling unit to stop on the tracks while other components' failures could cause catastrophic events like train derailment. The catastrophic failures are avoided by setting a safety margin between the assumed components failure and its actual failure through defining thresholds. Even though, we try to eliminate failures, they can still be authorized against the safety thresholds. Due to these reasons, we consider the fleet of rolling vehicles as a system with a high level of criticality.

- **Observability:** The degradation level of the components is supposed to be observable. This means that we can directly measure the degradation level accurately or the degradation level could be deduced from measurable signals like vibrations, motor electrical measures, temperature values, etc.
- **Degradation parameters:** From the definition of the degradation model in the problem statement we can set the attributes for the degradation parameters. Such attributes should be specific for each component of the machine. In this example, since we did not specify which components are considered and to not lose the generic aspect of the application, we define these attributes in a general manner.
  - *Degradation form:* Some considered components are supposed to be subject of wear and fatigue e.g. bogies. Some other components are supposed to be subject to wear and crack growth like for example the wheels or the pantograph. The form of degradation of each component of the system should be specified.
  - *Degradation causes:* Degradation evolution is supposed to be related to the operations of the machine. When the train is idle the degradation level of all its components is supposed to be constant. Therefore, the degradation evolution is operationally caused. The characteristics of this evolution is assumed to be influenced by operational conditions related to the mission the train is achieving. These operational conditions include the estimated load of the train (number of passengers), the number of stops during the trip, the length of the trip, etc. For

simplicity sake, we supposed that these operational conditions are summarized in a severity coefficient of the mission.

- *Degradation model:* In this problem, we considered that the predictive components' degradation level is stochastic and can be modeled using gamma process. The parameters of the gamma process (i.e. the shape and scale parameters) are estimated for each type of components based on the collected data. Hence, the degradation model is supposed to be data-based. However, since we do not actually have real data for the system, the parameters of the components are given empirically. The evolution of the component's degradation is assumed to be influenced by the severity of the missions the train is achieving.
- *Degradation consequences:* The railways vehicles are supposed to transport passengers. Hence, the system is considered to be highly critical. The components' failures differs according to the component's type. Some components e.g. the pantograph, are considered as low risk since their failure can only cause the traffic to stop and would cause a high penalty cost. Other components e.g. bogies or wheels are critical not only to the operations of the unit but also to the safety of the passengers. However, for these components we fixed a safety threshold to avoid the catastrophic effects of their failure and thus we assumed their failure (against the threshold) to only cause traffic problems and hence the high penalty cost.
- **Overall degradation dynamics:** Since the components are structurally dependent, then the machine overall degradation follows the component with the highest degradation level.
- **Action variables:** In this application the degradation evolution is influenced by the operational conditions of the missions scheduled. This influence of the operation conditions is summarized in the mission's severity coefficient. Therefore, the choice of the mission according to their severity coefficient or putting the considered train to rest are the only possible action variables.
- **Complexity level:** The definition of the complexity level of the system is a difficult task and no consensus method is yet to be proposed. Therefore, we are giving an empirical value for this attribute. We consider that the railways system is more complex than the CNC machine considered in the third chapter. We assume that the studied system has a medium complexity level. This is justified by the following reasons:
  - The system is composed of a multiple machines
  - Each machine is composed of a single level of granularity composed of several components.
  - components' dependencies are only structural and present a single type of structural dependencies in which all the components are considered serial.
  - The degradation level is supposed to be based on a gamma process statistical model and presenting uncertainties.

#### 4.2.2/ Application definition

Therefore, the application can be summarized as:

- **System:** As we discussed in the second chapter the application definition should include the detailed definition of the system. Therefore, for this attribute, the definition of the system done in the previous section is kept.
- **Objective:** this is a cost oriented application in which we are trying to minimize the total cost (i.e. operational and maintenance cost) over a decision horizon.
- **Performance indicators:** To follow and asses the efficiency of the methods we propose several indicators that are monitored in the decision-making process i.e. the optimization process. The obtained solution are compared and assessed upon these indicators that are computed over the simulation horizon:
  - *Total Cost:* This is the estimated total cost including the cost of operations and maintenance actions. This field is computed over the duration of the decision horizon as defined in Eq. 4.13.
  - *Total number of maintenance activity per unit per component:* This field contains the number of maintenance activities of each component during the decision horizon.
  - *Total number of corrective maintenance activities per component:* This field notes the number of corrective maintenance performed on each component during the decision horizon.
  - *Total number of corrective maintenance activities:* This is the number of corrective maintenance activities during the decision horizon. It reflects the number of train failures over the horizon.
  - *The average lost degradation level per component:* When performing predictive maintenance, there is a difference between the threshold of failure and the actual level of degradation of the component in question. This difference is used in Eqs. 4.8 and 4.9 to compute the lost RUL penalty. At each maintenance activity, this difference is computed and saved. Then, the average difference for each component per maintenance activities is computed. This allows to determine how well the maintenance activities are scheduled. A big value of this indicator shows a relative premature maintenance scheduling. A null value indicate that the schedule fails to anticipate the failure. The aim of PHM is to minimize this difference while still being able to avoid failure and then corrective maintenance.
  - *The proportion of done missions:* This represents the ration of the done missions over the decision horizon.
  - *The proportion of missed missions:* This represents the ration of missed missions over the decision horizon.
- **Decisions nature:** we are studying a joint problem between production and maintenance. Thus, the decisions in this application are mixed. The two nature of the actions can be defined as:
  - Action of maintenance type: selecting a set of components to be maintained
  - Action of operational type I: assigning a mission  $p$  to unit  $m$
  - Action of operational type II: putting unit  $m$  at rest

- **Simulation horizon:** In real life, there is no simulation horizon since the application is supposed to be running infinity. However, since this application is a proof of concept we defined a simulation horizon upon which we validate the approach and assess its performance over a long enough run.
- **Decision horizon:** We consider a rolling decision horizon. Such a horizon consists of a time window during which the planned tasks  $\mathcal{P}$  are assigned to the  $M$  trains and maintenance of trains are planned, then this time window is shifted and another decision is made. Details about rolling decision horizon can be found in the work of Dekker *et al.* [41].
- **Discretization Period:** The decision horizon is divided into smaller periods of time we call discretization period. In this application the discretization period has a one day duration.

### 4.2.3/ Elementary Actions

The studied application consists of a joint problem of maintenance scheduling and mission assignment. Therefore, two types of elementary actions exists: (i) operational elementary action and (ii) maintenance elementary action.

Lets start by defining the operational elementary action. The operational elementary actions could be of two nature: (i) assign a mission  $p$  to train  $m$  or (ii) placing train  $m$  at rest. In chapter 2, we have stated that elementary actions are transformed into local decision when they are scheduled i.e. the elementary action is specific period. Therefore, the elementary actions do not present the period in question.

We propose to detail the attributes used for defining the operational elementary action. In table 4.2, we present an example of a typical mission assignment action. While table 4.3 contains an example of a rest action by putting train  $m$  at rest.

Table 4.2: Definition of a mission assignment elementary action

Descriptor	Value
Type of action	Mission Assignment
Concerned Rolling Unit	$m$
Mission number	$p$
Mission's length	$d_p$
Mission's severity	$s_p$

Table 4.3: Definition of a put-to-rest elementary action

Descriptor	Value
Type of action	Put-to-Rest
Concerned Rolling Unit	$m$

The maintenance elementary action consists of choosing which component(s) of which vehicle to maintain. The definition of a maintenance elementary action is given in table 4.4. In these actions,  $\mathcal{PM}$  presents the set of the components to be maintained, it contains the identification of predictive  $k$  and preventive  $l$  components.

Table 4.4: Definition of a maintenance elementary action

Descriptor	Value
Type of action	Maintenance
Nature of action	Replacement
Concerned vehicle	$m$
Concerned component(s)	$\mathcal{PM} = \{k, l\}$

#### 4.2.4/ Local Decisions

At instant  $i$ , we propose to find all the possible local decisions. We consider the set of the elementary action and we estimate the outcome from scheduling these actions at this instant  $i$ . We have two types of elementary actions and consequently, we have two types of local decisions.

Let's start by defining the attributes for a typical maintenance local decision. The defined local decision is presented in table 4.5.

Table 4.5: Structure of a maintenance local decision

Descriptor	Value
Type of action	Maintenance
Nature of action	Replacement
Concerned vehicle	$m$
Concerned component(s)	$\mathcal{PM} = \{k, l\}$
Period	$i$
Initial Health State	$(H_{m,1}, \dots, H_{m,k}, H_{m,K})$
Initial Preventive State	$(\theta_{m,K+1}, \dots, \theta_{m,l}, \dots, \theta_{m,K+L})$
Estimated Health State	$(H_{m,1}, \dots, H_{m,k}, H_{m,K})$
Final Preventive State	$(\theta_{m,K+1}, \dots, \theta_{m,l}, \dots, \theta_{m,K+L})$
Cost	Equations 4.8 and 4.9

Considering the definition of mission assignment elementary action, the local decision derived from this action is defined in table 4.6.

Table 4.6: Structure of a mission assignment local decision

Descriptor	Value
Type of action	Mission Assignment
Concerned Rolling Unit	$m$
Mission number	$p$
Mission's length	$d_p$
Mission's severity	$s_p$
Period	$i$
Initial Health State	$(H_{m,1}, \dots, H_{m,k}, H_{m,K})$
Initial Preventive State	$(\theta_{m,K+1}, \dots, \theta_{m,l}, \dots, \theta_{m,K+L})$
Estimated Degradation Evolution	$(\delta_{p,1}, \dots, \delta_{p,k}, \dots, \delta_{p,K})$
Estimated Health State	$(H_{m,1} + \delta_{p,1}, \dots, H_{m,k} + \delta_{p,k}, H_{m,K} + \delta_{p,K})$
Final Preventive State	$(\theta_{m,K+1} + d_p, \dots, \theta_{m,l} + d_p, \dots, \theta_{m,K+L} + d_p)$

In table 4.7, we present the local decision derived from putting a train at rest.

Table 4.7: Definition of a put-to-rest local decision

Descriptor	Value
Type of action	Put-to-Rest
Concerned Rolling Unit	$m$
Period	$i$
Initial Health State	$(H_{m,1}, \dots, H_{m,k}, \dots, H_{m,K})$
Initial Preventive State	$(\theta_{m,K+1}, \dots, \theta_{m,l}, \dots, \theta_{m,K+L})$
Estimated Health State	$(H_{m,1}, \dots, H_{m,k}, \dots, H_{m,K})$
Final Preventive State	$(\theta_{m,K+1}, \dots, \theta_{m,l}, \dots, \theta_{m,K+L})$

#### 4.2.5/ Global Decisions

A global decision, in this application, is defined as the schedule of assigned missions and maintenance intervention during a decision horizon. As described in the previous chapter, the global decision can also be defined as a set of (attributes, values). The global decision is constructed by sequencing the local decisions. In this study, the considered system is formed of several parallel machines. For each rolling unit  $m$ , a sequence of local decision is constructed in a way that for each period  $i$  of the decision horizon, the unit  $m$  is either assigned to a mission, at rest, or in maintenance. Therefore, the structure of the global decision's schedule is constructed of  $M$  parallel sequences of local decision (one sequence local decision per rolling unit). An example of a global decision's schedule is given in Fig. 4.3. In this example, we considered a system composed of 5 trains to which we assign 4 daily missions (i.e. the period is equal to a day). We considered a decision horizon of 4 periods just to illustrate the structure of the global decision's schedule. The green boxes represent the operational local decision of mission assignment. The orange boxes represent the operational local decision of putting the train to rest. The red box represents the local decision of maintaining predictive components number 1 and 5 (i.e.  $\mathcal{PM} = \{k = 1, k = 5\}$ ). Each column represents the schedule  $Sch_i$  of a period  $i$ . While a row represents the schedule of a train over the decision horizon. The whole matrix represents the schedule of the global decision noted  $Sch$ .

		Time Periods			
		$i = 1$	$i = 2$	$i = 3$	$i = 4$
Railway Equipments	$m = 1$	Mission $p = 3$	Put - to - Rest	Mission $p = 12$	Mission $p = 13$
	$m = 2$	Mission $p = 2$	Mission $p = 5$	Put - to - Rest	Mission $p = 16$
	$m = 3$	Put - to - Rest	Mission $p = 8$	Mission $p = 9$	Mission $p = 14$
	$m = 4$	Mission $p = 1$	Mission $p = 6$	Mission $p = 11$	Maintenance $\{k = 1, k = 5\}$
	$m = 5$	Mission $p = 4$	Mission $p = 7$	Mission $p = 10$	Mission $p = 15$

Figure 4.3: An Example of Global Decision Schedule Structure.

To carry on with the implementation of the proposed framework, we define in table 4.8 the structure of the global decision used through this application. For the descriptors, we do present the used formula to compute the values instead of an example of values. The initial and final health (respectively preventive) states of each unit are also included in the



structure of the global decision. In the table, we present the generic presentation of these values for unit  $m$ .

Table 4.8: Structure of a Global Decision

Descriptor	Value
Schedule	$Sch$
Decision Horizon	$DH$
Total Cost	$Total\_Cost(DH)$ (Eq. 4.13)
Number of Maintenance per Predictive Component	$\sum_{i=1}^I \sigma_{m,k}(i) \forall 1 \leq k \leq K \forall m \in \mathcal{M}$
Number of Maintenance per Preventive Component	$\sum_{i=1}^I \sigma_{m,l}(i) \forall K+1 \leq k \leq K+L \forall m \in \mathcal{M}$
Total Corrective Maintenance Activities	$\sum_{i=1}^I \sum_{m=1}^M f_m(i)$
Done Missions	$\sum_{i=1}^I \sum_{p=1}^P \sum_{m=1}^M \beta_{p,m}$
Lost Missions	$\sum_{i=1}^I \sum_{p=1}^P (1 - \sum_{m=1}^M \beta_{p,m})$
Initial Health of Unit $m$	$(H_{m,1}(0), \dots, H_{m,k}(0), \dots, H_{m,K}(0))$
Final Health of Unit $m$	$(H_{m,1}(I), \dots, H_{m,k}(I), \dots, H_{m,K}(I))$
Initial Preventive State of Unit $m$	$(\theta_{m,K+1}(0), \dots, \theta_{m,l}(0), \dots, \theta_{m,K+L}(0))$
Final Preventive State of Unit $m$	$(\theta_{m,K+1}(I), \dots, \theta_{m,l}(I), \dots, \theta_{m,K+L}(I))$

#### 4.2.6/ Loops settings

According to the framework defined in the second chapter, to enhance the use of prognostics information in the decision-making process and vice-versa, three loops have been introduced:

- Decision building loop, in which we construct sequence of decisions iteratively by finding all possible local decisions, estimating their outcomes on the system's health, and selecting the suitable one. To achieve this loop, we need to implement a local prognostics algorithm in the **estimator** module and a **decision builder** module.
- The decision building loop outputs one or several global decisions. The decision-making module is responsible of selecting a global decision that will be applied on the system. This is the decision applying loop.
- As a consequence of applying the global decision, we obtain the real evolution of the components' degradation. By comparing the estimated and real evolutions, we obtain the errors of local prognostics that could be due to uncertainties sources that are not accounted for. The information loop uses the new available data with the local prognostics errors to fine tune the estimator.

In the manufacturing application, we focused on the implementation of the decision building loop. The decision applying loop was considered in consequence but not studied in details. Moreover, the information loop was not implemented because of the deterministic character of the degradation model. In this application, all the three loops of the proposed framework are implemented but with different levels. To ensure the implementation of these loops, we start by defining the added modules i.e. the estimator and the decision builder:

- Unlike the previous chapter in which the degradation model is supposed to be deterministic, in this application, we assume that the degradation evolution of the predictive components is influenced by different uncertainties. We assumed that the degradation of the predictive components are data-based and monotonically increasing over time by accumulating small independent positive increments. The choice of the estimator is not an aim of this study. Hence, we propose to use a linear regression model as an estimator. Since the predictive components of different types have different degradation dynamics, we proposed to set up an estimator (linear regression model) for each type of components. For simulation purposes, we proposed that the degradation of the predictive component could be modeled with a gamma process (as presented in paragraph 4.1.2.2). The gamma process models are used to generate the initial data and to simulate the real evolution of the components' degradation when the global decision is applied. We use the degradation data generated by the gamma process to train the estimator. Finally, during the construction process, for each predictive component we estimate the degradation rate of the mission using the correspondent linear regression model. Given the initial state of degradation and the mission to perform, the linear regression model returns the estimated evolution of degradation.
- The decision builder module is implicitly implemented in the used methods for the decision building loops. In this application, we propose several decision methods that implemented the decision builder functions in the heuristics.

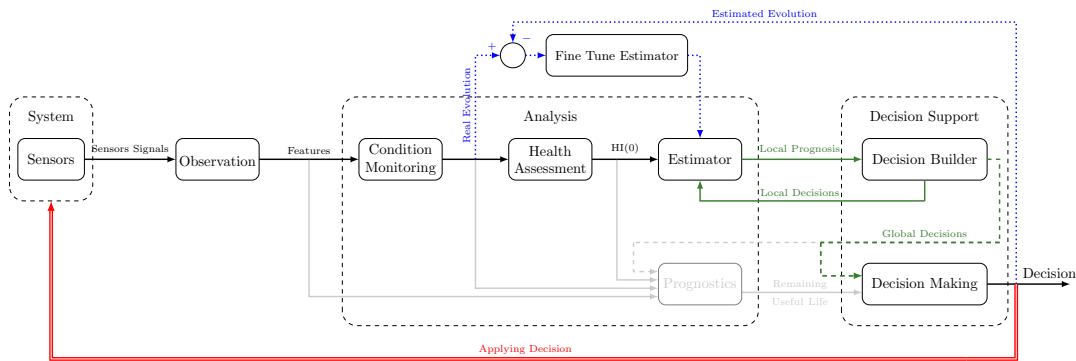


Figure 4.4: Proposed Framework Implementation.

For this application, we implemented the three loops of the proposed framework. These loops are presented in figure 4.4. However, due to the absence of real data, some components of the framework were not implemented e.g. the prognostics module. The implementation of the proposed loops is as follow:

- The decision building loop is implemented inside the used decision methods (i.e. the genetic algorithm and the heuristics). The purpose of this loop is to build global decisions that solve the joint problem of mission assignment and maintenance planning for a given decision horizon. Hence, the decision methods rely on the estimator to implement the iterative way of building the decisions. For the heuristics the process is quite direct, the heuristics assess the possible elementary actions, estimate the outcome of the assignment, and then validate it. For the genetic algorithm, the implementation of the decision building loop is quite different. We

used heuristics to validate the schedules constructed by the genetic algorithm, if the solution (i.e. schedule) is not valid or feasible, modifications are made based on the estimated outcomes provided by the estimator.

- The decision applying loop is implemented in the simulation process. Once a global decision is selected the schedule is applied to the system by simulating the degradation evolution. For each predictive component, we simulated the evolution of degradation using the proposed gamma process model. For the preventive components, we simply add the distance of the trip to their mileage state ( $\theta_{m,t}$ ). Since we assumed that the maintenance interventions are perfect, each component scheduled for maintenance is returned to an as-good-as-new status by setting its degradation level or traveled mileage to zero.
- When we apply the global decisions on the system during the simulation process (in the decision applying loop), we collect the real evolution of the component's degradation (i.e. the new degradation point generated by the gamma process models). We compute the error of the estimators. If the errors exceeds a predefined (empirically) threshold i.e. the train experienced new conditions unknown to the estimator. In this case, the collected degradation data is added to the training data of the linear regression models (i.e. estimators) and the model is fine tuned (i.e. retrained). Else, the model is supposed to be up-to-date and do not require further tuning. This process was implemented in this application. However, it was not subject to a detailed study nor to an optimization of its parameters.

At this point, we have initialized the proposed framework and set up its loop. We also chose and set up the estimator module. In the next section, we set up and implement the decision module.

### 4.3/ Implementation of the decision module

To study the influence of decision horizon, we should solve the joint problem of mission assignment and maintenance planning in rolling stock systems. This problem is a combinatorial discrete optimization problem. The use of prognostics information like degradation level and/or remaining useful life of the systems adds some non-linearity to the problem (e.g., by using a non-linear degradation model). However, we are not proposing to study and use the best resolution method for this problem. Instead, we only propose to use common optimization methods to illustrate the possibility of integrating and implementing the proposed framework. Therefore, we used a genetic algorithm and three heuristics as decision methods to solve this problem. These methods are described in this section.

#### 4.3.1/ Genetic Algorithm

Genetic algorithm (GA) is a well used mature method based on heuristic rules to produce improved approximations of the objective function over several iterations. GA search techniques are based on biological systems rules for natural survival in a different environment. The algorithm starts with a set of initial solutions called population, in which each solution (i.e., an individual of the population) is called a chromosome. Through successive generations (i.e., iterations), these chromosomes evolve as the result of crossover and

mutation operators. Each chromosome is evaluated using some measure of fitness. The new generation is created by selecting some chromosomes from the previous generation and new chromosomes resulted from the genetic operators (i.e., crossovers and mutations) [38]. Therefore, to implement GA, several components should be considered:

- the genetic representation of the solution,
- the fitness function,
- the method to generate initial population,
- the genetic operators (mutation and crossover), and
- the survival rules.

Even though GA does not guarantee the global optimum solution, it is a commonly used method in cases of combinatorial, high instances or non-linear optimization problems. The required components of the GA implementation are presented in this subsection.

#### 4.3.1.1/ Solution Representation

The coding of GA chromosomes is a key step when using such kind of approach. As detailed in the model the variables of the optimization problem are  $\beta_{p,m}(i)$  and  $\sigma_{x,m}(i)$   $\forall p \in \{1, \dots, P\}, m \in \{1, \dots, M\}$  and  $\forall x \in \{1, \dots, K + L\}$ . Then, we propose a coding consisting of a 2D matrix of integers, in which columns represent the  $M$  rolling stock units while rows represent the decision horizon  $DH$  (i.e.,  $I$  periods). This coding is a sequence of alleles, in which each allele is the schedule of a rolling stock unit  $m$  ( $1 \leq m \leq M$ ) over the decision horizon (i.e.,  $I$  periods). Element  $e_{i,m}$  of the 2D-array (i.e., element of the  $m^{th}$  column and the  $i^{th}$  row) represents the planned activity for unit  $m$  during period  $i$  ( $1 \leq i \leq I$ ). The value of any element can be negative or positive, depending on the activity it represents. Eq. 4.16 presents the possible values for an element  $e_{i,m}$  according to the scheduled activity. For scheduled missions, the value is equal to the identification number of the mission. For maintenance activities, the value of the element is a negative integer  $z_{i,m}$  that represents a coding of the identification number of the components to be maintained (Eq. 4.17). To represents the unit at rest, the value of the corresponding element is set to zero. An example of solution representation is given in Fig. 4.5.  $e_{4,4} = -0105$  which means that components number 1 and 5 of unit 4 are scheduled for maintenance during the period 4.

$$e_{i,m} = \begin{cases} p & \text{if } \beta_{p,m}(i) = 1 \\ z_{i,m} & \text{if } \omega_m(i) = 1 \\ 0 & \text{if } \pi_m(i) = 1 \end{cases} \quad (4.16)$$

$$z_{i,m} = (-1) \times \sum_{y=1}^{Card(Maint_{m,i})} (10^{a \times y} \times Maint_{m,i}[y]) \quad (4.17)$$

with  $a$  the number of digits in  $K + L$  and  $Maint_{m,i}$  the set of components' identification number that are scheduled for maintenance (Eq. 4.18).

$$Maint_{m,i} = \{x \in \{1, \dots, K + L\} \mid \sigma_{x,m}(i) = 1\} \quad (4.18)$$

		Railway Equipments				
		$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
Time Periods	$i = 1$	3	2	0	1	4
	$i = 2$	0	5	8	6	7
	$i = 3$	12	0	9	11	10
	$i = 4$	13	16	14	-0105	15

Figure 4.5: An Example of Solution Representation.

#### 4.3.1.2/ Initial Generation

GA is based on improving solutions from one generation to another over a predefined number of iterations. Therefore, this kind of algorithm requires an initial generation (or set) of valid and/or feasible solutions. To build the first generations' individuals, a heuristic algorithm (Algorithm 3) is used. The idea is to sort the set of rolling stock vehicles according to their degradation level which is defined by the degradation level of their most degraded components. Then the algorithm schedules maintenance for the most deteriorated components (that satisfies the maintenance condition Eq. 4.2). It assigns randomly the available tasks of the period  $i$  to the rest of the vehicles. The outcome of the  $i^{th}$  period is estimated, the health states of the predictive components and the mileage traveled of preventive ones are virtually updated, and the algorithm moves to schedule the next period until it reaches the end of the decision horizon. This heuristic is executed several times until the initial population is constructed.

---

**Algorithm 3:** Creation of the Initial Generation.

---

```

1 while  $i \leq DH$  do
2   Sort vehicles according to their degradation
3   while  $(\sum_m (\sum_k \sigma_{m,k} + \sum_l \sigma_{m,l}) \leq ML_{Comp})$  AND  $(\sum_m \omega_m(i) \leq ML_{Equip})$  AND
       $((\exists(m, k) | H_{m,k} \geq \Lambda_k) \text{ OR } (\exists(m, l) | \theta_{m,l} \geq \Phi_l))$  do
4     Schedule maintenance for most deteriorated component
5   Assign randomly tasks to the rest of vehicles
6   Estimate the outcome of the schedule
7   Update the health states and the traveled mileage
8    $i++$ 

```

---

#### 4.3.1.3/ Genetic Operators Results

The genetic operators do not take into consideration the health states of the rolling units. Therefore, the outcomes of these operators could result in the violation of the feasibility constraints defined in Eqs. 4.3 and 4.4. Therefore, the obtained offsprings from genetic operators undergo a check and reparation phase. During which, the outcomes of the schedule are predicted for each rolling unit using the estimator and if one of the constraints is violated, then the scheduled mission is replaced with putting the correspondent unit at rest.

Moreover, the used representation of chromosomes does not allow to directly use the genetic operators (mutation and crossover). Therefore, before applying these operators, the chromosomes undergo a special treatment to take out maintenance activities. This treatment consists of placing rolling stock units at rest if they are scheduled for maintenance.

Since the maintenance plans are excluded from the genetic operators, the resulted chromosomes from mutation and crossover represent mission assignments only. Hence, a reverse operation is required to schedule maintenance activities. A heuristic is defined to fix the offspring of genetic operators by scheduling maintenance and respecting the different constraints to obtain a feasible solution. This heuristic tries also to schedule the still unassigned mission to the available rolling units that could be missed by the genetic algorithm. Thus, it extends the search space of the problem.

At the end of the checking, fixing, and adding maintenance, we make sure that every considered offspring satisfy both validity and feasibility constraints. In other words, the offsprings are not allowed to have components with health indicators that exceeds the failure threshold which is the objective of using PHM i.e. maximize the use the components while avoiding their failure. However, accepting only offsprings that are feasible does not guarantee the absence of the corrective maintenance. During the decision applying loop, the selected schedule is simulated as if it was applied by the real set of trains. During this simulation, we generate the evolution of the degradation using the gamma process models presented in section 4.1.2.2. The gamma process takes into count the uncertainties in the evolution of the degradation. Hence, the obtained degradation level from the simulation could be and is probably different from the estimated one. This is mainly why we considered the implementation of the information loop, to account for this error of prognostics and re-adapt the estimator. The accumulation of this uncertainty could result in a simulated degradation level that exceeds the failure threshold of the component and thus, causing the component and the train to fail. This leads to adding a penalty in the cost function.

#### 4.3.1.4/ Mutation

The mutation is a genetic operator that provides spontaneous changes in the chromosome's genes. Usually, mutations are designed to alter one or more genes of a chromosome. Given the representation of the solution, it is necessary to adapt the mutation operator. Two mutation operators are used in this GA:

- **Simple Mutation:** First, the index of a column  $m$  (i.e., a vehicle) is randomly selected. Then, the schedule of the designated vehicle is altered. For each period  $i$ , if there are unassigned missions at period  $i$  (i.e., Eq. 4.19), then one of them is randomly assigned to  $e_{i,m}$ , else a second vehicle  $m_2$  is randomly selected and the elements  $e_{i,m}$  and  $e_{i,m_2}$  are interchanged.

$$\sum_{p=1}^P \sum_{m=1}^M \beta_{p,m}(i) \leq P \quad (4.19)$$

- **Exchange Mutation:** For this mutation, two indexes of columns (units)  $m_1$  and  $m_2$  to mutate are randomly selected. Then, for each period the schedules of the two vehicles are interchanged (Eq. 4.20).

$$e_{i,m_1} \Leftrightarrow e_{i,m_2}, \forall i \in \{1, \dots, I\} \quad (4.20)$$

#### 4.3.1.5/ Crossover

The Crossover is a genetic operator that operates on two individuals (called parents) at the same time. It combines the features of these parents to generate offsprings. We used a single-point order crossover adapted for a 2D-array representation of the solution. Basically, for each period the first part of alleles from the first parent drops down to the first child and remaining values are placed in the child in the order which they appear in the second parent. An example of this crossover is given in Fig. 4.6 for 6 vehicles two days example.

		<i>m</i> = 1	<i>m</i> = 2	<i>m</i> = 3	<i>m</i> = 4	<i>m</i> = 5	<i>m</i> = 6
Parent 1	<i>i</i> = 1	6	3	2	1	5	4
	<i>i</i> = 2	10	12	8	7	9	11
Parent 2	<i>i</i> = 1	1	4	6	3	2	5
	<i>i</i> = 2	11	10	12	9	8	7
Child 1	<i>i</i> = 1	6	3	2	5	1	4
	<i>i</i> = 2	10	12	8	9	7	11
Child 2	<i>i</i> = 1	1	4	6	5	3	2
	<i>i</i> = 2	11	10	12	7	9	8

Figure 4.6: Example of Single-Point Order Crossover.

#### 4.3.1.6/ Fitness Evaluation

Individuals in the GA are evaluated to measure their fitness toward an objective function. The fitness of a solution is defined as the value of the objective function. In this case, the fitness evaluation uses the definition of the total cost as in Eq. 4.13. The total cost is obtained by estimating the outcome of the chromosome. The effects of the execution of missions and maintenance activities are updated on the health indicators of each rail vehicle. And the total cost is computed progressively.

Each rail vehicle  $m$  is represented through a series of estimated health indicators  $\hat{H}_{m,k}$  and  $\hat{\theta}_{m,l}$ . For each period  $i$ , these indicators are updated according to the scheduled activity for the vehicle. For operational activities, the mission's degradation rate and its distance are added to the adequate virtual health indicators. Eq. 4.21 represents the update of vehicle  $m$  indicators at period  $i$  during which mission  $p$  is assigned to the vehicle (i.e.,  $e_{i,m} = p$ ). Maintenance activities are assumed to be perfect. Therefore, the update of maintenance is equivalent to reset the concerned component's virtual health indicator to zero. For example, if during period  $i$  component  $k$  of rail unit  $m$  is scheduled for maintenance (i.e.,



$e_{i,m} = -k$ ) then  $\hat{H}_{m,k}(i) = 0$ . At the end of the period, its corresponding cost is computed. Moreover, at the end of each period  $i$ , the cost of each activity is summed and the penalties on missing tasks are added if necessary. The fitness of a solution equals the sum of the periods' costs during the decision horizon  $DH$ .

$$\begin{cases} \hat{H}_{m,k}(i) = \hat{H}_{m,k}(i-1) + \delta_{p,k} & \forall k \in \{1, \dots, K\} \\ \hat{\theta}_{m,l}(i) = \hat{\theta}_{m,l}(i-1) + d_p & \forall l \in \{K+1, \dots, K+L\} \end{cases} \quad (4.21)$$

#### 4.3.1.7/ Construction of New Generation

The method to construct a novel generation is an important step in a GA. The transition from one generation to another is a driving force of the genetic search and evolutionary progress. New generations are created from previous generation survivors and genetic operators' offspring. In this context, a novel generation is built by; (i) a percentage  $X_{Survival}\%$  of best chromosomes from the previous generation, (ii) a percentage  $X_{Mutation}\%$  of best chromosomes of the mutation's offspring, and (iii) a percentage  $X_{Crossover}\%$  of best chromosomes of the crossover's offspring. The choice of these parameters subject to Eq. 4.22 influences the speed of convergence of the GA.

$$X_{Survival}\% + X_{Mutation}\% + X_{Crossover}\% = 100\% \quad (4.22)$$

#### 4.3.1.8/ Overall Algorithm

The overall GA is described in Algorithm 4. Lets note that parents selection in case of crossover is done according to the roulette wheel method [69].

#### 4.3.2/ Heuristics

Heuristics are strategies developed from previous experiences with the same problem. These approaches do not guarantee an optimal solution, but they are considered as practical methods to generate "good enough" solutions that are sufficient to reach an immediate goal, in reduced execution time.

For the resolution of the joint optimization of vehicle scheduling and maintenance planning, two heuristics are presented in this section. In all these heuristics, we used the remaining useful life (RUL) of the rolling stock units as a classification criteria. As presented in section 4.2, a rolling stock unit is assumed as a sequence of components  $K$  predictive and  $L$  preventive components. If any of these components fails, the whole unit e.g. train fails. Therefore, we assumed that the train's RUL is defined as the minimum RUL of its components. In this application, we propose to express the RUL in terms of miles. Therefore, we could also define a remaining useful life of a preventive component  $l$  as the remaining miles to reach the failure threshold  $\Theta_l$ . Let us denote  $RUL_{m,x}$  the remaining useful life of component  $x$  whether a preventive or predictive.  $RUL_m$  is the remaining useful life of rolling unit  $m$ . It is defined as in Eq 4.23.

$$RUL_m = \min[RUL_{m,x}] \quad \forall x \in \{1, \dots, K+L\} \quad (4.23)$$



**Algorithm 4:** GA for Rolling Stock Task Assignment and Maintenance Planning.

---

```

1 Create Initial_Generation;
2 Input_Generation  $\leftarrow$  Initial_Generation;
3 while number_of_generations < Generation_Limit do
4   Evaluate Individuals of the Input_Generation;
5   Sort Input_Generation ;
6   Survival of the best  $X_{Survival}\%$  Individuals of the Input_Generation;
7   Mutation_Res  $\leftarrow$  [ ];
8   Crossover_Res  $\leftarrow$  [ ];
9   for Individual  $\in$  Input_Generation do
10    Generate a Random Numbers a, b and c;
11    if a < Simple_Mutation_Probability then
12      Generate a random number pos for position;
13      Simple_Mutate Individual at position pos;
14      Add the offspring to Mutation_Res;
15    if b < Exchange_Mutation_Probability then
16      Generate two random numbers pos1 and pos2 for positions;
17      Exchange_Mutate Individual at positions pos1 and pos2;
18      Add the offspring to Mutation_Res;
19    if c < Crossover_Probability then
20      Generate one random numbers pos for position;
21      Select another parent from Initial_Generation;
22      Crossover Individual and the other parent at positions pos;
23      Add the offsprings to Crossover_Res;
24   Evaluate Individuals of the Mutation_Res;
25   Sort Mutation_Res;
26   Selection of the best  $X_{Mutation}\%$  Individuals of the Mutation_Res;
27   Evaluate Individuals of the Crossover_Res;
28   Sort Crossover_Res;
29   Selection of the best  $X_{Crossover}\%$  Individuals of the Crossover_Res;
30   Input_Generation  $\leftarrow$  Output_Generation;
31   number_of_generations++;

```

---

**4.3.2.1/ H1:**

Heuristic H1 is inspired by a “common sense” method of scheduling tasks and maintenance of rolling stocks. The heuristic consists of a succession of iterations corresponding to a decision for each period of the horizon  $DH$ . For each period  $i$  ( $1 \leq i \leq I$ ), the rolling stocks are sorted according to their RUL (in increasing order). Then, for each unit  $m$ , a set  $\mathcal{D}_{m,i}$  of possible decisions is constructed. this set is composed of put at rest action, maintenance intervention (containing all component eligible for maintenance) and all valid mission assignments (see Eqs. 4.3 and 4.4). Each decision of this set is associated with a regret value which is computed as a function of the rest of the decision horizon (periods  $i + 1$  to  $I$ ). The regret function depends on the nature of the decision to be taken and a set of rules that describe different scenarios that can occur.

Three rules have been used in this heuristics. For each rolling unit, they define a provisional

schedule for the rest of the horizon based on the potential decision for the current period. The potential decisions, in this study, are: maintenance, resting, mission performing. The rules are the following:

- **Rule 1:** for a maintenance decision, the rolling unit should be assigned to missions for the rest of the decision horizon.
- **Rule 2:** for a stay at rest, the rolling unit should be assigned to missions for the rest of the decision horizon.
- **Rule 3:** for mission performing, the rolling unit is put in maintenance or at rest during the next period ( $i + 1$ ) then, it should be assigned to missions for the rest of the decision horizon.

The outcome of the different rules is used to compute a regret value. This regret is computed for each rolling unit  $m$  and each possible decision  $dc \in \mathcal{D}_{m,i}$ . The regret value of one decision at period  $i$  is computed as the sum of two penalty values:

- **Maintenance penalty:** for each  $dc \in \mathcal{D}_{m,i}$ , the decision  $dc$  is supposed to be done, then the regret is an evaluation of the maintenance cost for the rest of the schedule. For that, the number of maintenance interventions per component is computed using Eqs. 4.24 and 4.25 (respectively for predictive and preventive components). We assume in these equations that the unit will always carry out the most severe missions for the rest of the decision horizon. The maintenance regret (noted  $Reg_m$ ) is then defined by Eq. 4.26.

$$N\_M(m, k) = \left\lfloor \frac{H_{m,k}(i+1) + \sum_{j=i+1}^{DH} \max_{p \in P_j} (\delta_{p,k})}{\Lambda_k} \right\rfloor \quad (4.24)$$

$$N\_M(m, l) = \left\lfloor \frac{\theta_{m,l}(i+1) + \sum_{j=i+1}^{DH} \max_{p \in P_j} (d_p)}{\Phi_l} \right\rfloor \quad (4.25)$$

$$\begin{aligned} Reg_m(dc, i) = & \sum_{k \in K} N\_M(m, k) * (CR_k + \frac{\Delta_k - \Lambda_k}{2} LP_k)) \\ & + \sum_{l \in L} N\_M(m, l) * (CR_l + LP_l * \frac{\Theta_l - \Phi_l}{2}) \end{aligned} \quad (4.26)$$

- **Operation penalty:** There are two parts in the operational regret (noted  $Reg_o$ ). The first one depends on the workload of maintenance operators. If this load is too high then some mission would be missed. the penalty corresponds to this evaluating cost, the first part of Eq. 4.27. The second part of the penalty concerns only maintenance decisions and putting rolling units at rest decision. If the number of remaining units *SortedEquip* is lower than the number of remaining tasks *RestMissions*, some missions cannot be achieved therefore causing cost penalty, the second part of Eq. 4.27.

$$\begin{aligned} Reg_o(dc, i) = & \tau * \frac{\sum_{x=1}^{K+L} N\_M(m, x)}{ML_{Comp}} * C_{lost} \\ & + (\omega_m(i) + \pi_m(i)) * MissionsToLose * C_{lost} \end{aligned} \quad (4.27)$$

With:

$$\begin{aligned} MissionsToLose = & card(RestMissions) \\ & - card(SortedEquip) \end{aligned}$$

Heuristic H1 provides a joint schedule of maintenance and tasks for a rolling decision horizon. The schedule is constructed based on the rolling stock unit's ability to carry out a task, the defined rules and the regret computation of possible actions. The process of decision-making of heuristic H1 for a decision horizon is presented in algorithm 5.

---

**Algorithm 5:** Heuristic H1.

---

```

1 while  $i \leq I$  do
2    $RestMissions \leftarrow [1..P]$ ;
3    $SortedEquip \leftarrow$  Rolling units sorted in increasing order of their RUL;
4   while  $SortedEquip \neq \emptyset$  do
5      $m \leftarrow$  first element of  $SortedEquip$ ;
6      $SortedEquip \leftarrow SortedEquip \setminus m$ ;
7      $\mathcal{D}_{m,i} \leftarrow []$ ;
8     if  $\exists k \in \{1, \dots, K\} / H_{m,k} \geq \Lambda_k$  Or  $\exists l \in \{K+1, \dots, K+L\} / \theta_{m,l} \geq \Phi_l$  then
9       | Add maintenance action to  $\mathcal{D}_{m,i}$ ;
10    Add 0 to  $\mathcal{D}_{m,i}$  (the action of putting  $m$  to rest);
11    for  $p \in RestMissions$  do
12      | if  $((H_{m,k} + \delta_{p,k}) < \Delta_k \forall k \in K)$  And  $((\theta_{m,l} + d_p) < \Theta_l \forall l \in L)$  then
13        | | Add task  $p$  to  $\mathcal{D}_{m,i}$ ;
14    Compute Regret for all decisions in  $\mathcal{D}_{m,i}$ ;
15    Choose the decision with the lowest regret value;
16    Update  $H_{m,k}(i+1) \forall k \in K$ ;
17    Update  $\theta_{m,l}(i+1) \forall l \in L$ ;
18    Update  $RestMissions$ ;
19   $i++$ 

```

---

#### 4.3.2.2/ H2:

Heuristic H2 is based on the health state of the rolling stocks. The aim is to guarantee a certain periodicity between the RULs of the vehicles in a way that the maintenance operation of different rolling units is well distributed. In this purpose, three sets of vehicles are defined according to their health state:

- **Rolling Units in Good Health:** The vehicles in this set ( $\mathcal{GH}$ ) can carry out a certain number of missions before needing any maintenance activities. To belong to this category, the RUL of the vehicle should be higher than a threshold  $Th_{GH}$ .
- **Rolling Units in Medium Health:** Once the RUL of a vehicle in Good Health falls behind  $Th_{GH}$  but still exceeds  $Th_{MH}$ , they are placed in the Medium Health Rolling Units set ( $\mathcal{MH}$ ).

- **Rolling Units in Poor Health:** Vehicles in this set ( $\mathcal{PH}$ ) have a RUL that falls behind  $Th_{MH}$ . These rolling units can carry on some missions depending on their severity.

The number of units in each set (i.e.,  $N_{PH}$ ,  $N_{MH}$ , and  $N_{GH}$ ) and the threshold of the sets (i.e.,  $Th_{GH}$  and  $Th_{MH}$ ) are the major decision variables for this heuristic. The assignment of missions is done in a way that tries to keep more or less the same number of vehicles in each category as defined by the parameters. Rolling Units in Poor Health are maintained to re-balance the Good Health and the Medium Health categories.

Two versions of heuristic H2 are developed in this paper, according to which service is prioritized over the other.

- **H2\_v1:** In this version of the heuristic H2, the maintenance activities are prioritized over the task assignment. Therefore at the beginning of the algorithm, the most deteriorated vehicles of  $\mathcal{PH}$  are placed into maintenance without checking their ability to carry on any of the missions. Once the maintenance activities are scheduled, the task is assigned while trying to appoint the most degraded unit to the hardest task it is capable of achieving it. This strategy allows the reduction of the remaining useful life wasted with maintenance activities. This heuristic is illustrated in algorithm 6. In this algorithm, two procedures are used *HealthSetsBalance* and *MissionAssignment*.

---

**Algorithm 6:** Heuristic H2\_v1.

---

**input** :  $\mathcal{PH}, \mathcal{MH}, \mathcal{GH}$

**input** :  $N_{PH}, N_{MH}, N_{GH}$

```

1 MaintenanceQueue  $\leftarrow []$ ;
2 while  $i \leq DH$  do
3   RestMissions  $\leftarrow \mathcal{P}$ ;
4   MaintenanceQueue  $\leftarrow \{m \in \mathcal{PH} \mid \exists k \in \{1, \dots, K\} \text{ where } H_{m,k} \geq \Lambda_k$ 
      or  $\exists l \in \{K+1, \dots, K+L\} \text{ where } \theta_{m,l} \geq \Phi_l$ 
      or  $\nexists p \in \text{RestMissions} \mid m \text{ is able to achieve } p\}$ ;
5   Sort MaintenanceQueue according to increasing RUL values;
6   while Maximum  $ML_{Equip}$  and  $ML_{Comp}$  are not met do
7     Plan Maintenance for the first unit of MaintenanceQueue;
8   Sort  $\mathcal{GH}, \mathcal{MH}$  and  $\mathcal{PH}$  according to increasing RUL values;
9   Sort RestMissions according to decreasing  $s_p$  values;
10  HealthSetsBalance( $\mathcal{GH}, \mathcal{MH}, \mathcal{PH}$  and RestMissions);
11  MissionAssignment( $\mathcal{GH}, \mathcal{MH}, \mathcal{PH}$  and RestMissions);
12  Update all  $H_{m,k}$  and  $\theta_{m,l} \forall m, k$  and  $l$ ;
13  Update  $\mathcal{PH}, \mathcal{MH}$  and  $\mathcal{GH}$ ;
14   $i++$ ;
```

---

- **H2\_v2:** Compared to H2\_v1, this version prioritizes task assignment over maintenance planning. All rolling units sets (i.e.,  $\mathcal{PH}$ ,  $\mathcal{MH}$ , and  $\mathcal{GH}$ ) are sorted according to their RUL. Most deteriorated vehicles of the poor health set  $\mathcal{PH}$  are checked to fulfill tasks before being sent to maintenance. In this heuristic, maintenance decisions are made while considering the possibility of having a bottleneck in the maintenance

**Procedure** HealthSetsBalance( $\mathcal{GH}, \mathcal{MH}, \mathcal{PH}$  and  $RestMissions$ )

---

```

1 Sort  $\mathcal{GH}$ ,  $\mathcal{MH}$  and  $\mathcal{PH}$  according to increasing RUL values;
2 Sort  $RestMissions$  according to decreasing  $s_p$  values;
3 forall  $X \in \{\mathcal{GH}, \mathcal{MH}, \mathcal{PH}\}$  do
4   while  $card(X) \geq N_X$  And  $RestMissions \neq \emptyset$  do
5      $m \leftarrow$  first element of  $X$ ;
6     forall  $p \in RestMissions$  do
7       if  $m \in X$  And  $m$  is able to achieve  $p$  (Eqs. 4.3, 4.4) then
8         Assign  $p$  to  $m$ ;
9          $RestMissions \leftarrow RestMissions \setminus p$ ;  $X \leftarrow X \setminus m$ ;

```

---

**Procedure** MissionAssignment( $\mathcal{GH}, \mathcal{MH}, \mathcal{PH}$  and  $RestMissions$ )

---

```

1 forall  $p \in RestMissions$  do
2   if  $\mathcal{PH} \neq \emptyset$  then
3     Assign last vehicles  $m$  of  $\mathcal{PH}$  to  $p$ ;  $\mathcal{PH} \leftarrow \mathcal{PH} \setminus m$ ;
4   else if  $\mathcal{MH} \neq \emptyset$  then
5     Assign last vehicles  $m$  of  $\mathcal{MH}$  to  $p$ ;  $\mathcal{MH} \leftarrow \mathcal{MH} \setminus m$ ;
6   else if  $\mathcal{GH} \neq \emptyset$  then
7     Assign last vehicles  $m$  of  $\mathcal{GH}$  to  $p$ ;  $\mathcal{GH} \leftarrow \mathcal{GH} \setminus m$ ;
8   else
9      $p$  is a missed mission;

```

---

workshop in the next period. Therefore, the number of vehicles (*WillNeedMaint* Algorithm 7) that would probably require a maintenance intervention in the next period is computed. If these vehicles are more than the maintenance limit  $ML_{Equip}$  in the next period ( $i + 1$ ), then some units will be scheduled to maintenance during the current period ( $i$ ) to avoid this bottleneck. Else, the heuristic assigns to each of these vehicles the hardest mission they can achieve. Tasks assignment, in this version, is similar to the previous version using the two procedures HealthSetsBalance and MissionAssignment. The choice of the task is based on their severity and the need to move a vehicle from one category to another. The H2\_v2 heuristic is presented in algorithm 7.

## 4.4/ Numeric application

### 4.4.1/ Case study

The numeric example is inspired by the problem presented by Herr *et al.* in [81]. The problem treats the assigning of a set of  $P = 15$  daily missions on a fleet of  $M = 18$  rolling vehicles over a simulation horizon  $SH = 300$  days where the periods are considered as days. The value of the simulation horizon is set to 300 days to guarantee that every component of the trains is at least maintained once during this horizon.

**Algorithm 7:** Heuristic H2\_v2.

---

```

input :  $\mathcal{PH}, \mathcal{MH}, \mathcal{GH}$ 
input :  $N_{PH}, N_{MH}, N_{GH}$ 
1  $MaintenanceQueue \leftarrow []$ ;
2 while  $i \leq DH$  do
3    $RestMissions \leftarrow \mathcal{P}$ ;
4   Sort  $RestMissions$  according to decreasing  $s_p$  values;
5   Sort  $\mathcal{GH}, \mathcal{MH}$  and  $\mathcal{PH}$  according to increasing RUL values;
6    $MaintenanceQueue \leftarrow \{m \in \mathcal{PH} \mid \nexists p \in RestMissions \mid m \text{ is able to achieve } p\}$ ;
7   while Maximum  $ML_{Equip}$  and  $ML_{Comp}$  are not met do
8     Plan Maintenance for the first units of  $MaintenanceQueue$ ;
9    $p \leftarrow$  first element of  $RestMissions$ ;
10   $WillNeedMaint \leftarrow \{m \in \mathcal{PH} \mid$ 
       $\exists k \in \{1, \dots, K\} \text{ where } H_{m,k} + \delta_{p,k} \geq \Lambda_k$ 
       $\text{ or } \exists l \in \{K+1, \dots, K+L\} \text{ where } \theta_{m,l} + d_p \geq \Phi_l\}$ ;
11  if  $Card(WillNeedMaint \cup MaintenanceQueue) \leq ML_{Equip}$  then
12    for  $m \in WillNeedMaint$  do
13      forall  $p \in RestMissions$  do
14        if  $m \in WillNeedMaint$  And  $m$  is able to achieve  $p$  (Eqs. 4.3, 4.4) then
15          Assign  $p$  to  $m$ ;
16           $RestMissions \leftarrow RestMissions \setminus p$ ;
17          Remove  $m$  from  $WillNeedMaint$ ;
18  else
19    while Maximum  $ML_{Equip}$  and  $ML_{Comp}$  are not met do
20      Plan Maintenance for the first units of  $WillNeedMaint$ ;
21  HealthSetsBalance( $\mathcal{GH}, \mathcal{MH}, \mathcal{PH}$  and  $RestMissions$ );
22  MissionAssignment( $\mathcal{GH}, \mathcal{MH}, \mathcal{PH}$  and  $RestMissions$ );
23  Update all  $H_{m,k}$  and  $\theta_{m,l} \forall m, k$  and  $l$ ;
24  Update  $\mathcal{PH}, \mathcal{MH}$  and  $\mathcal{GH}$ ;
25   $i++$ ;

```

---

Three types of missions are considered in this application. The task characteristics are presented in Table 4.9. Each rolling unit is composed of  $K = 13$  predictive components and  $L = 4$  preventive components. The characteristic of these components is defined in Tables 4.10 and 4.11 for predictive and preventive components respectively. All predictive components ( $k \in \{1..13\}$ ) have the same failure threshold  $\Delta_k = 0.95$  and maintenance threshold  $\Lambda_k = 0.7$ . All preventive components ( $l \in \{14..17\}$ ) have the same thresholds  $\Theta_l(\%) = 95\%$  and  $\Phi_k(\%) = 85\%$  of the mean mileage between maintenance.

Each day only  $ML_{Equip} = 2$  vehicles are allowed in the maintenance workshop, with a maximum total of  $ML_{Comp} = 4$  components to be maintained.  $LP_k$  and  $LP_l$  are set in a way that one lost mile of any component would cost 2u.m (unit of money). To summarize, all the used variables are presented in Table 4.12.

The decision horizon duration  $DH$  is defined as a divisor of the simulation horizon  $SH$  so that the number of decision-making steps  $N$  is an integer (i.e.,  $N \in \mathbb{Z}$ ).

Table 4.9: Characteristics of the Missions.

Type	Severity ( $s_p$ )	Length (mi)	Number
1	0.85	110	5
2	1	130	5
3	1.3	170	5

Table 4.10: Rolling Stocks Predictive Components Characteristics.

Type	Number	$\alpha_k$	$\mu_k$	$CR_k$
T_A	1	0,00346	0.002	100
T_B	2	0,0031	0.00178	150
T_C	2	0,01246	0.00166	75
T_D	8	0,00798	0.00208	100

In general, when dealing with train operational scheduling it is unusual to keep the same schedule after 60 days. Therefore, we only consider decision horizons that are a divisor of  $SH$  and have a duration of at most 60 days. In this case, the possible values for  $DH$  are defined in Eq. 4.28.

$$DH \in \{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 25, 30, 50, 60\} \quad (4.28)$$

#### 4.4.2/ Methods Comparison

For any value of the decision horizon duration and during the construction phase, the decision methods (whether GA or heuristic) make sure that the global decision selected is valid and feasible. In other words, the outcome of the decision building loop should satisfy the validity and feasibility constraints as defined in equations 4.3 and 4.4. During the construction phase, the global decision are evaluated according to the estimated evolution of the degradation. Therefore, the estimated cost of the global decisions does not include the possible corrective maintenance penalties. However, during the applying loop, the evolution of the degradation is simulated through the gamma process model. This model allows the presence of uncertainty. Thus, the simulated evolution of degradation would probably differ from the estimated one. During the application of the global decision, a simulated cost of this decision is computed according to equation 4.14 by including the simulated degradation evolution. The differences between the estimated and the simulated degradation evolution is accumulated through the decision horizon. Therefore, it could results in a degradation level that exceeds the failure threshold. And thus the failure of the train. The simulated cost of the global decision would then include the corrective maintenance penalty. For each decision horizon, no matter what decision method we use, we have two costs of the global decision; (i) an estimated total cost that we obtain during the construction phase and (ii) a simulated total cost that we obtain during the decision applying loop. The methods are compared to each other and their performance is evaluated based on the simulated total cost. The estimated total cost is used only inside the decision method. Therefore, in the rest of this chapter, we refer to the simulated total cost as the total cost. Respectively, the cumulative simulated total cost obtained by summing the simulated total costs over the simulation horizon, is referred to as the cumulative total cost.

Table 4.11: Rolling Stocks Preventive Components Characteristics.

Type	Number	Mean Mileage Between Maintenance(mi)	$CR_l$
T_E	1	31 250	100
T_F	3	15 625	100

Table 4.12: The Values of the Numeric Application.

Name	Significance	Value
$SH$	Simulation Horizon	300 Days
$M$	The Size of the Rolling Stock Fleet	18
$P$	Number of Daily Missions	15
$K$	Number of Predictive Components	13
$L$	Number of Preventive Components	4
$ML_{Equip}$	Capacity of Maintenance in Vehicles per Day	2
$ML_{Comp}$	Capacity of Maintenance in Components per Day	4
$LP$	Cost Penalty on a lost mile	2 u.m
$C_{lost}$	Cost of Missing a Mission	10 ku.m
$C_{cor}$	Cost of Failure During a Mission	100 ku.m

All the results are expressed in terms of the cumulative total cost over the simulation horizon which consists of the sum of the total cost over the decision horizons (see Eq. 4.15). This cost includes the cost of missed missions, corrective maintenance, missed mileage before maintenance, and normal maintenance. For example, for  $DH = 10$  days we have  $N = 30$  and the obtained cumulative total cost corresponds to the sum of the 30 total costs. With each total cost is obtained by solving the joint problem on a decision horizon of a duration equal to 10 days.

Results obtained from all the methods are presented in Fig. 4.7. The results of heuristic H1 are presented in the dashed orange line. The curve is almost stable around the (2800 k u.m) value. This can be explained by the fact that in H1 each vehicle is selecting the  $i^{th}$  decision based on the regret value. The regret values are computed for the rest of the decision horizon. Therefore, the longer the decision horizon gets the better the regret computation is, and thus the better the management of the vehicles. Although the cost obtained by H1 is improved from a longer decision horizon, the values are still far from the optimal solution. For this reason, H1 is not further studied in the rest of the chapter.

Results obtained by heuristic H2 (version 1 and 2) are represented by respectively dotted green line and dash-dot red line. We note that these results are almost stable regardless of the duration of the decision horizon. Also, these values are closer to the optimal solution.

In addition to the cumulative total cost over the simulation horizon, these methods are compared in terms of the total number of missed missions over the simulation horizon. In Fig. 4.8, we presented the mean number of missed missions from several simulations of the used methods (GA, H2\_v1, and H2\_v2). Each simulation corresponds to a test case with different initial conditions of the components. We can note that the curves of the missed missions are quite similar to the curves of the total cost. We can also notice



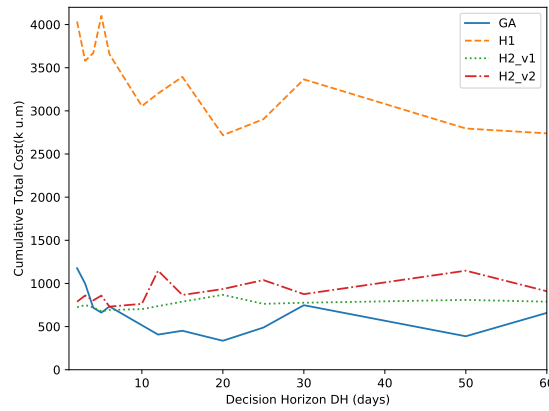


Figure 4.7: Results of the Different Methods.

that the genetic algorithm presents the minimal value of lost missions. We can observe that for very small decision horizon duration, the genetic algorithm could not see far ahead in the future to anticipate blocking situations. While for very large decision horizon, the computation complexity becomes more important and the genetic algorithm have too much information to process that it ends up losing its performance on one hand. On the other hand, the larger the decision horizon becomes the more important the accumulation of uncertainties is. Thus, the difference between the estimated total cost and the simulated cost becomes larger due to the presence of eventual corrective maintenance operations (Fig. 4.9 that represents the average number of failures for the genetic algorithm).

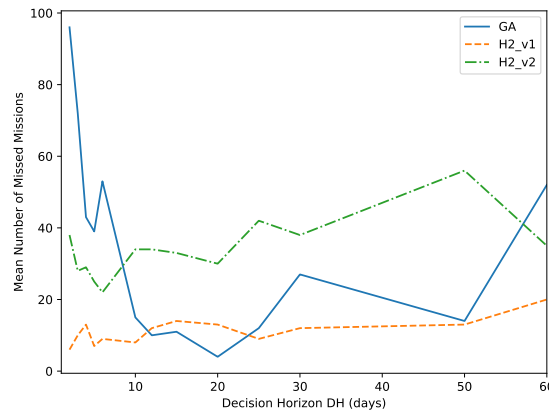


Figure 4.8: Mean Missed Missions of the Different Methods.

In Fig. 4.10, we present the mean value of the lost distance before maintenance. This performance is measured as the mean value for all components and rolling units. We can note that the mean lost distance is almost stable for the both heuristics with a small difference between them. However, it is clear that the genetic algorithm manages better the maintenance of the rolling stocks. The mean value of lost distance when using GA is minimal for small decision horizon.

Fig. 4.11 presents the execution time spent by the proposed methods to solve the problem over the simulation horizon for a specific decision horizon value. We easily note that the genetic algorithm takes much more time to find the solutions over the simulation horizon. We can note that the time spent by the GA to find solutions for the simulation horizon is

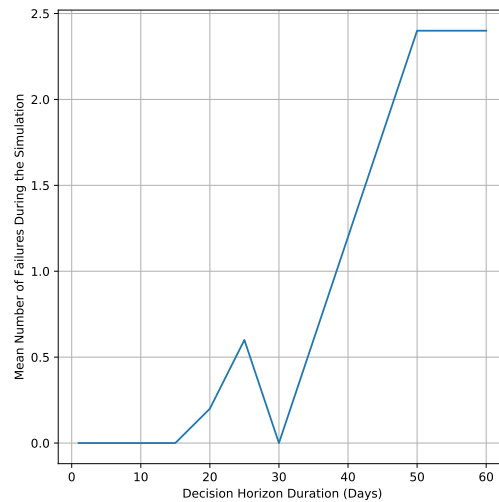


Figure 4.9: Mean Number of Failures Over the Simulation Horizon.

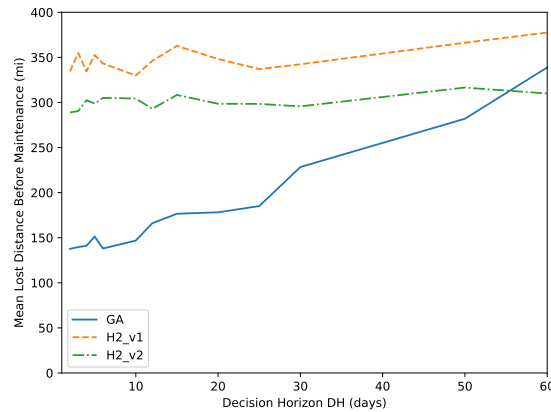


Figure 4.10: Mean Lost Mileage of the Different Methods.

decreasing exponentially. This is logical. For a decision horizon of duration 1 day, we need to execute the GA 300 times to solve the problem over the simulation horizon. While for a decision horizon of duration 2 day, we only need 150 executions of the GA. This explains the drop in the execution time. However, at some point, the execution time becomes almost constant due to the increased complexity of the computation. When the duration of the decision horizon increase, the chromosome representation becomes larger. Therefore, the manipulation of the chromosomes becomes more complex and thus more time consuming.

#### 4.4.3/ Decision Horizon Duration Study

In Fig. 4.7, the cumulative total cost of the solution over the simulation horizon depends on the duration of the chosen  $DH$ . This influence can be seen through the fluctuation of the different curves. Almost all the methods present some fluctuations. Based on this observation, a further investigation of this aspect is done by launching different simulations to see if these fluctuations are caused by the initial condition of the vehicles or the decision horizon. The obtained results are discussed per method in the following paragraphs.

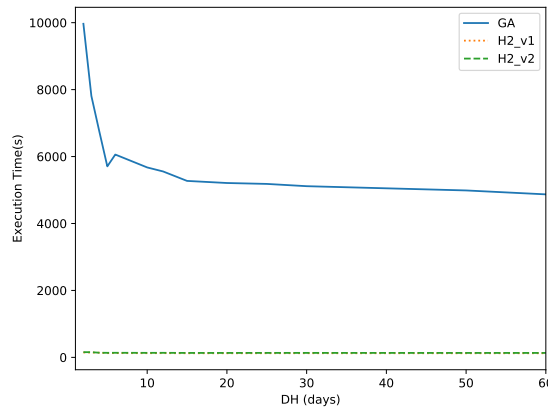


Figure 4.11: Execution Time of the Different Methods.

#### 4.4.3.1/ Genetic Algorithm

In the case of GA, the one-day decision horizon is ruled out of the study. The one-day decision horizon can provide locally optimized decisions for a certain number of periods by avoiding maintenance activities and using rolling units to their full degradation level. Then once all units are degraded, the algorithm can no longer schedule missions, thus all the missions are missed and no maintenance activities are done since it is less expensive to miss all missions without maintenance planned than to miss missions and plan maintenance interventions on the vehicles. This can be considered as a limit of the genetic algorithm performance. Fig. 4.12 presents these results where the cumulative cost of a one-day decision horizon is compared to the ten-days decision horizon. It is important to note that in Fig. 4.12, the time axis is limited to 80 days for clarity purposes.

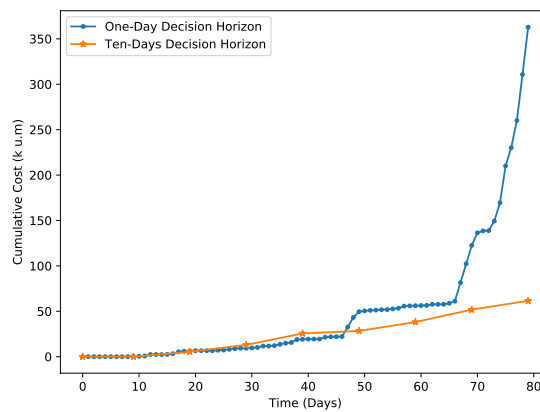


Figure 4.12: Cumulative cost of one-day and ten-days decision horizon.

When solving the same problem with different rolling vehicles' initial conditions, the obtained results of the cumulative total cost in function of the decision horizon have almost the same shape of fluctuation. These results can be seen in Fig. 4.13. The fact that the evolution of the total cost has the same shape for different initial conditions, proves that the initial conditions of the vehicles have no big influence on the decision horizon value in the case of using the genetic algorithm as a decision-making method.

The results presented in Fig. 4.13 are obtained through executing the genetic algorithm

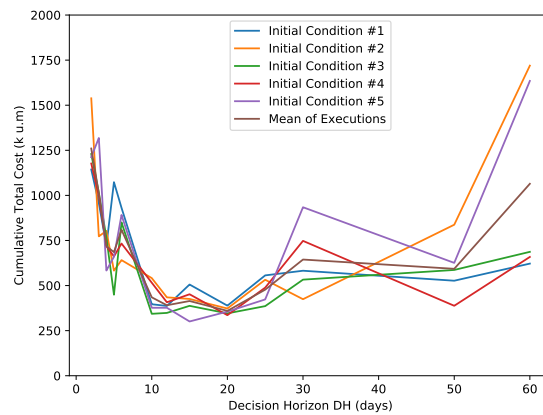


Figure 4.13: Total Cost evolution for different initial conditions.

for different initial conditions. We used these results to generate a box plot to display the variation of the total cost for each value of the decision horizon. For clarity purposes, the x-axis values, still, represents the decision horizon duration, but the scale is no longer applied.

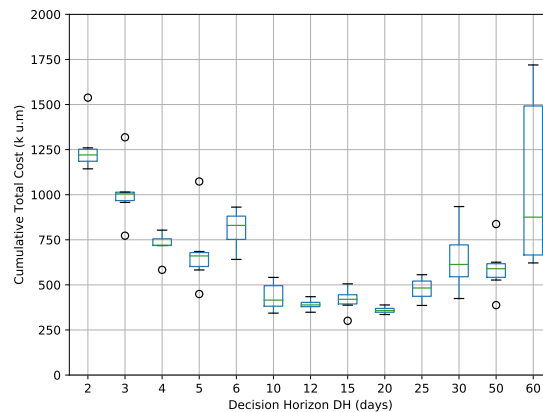


Figure 4.14: Box plot of total cost variation for the GA.

Fig. 4.13 and the box plot of Fig. 4.14 show that the cumulative total cost over the simulation horizon, starts high, decreases to a minimal value, then increases once more. The minimal value of the cumulative total cost is obtained for a decision horizon duration around 10-days to 25-days. For these same values of the decision horizon, the variation of the cumulative total cost is minimal from one execution to another. Thus, we conclude that for a task assignment and maintenance planning of rolling stocks as described in this problem and using a genetic algorithm, it is better to choose a rolling horizon of a duration that can vary from 10 to 25 days.

#### 4.4.3.2/ Heuristic H2\_v1 and H2\_v2

The variation of the cumulative total cost per decision horizon duration is captured in the case of the proposed heuristics H2\_v1 and H2\_v2. The obtained results confirm the observation of Fig. 4.7. The heuristics are almost stable for all different decision horizon duration. Although there are some variations, as shown by the box plots in Figs. 4.15

and 4.16 that represents results from heuristic H2\_v1 and H2\_v2 respectively.

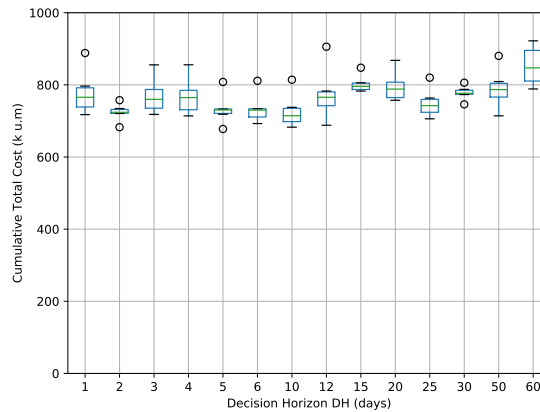


Figure 4.15: Box plot of total cost variation for heuristic H2\_v1.

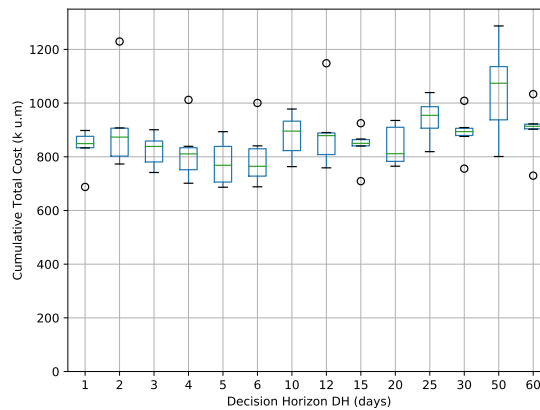


Figure 4.16: Box plot of total cost variation for heuristic H2\_v2.

Despite the small variation in the cumulative total cost for different decision horizon duration, minimal fluctuation and minimal values are obtained for a decision horizon of 5-days to 10-days in the case of heuristic H2\_v1. As for heuristic H2\_v2, the decision horizon of 4-days to 6-days presents the minimal variance of total cost and a globally minimal cost compared to other decision horizons.

#### 4.4.3.3/ Synthesis

The obtained values of decision horizon duration are different from one method to another. This proves that the decision horizon duration is a characteristic of the decision problem and depends also on the resolution method. The best decision horizon duration is presented in Table 4.13 per method. In the case of railway planning, it is more efficient to choose longer decision horizons. Therefore the genetic algorithm is more suited to solve this problem plus it provides the lowest total cost for decision horizons below 60-days.

Table 4.13: Best Decision Horizon per Method.

Method	Decision Horizon Duration
GA	$DH \in \{10, 12, 15, 20\}$
H2_v1	$DH \in \{5, 6, 10\}$
H2_v2	$DH \in \{4, 5, 6\}$

#### 4.4.4/ The effects of the problem's size on the decision horizon

In this section, the problem sizes are varied (number of trains  $M$  and the number of missions  $P$ ) to see the effects of the problem characteristics over the decision horizon duration. Proportions between the number of trains, the number of missions per period, and the maintenance constraints are kept the same. These variables and the obtained mean total cost over the multiple executions for each test case are grouped in Table 4.15. These results are shown in Figure 4.17. One can notice that the mean cost over the executions of each of the test cases is minimal for the same range of decision horizon duration. Therefore, one can conclude that the decision horizon duration influences the total cost over a simulation horizon and that when using a genetic algorithm, the number of trains and missions (i.e., the size of the problem) does not influence the optimal duration of the decision horizon.

Table 4.14: Components Configuration

K=13		L=4	
Type	Number	Type	Number
$T'_A$	1	$T'_G$	1
$T'_B$	2		
$T_C$	2	$T_H$	3
$T_D$	8		

Table 4.15: Configurations for Different Problem Sizes

M	18	25	45	90
P	15	21	38	75
$ML_T$	2	3	5	10
$ML_C$	4	6	10	20

#### 4.4.5/ The effects of the trains' configuration on the decision horizon

In this section, the configuration of the train is altered in the number of components per type and the characteristics of the components. The objective of this study is to see if the dynamics of the degradation of the components influence the duration of the decision horizon. In this section, all trains are composed of  $K = 13$  predictive components and  $L = 4$  preventive components. The dynamics of the component's degradation (mean traveled miles before the component's end of life) are varied through the variation of either the gamma process parameters ( $\alpha_k, \mu_k$ ) for predictive components or the MTBF for preventive components. Three categories of components are considered: (i) rapidly deteriorating ( $T_C, T_D, T'_D, T_H$ , and  $T'_H$ ), (ii) normally deteriorating ( $T'_A, T'_B, T_G$ , and  $T'_G$ ), and

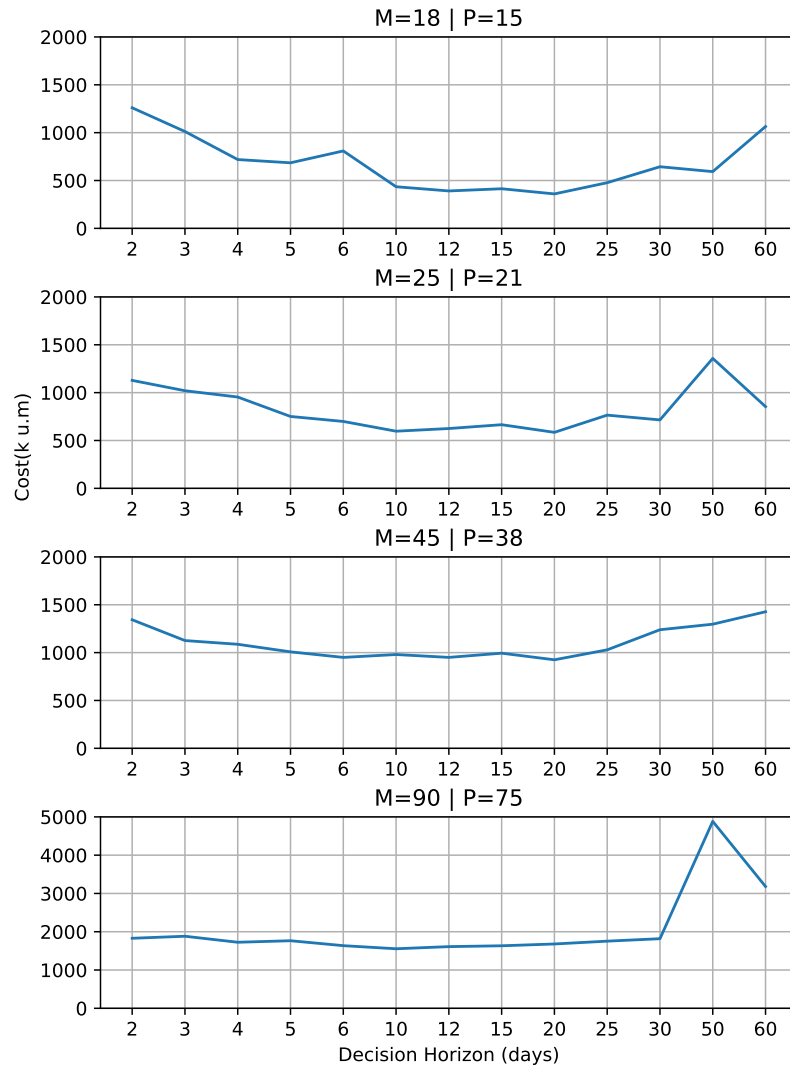


Figure 4.17: Mean Total Cost Variation Under Different Problem Sizes

(iii) slowly deteriorating ( $T_A$ ,  $T_B$ ,  $T_E$ , and  $T_F$ ). One can refer to Tables 4.10 and 4.11 for more information about these components characteristics.

For this purpose, four configurations of trains are considered:

- *Config\_1*: each train is composed of only rapidly deteriorating components ( $T_C$ ,  $T_D$ ,  $T'_D$ ,  $T_H$ , and  $T'_H$ ).
- *Config\_2*: this is the configuration used in the numeric example. In this case, each train is composed of a variety of normally deteriorating components and rapidly deteriorating components.
- *Config\_3*: each train is composed of only slowly deteriorating components ( $T_A$ ,  $T_B$ ,  $T_E$ , and  $T_F$ ).
- *Config\_4*: trains in this configuration are composed of components that have two times more slower degradation than those of *Config\_3*.

Figure 4.18 presents the mean total cost over different executions for each train configuration. One can note that the degradation dynamics of the components has a big influence on the total cost. For very slow degradation components, the cost is in the order of  $75ku.m.$ . While for rapidly deteriorating trains, the cost is in the order of  $10\,000ku.m.$ . This is explained that rapidly deteriorating components will be more frequently changed over the decision horizon compared to components that deteriorate slowly. Moreover, rapid deterioration of components will cause the failure of several trains in action and the avoidance of failure will cause more frequent maintenance activities. Since the maintenance resources are limited, this causes the system to have a lot of trains unavailable to achieve missions and waiting for maintenance. One can also notice that for medium and slow degradation dynamics (*Config<sub>2</sub>* - *Config<sub>4</sub>*) the decision horizon value is almost in the same interval (10 – 25days). However, this duration is different in the case of rapid degradation in which the more higher the decision horizon gets the lower the total cost is. This is explained by the fact that the genetic algorithm finds a way to schedule the maintenance of the predictive components in a systematic cyclic way. In this case, the cost generated by missing several missions is more expensive than the penalty of early maintenance. Thus, allowing the train to be maintained at the first opportunity regardless of the penalty that can be caused by this date of maintenance to guarantee its availability for the next period.

## 4.5/ Discussion

### 4.5.1/ Limits

Even though the described decision problem is more realistic from previous works in the literature, this formulation is still distant from a real world application. The main limits in this application are as follows:

- The rolling units in this application are assumed to be multi-component in a simple serial structure. While in real life situations, we find complex scheme for the components dependencies.
- Every component is critical to the unit's functions. However, in real life, some of the trains components are not critical to its operations. And, their failure will only cause user discomfort and hence introduce some cost penalties.
- Moreover, some components are installed in a redundant schemes which allows a more flexible corrective maintenance situation without introducing cost penalties.
- In real life, components are highly dependent one another, we know that the degradation of one component has an influence on the degradation evolution of another component. Such dependencies should be accounted for when dealing with PHM problems.
- In some cases, the early maintenance of a component could cost less if the rolling unit is already scheduled for the maintenance of another component. This could introduce an improvement to the proposed decision methods.
- The duration of maintenance for the components varies according to the component type. However, in the proposed application we supposed that all components require the same duration for maintenance.



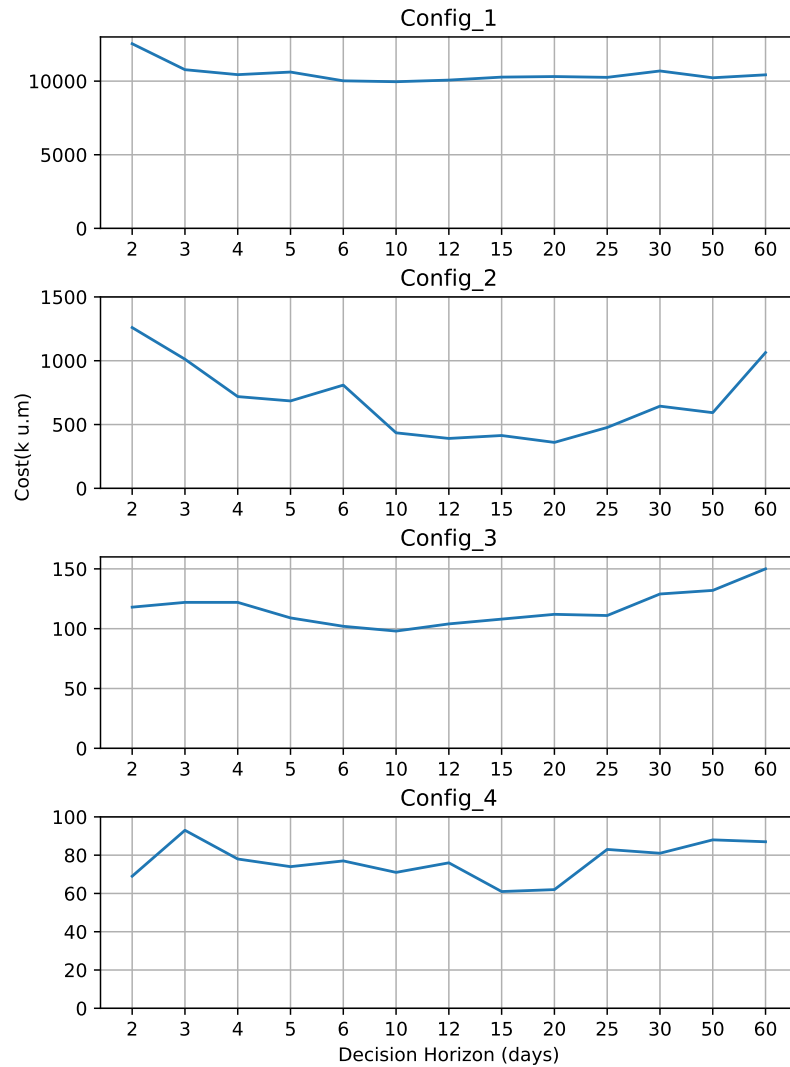


Figure 4.18: Mean Total Cost Variation Under Different Trains Configurations

Moreover, the framework instantiation presented is also quite limited. The limits of the implementation of the decision-enhanced framework can be summarized as follows:

- In PHM projects, the degradation modeling and the extraction of health indicators are quite challenging and not a straight forward process.
- Also, the development of a short-term prognostics algorithm could be a challenge in itself.
- The proposed approach is based on a prediction of health indicators for each possible combination of rolling unit and mission. For small problems this could be feasible. However, with large sets of rolling vehicles and missions the complexity and computation efforts would grow exponentially. And, at some point it would be absurd to compute all the combination.

Nevertheless, this research work is a step ahead in applying post-prognostics decisions in railways context.

#### 4.5.2/ Results Summary

To summarize, we have validated our idea about the influence of the duration of the decision horizon ( i.e. the frequency of executing the PHM process) on the performance of the decision module. Such a variable have not been studied in the literature. In this work, we proved the need to give the duration of the decision horizon the attention it requires to fully exploit the performances of the decision method and the framework.

We reinforced the fact that a sequence of local minimum does not lead to a global minimum in an optimization process. On the one hand, choosing a very short decision horizon would lead to a sequence of local minimum but the decision module would lose its ability to foreseen the possible future blocking situations such in the case of a 1 day decision horizon. On the other hand, a very large decision horizon would lead into large differences between the estimated values of the degradation evolution and their real evolution because the system is subject to more uncertainties. This might cause potential component failure, and hence, corrective maintenance that would raise the overall cost of the schedule.

Normally, the joint problem of rolling stock scheduling and maintenance planning is solved manually by dispatchers. The schedules are generally built over the duration of 7 to 10 days [200]. These values are very similar to the ones obtained by the common sense heuristics. While the genetic algorithm propose solutions that are scheduled over a larger horizons with lower costs. Maybe the duration of the manual 7 to 10 days is defined by the human capacity to analyze and consider several variables.

When varying the problem sizes i.e. the number of trains and the number of daily missions, we noticed that the values of the decision horizon are almost unchanged. However, the gained margin compared to other horizon gets lower the bigger the problem is. This could be caused by the fact that we kept the same parameters of the genetic algorithm for all different problem sizes. Moreover, the genetic algorithm could experience some drop in its performance once the problem size get bigger. Because, the exploration of the solutions space should be reinforced. This can be done by adding several genetic operations, increasing the size of a generation and the number of generations.

However, when changing the degradation speed of the railways vehicles' components, we noted that the values of the decision horizon are different. Moreover, for rapidly deteriorating components, the cumulative cost over the simulation horizon seems to be decreasing at first that it seems to be stable for any horizon's duration greater than 3 days. It seems like the genetic algorithm looses its capacity to treat the predictive components and tends to schedule their maintenance in a systematic cyclic way.

In this case study, when we changed the problem size we kept the proportions between the number of the trains, the number of missions, and the available maintenance resources. These proportions could be causing the maintenance or the operation capacities to be over-sized. Moreover, one can explore the effects of other problem parameters such as the proportion of trains and missions, or the different cost parameters on the total cost and the optimal decision horizon.

In this application, we focused on optimizing the decision horizon from a decision perspective. By implementing the information loop, we believe that the optimization of the decision horizon is indirectly impacted by the fine tuning of the prognostic method (i.e. the estimator). Moreover, the proposed framework defines a parameter that allows the optimization of the decision applying duration from a prognostics perspective. The partial decision parameter is defined as the applied part of the selected global decision between

two consecutive stages of PHM framework execution. An interesting future work is to study the performances of the decision method along with the prognostic method as a function of the duration of a partial decision. hence, instead of assuming that the built schedule would be applied in totally before re-acquiring new health data and built another plan (i.e., before launching another PHM cycle). One can imagine that the built schedule is not applied in totally but partially and investigate the prognostics and decision performances according to the new variables.

## 4.6/ Summary

This chapter presents the instantiation and implementation of the proposed post-prognostic decision framework on a more complex case study. The application consists of solving the joint problem of maintenance planning and mission assignment for a fleet of rolling vehicles. Each of the rolling stock unit is composed of several preventive and predictive components in a serial structure. In this case study all the loops of the proposed framework are implemented. Several decision modules have been proposed to solve the joint problem. Moreover, a second objective of this application is to find the best duration of the decision horizon and study what factors might influence this duration. The frequency of executing the decision-enhanced PHM framework seems to have a big influence on the overall performance of the decision method. The results obtained by using the genetic algorithm and the proposed common sense heuristics are compared over the values of the decision horizon. For short decision horizon duration, the decision method cannot look enough in the future to make coherent decisions instead, they tend to optimize local problem which could lead to blocking situation. For long decision horizon duration, the build decisions would be based on an estimated evolution of degradation that would eventually diverge from the real evolution due to the cumulative sum of uncertainties. This proves the importance of studying the decision horizon duration and the frequency of executing the PHM framework. The effects of the problem's size and the trains' configuration on the decision horizon duration have been studied. For this application, neither of the factors have a huge influence on the decision horizon. The chapter ended on a discussion of the important results and some perspectives for future investigation.



# Conclusion & future works

## Conclusion

Nowadays, enhancing the decision-making process has become challenging due to the presence of a variety of types and a huge amount of data. In any field of studies, the emerging technology of integrating data analysis and predictive modeling present a promising opportunity to boost the decision-making performance. In particular, in the industrial field, the use of prognostics and health management methodology is an emergent research context. The main goal of prognostics is to link the studies of degradation processes and the life cycle management of critical assets by intelligently using condition monitoring data and estimate the remaining useful life of the asset. However, obtaining the predictive information about the asset is not the final aim of prognostics and health management. Industries want to make full use of such information to better manage their equipment. In this thesis, we focused on the problem of enhancing the decision-making process (i.e. the health management part of PHM) by increasing the integration of predictive information in the decision process and enhancing the decisions in the prognostic phase.

In this purpose, we started by doing a synthesis of the current challenges in the prognostics and health management context. In particular, we focused on the challenges concerning the prognostics and decision-making phases. We performed a review of existing works in the post-prognostics decision-making. The previous works have been carefully studied to obtain a state-of-the-art summary. This allowed us to identify the scientific locks that we addressed all along this thesis.

Based on the findings of the bibliographic review, we have proposed new enhancements to the existing PHM framework. The main idea is to upgrade the decision-making process into a closed loop process, in which, decisions are considered in the prognostics process and a more integrated prognostics information in the decision-making module. This closed loop process is modeled through an iterative process of decision building. We called this enhancement the decision building loop. Another contribution to the existing PHM framework is the emphasis on the decision application and its influence on the evolution of the system health state. For this purpose, we proposed a formalization to the application of the selected decision on the system. The implementation of this loop, called decision applying loop, allows the user to define and study the frequency of execution of the PHM framework, and the definition of a suitable decision horizon. Finally, when decisions are applied to the system, we can expect the real evolution of the assets' degradation. This application generates new data from the system. We propose to collect this data, process it, and use it to evaluate the performance of the prognostics and potentially fine tune and maintain the prognostic models. Thus, this information loop allows the integration of a

dynamic evaluation for the prognostic models, an increased reactivity to changes in the system's operating conditions and a greater fidelity to the reality of the system conditions.

The proposed framework was then the subject of validation. For this aim, we proposed two applications of different nature to demonstrate the flexibility of the framework and a potential generic characteristic. At first, we proposed a single multi-purpose machine application, in which, we focused on implementing the decision building loop. In this application, we solved the maintenance and production joint scheduling problem. Then, we proposed a more developed application. This time we focused on a railway context, in which we applied the framework to the problem of jointly optimizing mission assignment and maintenance planning for a fleet of rolling stock units. The second application present the implementation of the full framework, i.e. the decision building, the decision applying, and the information loops.

Beside validating the framework, these applications presented other contribution to the prognostics and health management community. For the first application, we have compared the proposed method to a classic case of using condition-based maintenance with production scheduling. We have considered different study cases by varying parameters of the machines degradation speed, the production task to be achieve etc. By comparing the performance of the proposed approach and the CBM method, we discovered that unlike works in literature, PHM methodology is not always optimal. Moreover, if the intended application of the PHM framework presents several variation in the operating conditions, e.g. different speed configurations, different natures of tasks, etc, in these cases PHM outperform CBM. For the second application, we have focused on the study of the execution frequency of the proposed framework. We have also studied different configuration of the problems and investigated their influence on the decision horizon duration. We found that for a given problem and a decision method, the duration of the decision horizon has a huge influence on the long run performance of the system. Thus, this duration should be a subject of investigation when implementing the framework on real life applications.

Obviously no framework is perfect, however, the proposed enhancements on the existing prognostics and health management framework show great potential, tackled omitted problems in the decision-making process, and improved performances as compared to traditional methods. Although the work presented in this thesis contributes to the PHM community by emphasizing the decision-prognostics inter-dependencies, nevertheless improvements to these works can be conducted:

- As a start point, we assumed during the elaboration of this framework that sensors' data are of good quality and the process of extraction relevant health indicator is mastered. However, in real life the processing of sensors' data and the extraction of reliable health indicators is a challenging task in the PHM context. A large amount of uncertainties are generated in this phase. While in some cases this assumption is real, in most studies it is quite hard to find sustainable and robust health indicators.
- The observability of the degradation levels and health indicators is also a big challenge for the PHM community. However, most works tend to assume such hypothesis.
- In this thesis, we based the proposed framework on the assumption of existing short and long term prognostic algorithm. However, this is not always the case. Recent research are still studying methods to increase the performance of prognostics. This phase of the PHM framework is not yet mature and requires further studies. However, in our opinion, prognostics and decision-making phases of the PHM framework

should be developed simultaneously. As we have discussed the two processes are highly dependent and developing each one independently of the other would not speed up the advance of PHM. When developing novel predictive techniques and algorithms, the applicability and relevance to the decision phase should be considered.

Nevertheless, this research work is a step ahead in PHM domain toward the maturity of the prognostics and health management framework that focuses on the development of prognostics and decision-making jointly.

## Future work

Among the limits highlighted above, the key area of future focus is the applicability of the framework on a real life problem. This can be done by suppressing some of the hypothesis made. Therefore, current developments of this thesis will be further extended as follows:

1. On the short-term, increasing the complexity of the considered applications in this thesis would result in a more realistic test cases. In this aim, some of the hypothesis (e.g. increasing the dependencies between the components) could be suppressed, for in real life the degradation level of one component would influence the deteriorating speed of another. Moreover, more constraints about the maintenance and operation could be added (e.g. in the railway application, day-long missions and day-long maintenance duration could be reduced to half-day duration or even expressed in hours).
2. On the medium-term, the study of the applicability of the proposed framework on real life industrial problems is a very interesting perspective of this thesis. One challenging task is how to link real equipment to the framework. Knowing that in real life, the access to the components is quite difficult and the observability of their degradation is not always guaranteed. One can think of extracting health indicators that could present an approximation of the component degradation through vibratory analysis and temperature data. However, this would lead to a higher level of uncertainties in the RUL estimation. Therefore, one can study the effects of uncertainties in the observation phase on the performance of the proposed post-prognostic decision framework.
3. By applying the framework on different application domains (i.e. the multi-purposes production machine and railway rolling stocks), a potential generic characteristic of the framework is introduced. This track could be investigated further by studying the implementation of the proposed framework in other fields besides industrial context. One potential application is the medical field on long-term. One could study the applicability of the framework on the therapy decision-making process and the scheduling of treatments for breast cancer patients to increase their survival chances and life quality. The medical field is known for its high uncertainties level. There is no general rule of accurately how would a patient respond to a certain treatment. However, in general, similar treatments are subscribed to resembling patients. Therefore, the similarity analysis could be a starting point for the application of the new framework. The case-based reasoning is a potential decision method for such studies.





# Bibliography

- [1] Rosmaini Ahmad and Shahrul Kamaruddin. A review of condition-based maintenance decision-making. *European J. of Industrial Engineering*, 6(5):519, 2012.
- [2] J.I. Aizpurua, V.M. Catterson, Y. Papadopoulos, F. Chiacchio, and D. D'Urso. Supporting group maintenance through prognostics-enhanced dynamic dependability prediction. *Reliability Engineering and System Safety*, 168:171–188, December 2017.
- [3] Keiiti Aki. Asperities, barriers, characteristic earthquakes and strong motion prediction. *Journal of Geophysical Research: Solid Earth*, 89(B7):5867–5872, 1984.
- [4] Halim Alwi, Christopher Edwards, and Chee Pin Tan. Fault tolerant control and fault detection and isolation. In *Fault Detection and Fault-Tolerant Control Using Sliding Modes*, pages 7–27. Springer, 2011.
- [5] Saumil Ambani, Lin Li, and Jun Ni. Condition-Based Maintenance Decision-Making for Multiple Machine Systems. *Journal of Manufacturing Science and Engineering*, 131(3):031009, 2009.
- [6] Javier Andrés, Luis Cadarso, and Ángel Marín. Maintenance scheduling in rolling stock circulations in rapid transit networks. *Transportation Research Procedia*, 10:524–533, 2015.
- [7] Issam Baghdadi, Olivier Briat, Jean-Yves Delétage, Philippe Gyan, and Jean-Michel Vinassa. Lithium battery aging model based on dakin's degradation approach. *Journal of Power Sources*, 325:273–285, 2016.
- [8] Edward Balaban and Juan J. Alonso. An approach to prognostic decision making in the aerospace domain. In *Annual Conference of the Prognostics and Health Management Society*, 2012.
- [9] Egon Balas and Paolo Toth. Branch and bound methods for the traveling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1983.
- [10] Sumanta Basu and Diptesh Ghosh. A review of the tabu search literature on traveling salesman problems. 2008.
- [11] John E Bell and Patrick R McMullen. Ant colony optimization techniques for the vehicle routing problem. *Advanced engineering informatics*, 18(1):41–48, 2004.

- [12] Fatima Benbouzid, Christophe Varnier, and Nourredine Zerhouni. Resolution of joint maintenance/production scheduling by sequential and integrated strategies. In José Mira and José R. Álvarez, editors, *Artificial Neural Nets Problem Solving Methods*, volume 2687, pages 782–789. Springer Berlin Heidelberg, 2003.
- [13] G. Bencheikh, A. Letouzey, and X. Desforges. Process for joint scheduling based on health assessment of technical resources. In *10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2018*, volume 51, pages 192–199, Warsaw, Poland, august 2018.
- [14] Mogens Blanke, Michel Kinnaert, Jan Lunze, Marcel Staroswiecki, and Jochen Schröder. *Diagnosis and fault-tolerant control*, volume 2. Springer, 2006.
- [15] Alexandre Bogdanov, Stephen Chiu, Levent U. Gokdere, and John Vian. Stochastic Optimal Control of a Servo Motor with a Lifetime Constraint. In *Proceedings of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, May 2007. IEEE.
- [16] Brian Bole, Liang Tang, Kai Goebel, and George Vachtsevanos. Adaptive Load-Allocation for Prognosis-Based Risk Management. In *Annual Conference of the Prognostics and Health Management Society, 2011*, Montreal, Canada, July 2011.
- [17] Alexandros Bousdekis, Nikos Papageorgiou, Babis Magoutas, Dimitris Apostolou, and Gregoris Mentzas. A Proactive Event-driven Decision Model for Joint Equipment Predictive Maintenance and Spare Parts Inventory Optimization. *Procedia CIRP*, 59:184–189, 2017.
- [18] Mehdi Brahimi, Kamal Medjaher, Mohammed Leouatni, and Nouredine Zerhouni. Prognostics and health management for an overhead contact line system-a review. *International Journal of Prognostics and Health Management*, 8, 2017.
- [19] Jurgen Branke, Su Nguyen, Christoph W. Pickardt, and Mengjie Zhang. Automated design of production scheduling heuristics: A review. *IEEE Transactions on Evolutionary Computation*, 20(1):110–124, February 2016.
- [20] Douglas W. Brown, George Georgoulas, Brian Bole, Hai-Long Pei, Marcos Orchard, Liang Tang, Bhaskar Saha, Abhinav Saxena, Kai Goebel, and George Vachtsevanos. Prognostics Enhanced Reconfigurable Control of Electro-Mechanical Actuators. In *Annual Conference of the Prognostics and Health Management Society, 2009*, San Diego, CA, USA, September 2009.
- [21] Douglas W. Brown and George Vachtsevanos. A Prognostic Health Management Based Framework for Fault-Tolerant Control. In *Annual Conference of the Prognostics and Health Management Society, 2011*, Montreal, Canada, July.
- [22] Gabriella Budai, Dennis Huisman, and Rommert Dekker. Scheduling preventive railway maintenance activities. *Journal of the Operational Research Society*, 57(9):1035–1044, 2006.
- [23] C.S. Byington, M. Watson, and D. Edwards. Data-driven neural network methodology to remaining life predictions for aircraft actuator components. In *Proceedings of the IEEE Aerospace Conference*. IEEE, 2004.

- [24] Jing Cai, Xin Li, and Xi Chen. Optimization of Aeroengine Shop Visit Decisions Based on Remaining Useful Life and Stochastic Repair Time. *Mathematical Problems in Engineering*, 2016:1–11, 2016.
- [25] Jing Cai, Xin Li, and Xi Chen. Joint optimization of maintenance inspection and spare provisioning for aircraft deteriorating parts. *Journal of Systems Engineering and Electronics*, 28(6):1133 – 1140, December 2017.
- [26] Jing Cai, Yibing Yin, Li Zhang, and Xi Chen. Joint Optimization of Preventive Maintenance and Spare Parts Inventory with Appointment Policy. *Mathematical Problems in Engineering*, 2017:1–12, 2017.
- [27] F. Camci. System Maintenance Scheduling With Prognostics Information Using Genetic Algorithm. *IEEE Transactions on Reliability*, 58(3):539–552, September 2009.
- [28] Fatih Camci. The travelling maintainer problem: integration of condition-based maintenance with the travelling salesman problem. *Journal of the Operational Research Society*, 65(9):1423–1436, September 2014.
- [29] Fatih Camci. Maintenance scheduling of geographically distributed assets with prognostics information. *European Journal of Operational Research*, 245(2):506–516, September 2015.
- [30] Brigitte Chebel-Morello, Jean-Marc Nicod, and Christophe Varnier. *From Prognostics and Health Systems Management to Predictive Maintenance 2*. John Wiley & Sons, 2017.
- [31] Xiaohui Chen, Dawei Xu, and Lei Xiao. Joint optimization of replacement and spare ordering for critical rotary component based on condition signal to date. *Eksploatacja i Niezawodność - Maintenance and Reliability*, 19(1):76–85, December 2016.
- [32] Guo Qing Cheng, Bing Hai Zhou, and Ling Li. Integrated production, quality control and condition-based maintenance for imperfect production systems. *Reliability Engineering and System Safety*, 175:251–264, July 2018.
- [33] Yung-Hsiang Cheng and Hou-Lei Tsao. Rolling stock maintenance strategy selection, spares parts' estimation, and replacements' interval calculation. *International Journal of Production Economics*, 128(1):404–412, 2010.
- [34] Michael E. Cholette, Merve Celen, Dragan Djurdjanovic, and John D. Rasberry. Condition Monitoring and Operational Decision Making in Semiconductor Manufacturing. *IEEE Transactions on Semiconductor Manufacturing*, 26(4):454–464, November 2013.
- [35] Stéphane Chrétien, Nathalie Herr, Jean-Marc Nicod, and Christophe Varnier. Post-Prognostics Decision for Optimizing the Commitment of Fuel Cell Systems. In *3rd IFAC Workshop on Advanced Maintenance Engineering, Services and Technology AMEST 2016*, volume 49, pages 168–173, Biarritz, France, october 2016.
- [36] Yiqian Cui, Junyou Shi, and Zili Wang. Discrete Event Logistics Systems (DELS) simulation modeling incorporating two-step Remaining Useful Life (RUL) estimation. *Computers in Industry*, 72:68–81, September 2015.

- [37] Matthew Daigle and Kai Goebel. improving computational efficiency of prediction in model-based prognostics using the unscented transform. *Annual Conference of the Prognostics and Health Management Society 2010*, 2010.
- [38] Lawrence Davis. Handbook of genetic algorithms. 1991.
- [39] Ivo Paixão De Medeiros, Leonardo Ramos Rodrigues, Christian Strottmann Kern, Rafael Duarte Coelho Dos Santos, and Elcio Hideiti Shiguemori. Integrated task assignment and maintenance recommendation based on system architecture and PHM information for UAVs. In *Systems Conference (SysCon), 2015 9th Annual IEEE International*, pages 182–188. IEEE, 2015.
- [40] Ivo Paixão de Medeiros, Leonardo Ramos Rodrigues, Rafael Santos, Elcio Hideiti Shiguemori, and Cairo Lúcio Nascimento Júnior. PHM-based Multi-UAV task assignment. In *Systems Conference (SysCon), 2014 8th Annual IEEE*, pages 42–49. IEEE, 2014.
- [41] Rommert Dekker, Ralph E Wildeman, and Frank A Van der Duyn Schouten. A review of multi-component maintenance models with economic dependence. *Mathematical methods of operations research*, 45(3):411–435, 1997.
- [42] Xavier Desforges, Mickael Diévar, and Bernard Archimède. A prognostic function for complex systems to support production and maintenance co-operative planning based on an extension of object oriented Bayesian networks. *Computers in Industry*, 86:34–51, April 2017.
- [43] Feng Ding and Zhengjia He. Cutting tool wear monitoring for reliability analysis using proportional hazards model. *The International Journal of Advanced Manufacturing Technology*, (57):565–574, 2011.
- [44] Siew-Hong Ding and Shahrul Kamaruddin. Maintenance policy optimization—literature review and directions. *The International Journal of Advanced Manufacturing Technology*, 76(5-8):1263–1283, February 2015.
- [45] Phuc Do, Alexandre Voisin, Eric Levrat, and Benoit lung. A proactive condition-based maintenance strategy with both perfect and imperfect maintenance actions. *Reliability Engineering and System Safety*, 133:22–32, January 2015.
- [46] Hancheng Dong, Xiaoning Jin, Yangbing Lou, and Changhong Wang. Lithium-ion battery state of health monitoring and remaining useful life prediction based on support vector regression-particle filter. *Journal of power sources*, 271:114–123, 2014.
- [47] Xinfeng Dong and Weimin Zhang. Degradation analysis of grinding machine spindle systems based on complexity. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 229(8):1467–1471, 2015.
- [48] M. Dorigo and G. Di Caro. Ant colony optimization: a new meta-heuristic. pages 1470–1477. IEEE, 1999.
- [49] Marco Dorigo and Luca Maria Gambardella. Ant colonies for the travelling salesman problem. *biosystems*, 43(2):73–81, 1997.

- [50] Isidro Durazo-Cardenas, Andrew Starr, Christopher J. Turner, Ashutosh Tiwari, Leigh Kirkwood, Maurizio Bevilacqua, Antonios Tsourdos, Essam Shehab, Paul Baguley, Yuchun Xu, and Christos Emmanouilidis. An autonomous system for maintenance scheduling data-rich complex infrastructure: Fusing the railways' condition, planning and cost. *Transportation Research Part C: Emerging Technologies*, 89:234–253, April 2018.
- [51] Graham Elliott and Allan Timmermann. Economic forecasting. *Journal of Economic Literature*, 46(1):3–56, 2008.
- [52] EN13306: Maintenance Terminology. Standard, European Standard, 2001.
- [53] Stephen J Engel, Barbara J Gilmartin, Kenneth Bongort, and Andrew Hess. Prognostics, the real issues involved with predicting life remaining. In *2000 IEEE Aerospace Conference. Proceedings (Cat. No. 00TH8484)*, volume 6, pages 457–469. IEEE, 2000.
- [54] Manuel Esperon-Miguez, Philip John, and Ian K Jennions. A review of integrated vehicle health management tools for legacy platforms: challenges and opportunities. *Progress in Aerospace Sciences*, 56:19–34, 2013.
- [55] Ding Feng, Sheng Lin, Zhengyou He, and Xiaojun Sun. A technical framework of phm and active maintenance for modern high-speed railway traction power supply systems. *International Journal of Rail Transportation*, 5(3):145–169, July 2017.
- [56] Qiang Feng, Xiong Bi, Xiujie Zhao, Yiran Chen, and Bo Sun. Heuristic hybrid game approach for fleet condition-based maintenance planning. *Reliability Engineering and System Safety*, 157:166–176, January 2017.
- [57] Gerald R Ferris, Dwight D Frink, Maria Carmen Galang, Jing Zhou, K Michele Kacmar, and Jack L Howard. Perceptions of organizational politics: Prediction, stress-related implications, and outcomes. *Human relations*, 49(2):233–266, 1996.
- [58] Mohamed-Chahir Fitouhi and Mustapha Nourelfath. Integrating noncyclical preventive maintenance scheduling and production planning for multi-state systems. *Reliability Engineering & System Safety*, 121:175–186, January 2014.
- [59] Cyrine Fitouri, Nader Fnaiech, Christophe Varnier, Farhat Fnaiech, and Noureddine Zerhouni. A Decison-Making Approach for Job Shop Scheduling with Job Depending Degradation and PredictiveMaintenance. volume 49, pages 1490–1495, Troyes, France, 2016.
- [60] Richard Foote. Mathematics and complex systems. *Science*, 318(5849):410–412, 2007.
- [61] Philip Hans Franses, Philip Hans BF Franses, et al. *Time series models for business and economic forecasting*. Cambridge university press, 1998.
- [62] Susan Frost, Kai Goebel, Alan Frost, Khanh Trinh, and Mark Balas. Integrating systems health management with adaptive controls for a utility-scale wind turbine. In *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 314. 2013.
- [63] Joseph J George. *Weather forecasting for aeronautics*. Academic press, 2014.

- [64] Giovanni Luca Giacco, Donato Carillo, Andrea D'Ariano, Dario Pacciarelli, and Ángel G Marín. Short-term rail rolling stock rostering and maintenance scheduling. *Transportation Research Procedia*, 3:651–659, 2014.
- [65] Harry R Glahn and Dale A Lowry. The use of model output statistics (mos) in objective weather forecasting. *Journal of applied meteorology*, 11(8):1203–1211, 1972.
- [66] Kai Goebel and Alice Agogino. Milling data set. NASA Ames Prognostics Data Repository, (<http://ti.arc.nasa.gov/project/prognostic-data-repository>), NASA Ames Research Center, Moffett Field, CA.
- [67] Kai Goebel, Jose Celaya, Shankar Sankararaman, Indranil Roychoudhury, Matthew J. Daigle, and Abhinav Saxena. *Prognostics: The Science of Making Predictions*. CreateSpace Independent Publishing Platform, 2017.
- [68] Kai Frank Goebel. *Management of uncertainty in sensor validation, sensor fusion, and diagnosis of mechanical systems using soft computing techniques*. University of California, Berkeley, 1996.
- [69] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
- [70] Nima Gorjian, Lin Ma, Murthy Mittinty, Prasad Yarlagadda, and Yong Sun. A review on degradation models in reliability analysis. In *Engineering Asset Lifecycle Management*, pages 369–384. Springer, 2010.
- [71] Rafael Gouriveau, Kamal Medjaher, and Noureddine Zerhouni. *From Prognostics and Health Systems Management to Predictive Maintenance 1*. John Wiley & Sons, 2016.
- [72] Stephen C. Graves. A review of production scheduling. *Operations Research*, 29(4):646–675, August 1981.
- [73] D. Todd Griffith, Nathanael C. Yoder, Brian Resor, Jonathan White, and Joshua Paquette. Structural health and prognostics management for the enhancement of offshore wind turbine operations and maintenance strategies: Structural health and prognostics management for offshore O&M. *Wind Energy*, 17(11):1737–1751, November 2014.
- [74] Juan M. Grosso, Carlos Ocampo-Martinez, and Vicenç Puig. Reliability-based economic model predictive control for generalised flow-based networks including actuators' health-aware capabilities. *International Journal of Applied Mathematics and Computer Science*, 26(3), January 2016.
- [75] Gilbert Haddad, Sandborn Peter, and Michael Pecht. A Real Options Optimization Model to Meet Availability Requirements for Offshore Wind Turbines. In *Proceedings of MFPT: The Applied Systems Health Management Conference*, Virginia Beach, VA, USA, 2011.
- [76] Wael Hafsa, Brigitte Chebel-Morello, Christophe Varnier, Kamal Medjaher, and Noureddine Zerhouni. Prognostics of health status of multi-component systems with degradation interactions. In *International Conference on Industrial Engineering and Systems Management*. IEEE, 2015.

- [77] Aiwina Heng, Sheng Zhang, Andy CC Tan, and Joseph Mathew. Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical systems and signal processing*, 23(3):724–739, 2009.
- [78] Nathalie Herr. *Post-Prognostic scheduling of heterogeneous distributed platforms*. PhD thesis, Franche-Comte University, Besancon, France, 2015.
- [79] Nathalie Herr, Jean-Marc Nicod, and Christophe Varnier. Prognostics-based scheduling in a distributed platform: Model, complexity and resolution. In *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*, pages 1054–1059. IEEE, 2014.
- [80] Nathalie Herr, Jean-Marc Nicod, Christophe Varnier, Louise Jardin, Antonella Sorrentino, Daniel Hissel, and Marie-Cécile Péra. Decision process to manage useful life of multi-stacks fuel cell systems under service constraint. *Renewable Energy*, 105:590–600, May 2017.
- [81] Nathalie Herr, Jean-Marc Nicod, Christophe Varnier, Noureddine Zerhouni, Malek Cherif, and Nader Fnaiech. Joint optimization of train assignment and predictive maintenance scheduling. Lille, France, 2017.
- [82] Nathalie Herr, Jean-Marc Nicod, Christophe Varnier, Noureddine Zerhouni, and Pierre Dersin. Predictive maintenance of moving systems. In *Prognostics and System Health Management Conference (PHM-Harbin) (2017)*, pages 1–6. IEEE, 2017.
- [83] Andrew Hess, Giulio Calvello, Peter Frith, SJ Engel, and David Hoitsma. Challenges, issues, and lessons learned chasing the “big p”: real predictive prognostics part 2. In *2006 IEEE Aerospace Conference*, pages 1–19. IEEE, 2006.
- [84] Khac Tuan Huynh, Anne Barros, and Christophe Berenguer. Multi-Level Decision-Making for The Predictive Maintenance of  $k$ -Out-of- $n$  Failing Deteriorating Systems. *IEEE Transactions on Reliability*, 64(1):94–117, March 2015.
- [85] Khac Tuan Huynh, Inmaculada Torres Castro, Anne Barros, and Christophe Berenguer. On the Use of Mean Residual Life as a Condition Index for Condition-Based Maintenance Decision-Making. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(7):877–893, July 2014.
- [86] K.T. Huynh, A. Grall, and C. Bérenguer. Assessment of diagnostic and prognostic condition indices for efficient and robust maintenance decision-making of systems subject to stress corrosion cracking. *Reliability Engineering and System Safety*, 159:237–254, March 2017.
- [87] Naresh Iyer, Kai Goebel, and Piero Bonissone. Framework for post-prognostic decision support. In *Aerospace Conference, 2006 IEEE*, pages 10–pp. IEEE, 2006.
- [88] Amit Kumar Jain and Bhupesh Kumar Lad. Dynamic Optimization of Process Quality Control and Maintenance Planning. *IEEE Transactions on Reliability*, 66(2):502–517, June 2017.
- [89] Anil K Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.

- [90] Chao Jin, Dragan Djurdjanovic, Hossein D. Ardakani, Keren Wang, Matthew Buzza, Behrad Begheri, Patrick Brown, and Jay Lee. A comprehensive framework of factory-to-factory dynamic fleet-level prognostics and operation management for geographically distributed assets. In *Automation Science and Engineering (CASE), 2015 IEEE International Conference on*, pages 225–230. IEEE, 2015.
- [91] Nirupam Julka, Annamalai Thirunavukkarasu, Peter Lendermann, Boon Ping Gan, Arnd Schirrmann, Helge Fromm, and Elaine Wong. Making use of prognostics health management information for aerospace spare components logistics network optimisation. *Computers in Industry*, 62(6):613–622, August 2011.
- [92] J. Kaabi, C. Varnier, and N. Zerhouni. Heuristics for scheduling maintenance and production on a single machine. volume vol.5, page 5. IEEE, 2002.
- [93] Kevin A Kaiser and Nagi Z Gebraeel. Predictive maintenance management using sensor-based degradation models. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(4):840–849, 2009.
- [94] Si Larabi Khelifati and Fatima Benbouzid-Sitayeb. A Multi-Agent Scheduling Approach for the Joint Scheduling of Jobs and Maintenance Operations in the Flow Shop Sequencing Problem. In Piotr Jędrzejowicz, Ngoc Thanh Nguyen, and Kiem Hoang, editors, *Computational Collective Intelligence. Technologies and Applications*, volume 6923, pages 60–69. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [95] E. Khoury, E. Deloux, A. Grall, and C. Bérenguer. On the Use of Time-Limited Information for Maintenance Decision Support: A Predictive Approach under Maintenance Constraints. *Mathematical Problems in Engineering*, 2013:1–11, 2013.
- [96] J. R. King and A. S. Spachis. Heuristics for flow-shop scheduling. *International Journal of Production Research*, 18(3):345–357, May 1980.
- [97] Chetan S Kulkarni, Gautam Biswas, José R Celaya, and Kai Goebel. Physics based degradation models for electrolytic capacitor prognostics under thermal over-stress conditions. *International Journal of Prognostics and Health Management*, 825, 2013.
- [98] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014.
- [99] Asma Ladj, Fatima Benbouzid-Si Tayeb, Christophe Varnier, Ali Ayoub Dridi, and Nacer Selmane. A Hybrid of Variable Neighbor Search and Fuzzy Logic for the permutation flowshop scheduling problem with predictive maintenance. *Procedia Computer Science*, 112:663–672, 2017.
- [100] Asma Ladj, Christophe Varnier, Fatima Benbouzid Si Tayeb, and Nouredine Zerhouni. Exact and heuristic algorithms for post prognostic decision in a single multifunctional machine. *International Journal of of Prognostics and Health Management*, 8(2), 2017.
- [101] Yung-Cheng Lai, Dow-Chung Fan, and Kwei-Long Huang. Optimizing rolling stock assignment and maintenance plan for passenger railway operations. *Computers & Industrial Engineering*, 85:284–295, 2015.



- [102] Y. Langeron, A. Grall, and A. Barros. A modeling framework for deteriorating control system and predictive maintenance of actuators. *Reliability Engineering and System Safety*, 140:22–36, August 2015.
- [103] Yves Langeron, Mitra Fouladirad, and Antoine Grall. Controlled systems, failure prediction and maintenance. In *8th IFAC Conference on Manufacturing Modelling, Management and Control MIM*, volume 49, pages 805–808, Troyes, France, june 2016.
- [104] Yves Langeron, Antoine Grall, and Anne Barros. Actuator Health Prognosis for Designing LQR Control in Feedback Systems. *Chemical engineering transactions*, 33, 2013.
- [105] Yves Langeron, Antoine Grall, and Anne Barros. Joint maintenance and controller reconfiguration policy for a gradually deteriorating control system. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 231(4):339–349, August 2017.
- [106] Pedro Larranaga, Cindy M. H. Kuijpers, Roberto H. Murga, Inaki Inza, and Sejla Dizdarevic. Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review*, 13(2):129–170, 1999.
- [107] Eugene L Lawler. The traveling salesman problem: a guided tour of combinatorial optimization. *Wiley-Interscience Series in Discrete Mathematics*, 1985.
- [108] Mitchell Lebold and Michael Thruston. Open Standards For Condition-Based Maintenance and Prognostic Systems. In *Maintenance and Reliability Conference (MARCON)*, Gatlinburg, USA, January 2001.
- [109] Jay Lee, Fangji Wu, Wenyu Zhao, Masoud Ghaffari, Linxia Liao, and David Siegel. Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. *Mechanical systems and signal processing*, 42(1-2):314–334, 2014.
- [110] Xin Lei and Peter A. Sandborn. PHM-based wind turbine maintenance optimization using real options. *International Journal of of Prognostics and Health Management*, 7(1):1–14, 2016.
- [111] Xin Lei and Peter A. Sandborn. Maintenance scheduling based on remaining useful life predictions for wind farms managed using power purchase agreements. *Renewable Energy*, 116:188–198, February 2018.
- [112] Gustavo de Novaes Pires Leite, Alex Maurício Araújo, and Pedro André Carvalho Rosas. Prognostic techniques applied to maintenance of wind turbines: a concise and specific review. *Renewable and Sustainable Energy Reviews*, 81:1917–1925, 2018.
- [113] Hongfei Li, Dhaivat Parikh, Qing He, Buyue Qian, Zhiguo Li, Dongping Fang, and Arun Hampapur. Improving rail network velocity: A machine learning approach to predictive maintenance. *Transportation Research Part C: Emerging Technologies*, 45:17–26, 2014.

- [114] Rong Li and Jennifer K. Ryan. A Bayesian Inventory Model Using Real-Time Condition Monitoring Information. *Production and Operations Management*, 20(5):754–771, September 2011.
- [115] Zhaojun Li, Jian Guo, and Ruolin Zhou. Maintenance scheduling optimization based on reliability and prognostics information. In *Reliability and Maintainability Symposium (RAMS), 2016 Annual*, pages 1–5. IEEE, 2016.
- [116] Boliang Lin. A short-term planning model for high-speed train assignment and maintenance scheduling. *arXiv preprint arXiv:1709.04324*, 2017.
- [117] Lin Lin, Bin Luo, and ShiSheng Zhong. Development and application of maintenance decision-making support system for aircraft fleet. *Advances in Engineering Software*, 114:192–207, December 2017.
- [118] Sheng Lin, Ao Zhang, and Ding Feng. Maintenance decision-making model based on POMDP for traction power supply equipment and its application. In *2016 Prognostics and System Health Management Conference (PHM-Chengdu)*, pages 1–6, Chengdu, China, October 2016. IEEE.
- [119] X. Lin, R.J.I. Basten, A.A. Kranenburg, and G.J. van Houtum. Condition based spare parts supply. *Reliability Engineering and System Safety*, 168:240–248, December 2017.
- [120] Qinming Liu, Ming Dong, F.F. Chen, Wenyuan Lv, and Chunming Ye. Single-machine-based joint optimization of predictive maintenance planning and production scheduling. *Robotics and Computer-Integrated Manufacturing*, 55:173–182, February 2019.
- [121] Qinming Liu, Ming Dong, Wenyuan Lv, and Chunming Ye. Manufacturing system maintenance based on dynamic programming model with prognostics information. *Journal of Intelligent Manufacturing*, February 2017.
- [122] Xinbao Liu, Tianji Yang, Jun Pei, Haitao Liao, and Edward A. Pohl. Replacement and Inventory Control for a Multi-Customer Product Service System with Decreasing Replacement Costs. *European Journal of Operational Research*, 273(2):561–574, March 2019.
- [123] Israel Lopez and Nesrin Sarigul-Klijn. A review of uncertainty in flight vehicle structural damage monitoring, diagnosis and control: Challenges and opportunities. *Progress in Aerospace Sciences*, 46(7):247–273, 2010.
- [124] Bin Luo and Lin Lin. Multi-objective decision-making model based on CBM for an aircraft fleet. Xi'an City, China, 2018.
- [125] Holger R Maier, Angus R Simpson, Aaron C Zecchin, Wai Kuan Foong, Kuang Yeow Phang, Hsin Yeow Seah, and Chan Lim Tan. Ant colony optimization for design of water distribution systems. *Journal of water resources planning and management*, 129(3):200–209, 2003.
- [126] Kurt Matyas, Tanja Nemeth, Klaudia Kovacs, and Robert Glawar. A procedural approach for realizing prescriptive maintenance planning in manufacturing industries. *CIRP Annals*, 66(1):461–464, 2017.

- [127] Peyman Mazidi, Lina Bertling Tjernberg, and Miguel A Sanz Bobi. Wind turbine prognostics and maintenance management based on a hybrid approach of neural networks and a proportional hazards model. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 231(2):121–129, April 2017.
- [128] Safa Meraghni, Labib Sadek Terrissa, Soheyb Ayad, Nouredine Zerhouni, and Christophe Varnier. Post-prognostics decision in cyber-physical systems. In *2018 International Conference on Advanced Systems and Electric Technologies (IC-ASET)*, pages 201–205, Hammamet, March 2018. IEEE.
- [129] Safa Meraghni, Labib Sadek Terrissa, Nouredine Zerhouni, Christophe Varnier, and Soheyb Ayad. A post-prognostics decision framework for cell site using Cloud computing and Internet of Things. In *Cloud Computing Technologies and Applications (CloudTech), 2016 2nd International Conference on*, pages 310–315. IEEE, 2016.
- [130] Daniel Merkle, Martin Middendorf, and Hartmut Schneck. Ant colony optimization for resource-constrained project scheduling. *IEEE transactions on evolutionary computation*, 6(4):333–346, 2002.
- [131] Ir Falco PJH Mooren Ceng MIMechE and Leo AM van Dongen. Application of remote condition monitoring in different rolling stock life cycle phases. *Procedia CIRP*, 11:135–138, 2013.
- [132] Luís Mira, António R Andrade, and Marta Castilho Gomes. Maintenance scheduling within rolling stock planning in railway operations under uncertain maintenance durations. *Journal of Rail Transport Planning & Management*, page 100177, 2020.
- [133] Ramin Moghaddass and Seyda Ertekin. Joint optimization of ordering and maintenance with condition monitoring data. *Annals of Operations Research*, 263(1):271–310, April 2018.
- [134] Lorrie Molent, Rhys Jones, Simon Barter, and Susan Pitt. Recent developments in fatigue crack growth assessment. *International Journal of fatigue*, 28(12):1759–1768, 2006.
- [135] J. Michael Moore. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science*, 15(1):102–109, September 1968.
- [136] Patrick Nectoux, Rafael Gouriveau, Kamal Medjaher, Emmanuel Ramasso, Brigitte Chebel-Morello, Nouredine Zerhouni, and Christophe Varnier. Pronostia: An experimental platform for bearings accelerated degradation tests. In *IEEE International Conference on Prognostics and Health Management, PHM'12.*, pages 1–8. IEEE Catalog Number: CPF12PHM-CDR, 2012.
- [137] Patrick Nectoux, Rafael Gouriveau, Kamal Medjaher, Emmanuel Ramasso, Brigitte Chebel-Morello, Nouredine Zerhouni, and Christophe Varnier. Pronostia: An experimental platform for bearings accelerated degradation tests. In *IEEE International Conference on Prognostics and Health Management, PHM'12.*, pages 1–8. IEEE Catalog Number: CPF12PHM-CDR, 2012.
- [138] Richard Neufville. Real options: dealing with uncertainty in systems planning and design. *Integrated Assessment*, 4(1):26–34, 2003.

- [139] Danh Ngoc Nguyen, L. Dieulle, and Antoine Grall. Feedback Control System with Stochastically Deteriorating Actuator: Remaining Useful Life Assessment. In *Proceedings of the 19th World Congress The International Federation of Automatic Control*, Cape Town, South Africa, August 2014.
- [140] Kim-Anh Nguyen, Phuc Do, and Antoine Grall. Multi-level predictive maintenance for multi-component systems. *Reliability Engineering and System Safety*, 144:83–94, December 2015.
- [141] Robin P. Nicolai and Rommert Dekker. *Optimal Maintenance of Multi-Component Systems: A Review*. In: *Complex System Maintenance Handbook*. Springer Series in Reliability Engineering, Springer, London, 2008.
- [142] Seyed A. Niknam, John E. Kobza, and J. Wesley Hines. Operation and maintenance decision-making using prognostic information. In *Reliability and Maintainability Symposium (RAMS), 2015 Annual*, pages 1–7. IEEE, 2015.
- [143] C. Nzukam, A. Voisin, E. Levrat, D. Sauter, and B. lung. A dynamic maintenance decision approach based on maintenance action grouping for HVAC maintenance costs savings in Non-residential buildings. In *20th IFAC World Congress*, volume 50, pages 13722–13727, July 2017.
- [144] C. Nzukam, A. Voisin, E. Levrat, D. Sauter, and B. lung. Opportunistic maintenance scheduling with stochastic opportunities duration in a predictive maintenance strategy. In *16th IFAC Symposium on Information Control Problems in Manufacturing INCOM*, volume 51, pages 453–458, Bergamo, Italy, June 2018.
- [145] Ershun Pan, Wenzhu Liao, and Lifeng Xi. A joint model of production scheduling and predictive maintenance for minimizing job tardiness. *The International Journal of Advanced Manufacturing Technology*, 60(9):1049–1061, June 2012.
- [146] Chanseok Park and William J Padgett. Stochastic degradation models with several accelerating variables. *IEEE Transactions on Reliability*, 55(2):379–390, 2006.
- [147] Ron J Patton. Fault-tolerant control: the 1997 situation. *IFAC Proceedings Volumes*, 30(18):1029–1051, 1997.
- [148] Eduardo Bento Pereira, Roberto Kawakami Harrop Galvão, and Takashi Yoneyama. Model predictive control using prognosis and health monitoring of actuators. In *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, pages 237–243. IEEE, 2010.
- [149] Hai Qiu, Jay Lee, Jing Lin, and Gang Yu. Robust performance degradation assessment methods for enhanced rolling element bearing prognostics. *Advanced Engineering Informatics*, 17(3-4):127–140, 2003.
- [150] Vikas Raman and Nasib Singh Gill. Review of different heuristic algorithms for solving travelling salesman problem. *International Journal of Advanced Research in Computer Science*, 8(5), 2017.
- [151] Leonardo Ramos Rodrigues, Ivo Paixao de Medeiros, and Christian Strottmann Kern. Maintenance cost optimization for multiple components using a condition based method. In *2015 Annual IEEE Systems Conference (SysCon) Proceedings*, pages 164–169, Vancouver, BC, Canada, April 2015. IEEE.

- [152] Gerhard Reinelt. *The traveling salesman: computational solutions for TSP applications*. Springer-Verlag, 1994.
- [153] Leonardo R. Rodrigues, Joao P. P. Gomes, Felipe A. S. Ferri, Ivo P. Medeiros, Roberto K. H. Galvao, and Cairo L. Nascimento Junior. Use of PHM Information and System Architecture for Optimized Aircraft Maintenance Planning. *IEEE Systems Journal*, 9(4):1197–1207, December 2015.
- [154] Leonardo R. Rodrigues, João P. P. Gomes, and João F. L. Alcântara. Embedding Remaining Useful Life Predictions into a Modified Receding Horizon Task Assignment Algorithm to Solve Task Allocation Problems. *Journal of Intelligent & Robotic Systems*, 90(1-2):133–145, May 2018.
- [155] Shankar Sankararaman, You Ling, Chris Shantz, and Sankaran Mahadevan. Uncertainty quantification in fatigue damage prognosis. In *Annual conference of the prognostics and health management society*, pages 1–13, 2009.
- [156] J-H Schaefer, F Jochimsen, F Keller, K Wegscheider, and A Distler. Outcome prediction of acute renal failure in medical intensive care. *Intensive care medicine*, 17(1):19–24, 1991.
- [157] M. Alaoui selsouli, N.M. Najib, A. Mohafid, and E.H. Aghezzaf. An Integrated Production and Maintenance Planning Model with time windows and shortage cost. *IFAC Proceedings Volumes*, 42(4):822–827, 2009.
- [158] Hui Shi and Jianchao Zeng. Real-time prediction of remaining useful life and preventive opportunistic maintenance strategy for multi-component systems considering stochastic dependence. *Computers & Industrial Engineering*, 93:192–204, March 2016.
- [159] Xiaosheng Si, Tianmei Li, Qi Zhang, and Xiaoxiang Hu. An Optimal Condition-Based Replacement Method for Systems With Observed Degradation Signals. *IEEE Transactions on Reliability*, 67(3):1281–1293, September 2018.
- [160] Gina Sierra, M Orchard, Kai Goebel, and C Kulkarni. Battery health management for small-size rotary-wing electric unmanned aerial vehicles: An efficient approach for constrained computing platforms. *Reliability Engineering & System Safety*, 182:166–178, 2019.
- [161] Haithem Skima. *Prognostics and distributed algorithms for post-prognostic decision making in MEMS based Systems*. PhD thesis, Franche-Comte University, Besancon, France, 2016.
- [162] Haithem Skima, Christophe Varnier, Eugen Dedu, Kamal Medjaher, and Julien Bourgeois. Post-prognostics decision making in distributed MEMS-based systems. *Journal of Intelligent Manufacturing*, February 2017.
- [163] Jyoti Soni, Ujma Ansari, Dipesh Sharma, and Sunita Soni. Predictive data mining for medical diagnosis: An overview of heart disease prediction. *International Journal of Computer Applications*, 17(8):43–48, 2011.
- [164] A. Soualhi, K. Medjaher, and N. Zerhouni. Bearing health monitoring based on hilbert–huang transform, support vector machine, and regression. *IEEE Transactions on Instrumentation and Measurement*, 64(1):52–62, Jan 2015.

- [165] Bo Sun, Shengkui Zeng, Rui Kang, and Michael Pecht. Benefits analysis of prognostics in systems. *2010 Prognostics and System Health Management Conference*, pages 1–10, January 2010.
- [166] Diyin Tang, Viliam Makis, Leila Jafari, and Jinsong Yu. Optimal maintenance policy and residual life estimation for a slowly degrading system subject to condition monitoring. *Reliability Engineering and System Safety*, 134:198–207, February 2015.
- [167] Liang Tang, Eric Hettler, Bin Zhang, and Jonathan DeCastro. A Testbed for Real-Time Autonomous Vehicle PHM and Contingency Management Applications. In *Annual Conference of the Prognostics and Health Management Society, 2011*, Montreal, Canada, July 2011.
- [168] Liang Tang, Gregory J Kacprzynski, Kai Goebel, and George Vachtsevanos. Methodologies for uncertainty management in prognostics. In *2009 IEEE Aerospace conference*, pages 1–12. IEEE, 2009.
- [169] Zhigang Tian, Tongdan Jin, Bairong Wu, and Fangfang Ding. Condition based maintenance optimization for wind power generation systems under continuous monitoring. *Renewable Energy*, 36(5):1502–1509, May 2011.
- [170] Zhigang Tian and Haitao Liao. Condition based maintenance optimization for multi-component systems using proportional hazards model. *Reliability Engineering and System Safety*, 96(5):581–589, May 2011.
- [171] Serdar Uckun, Kai Goebel, and Peter J.F. Lucas. Standardizing research methods for prognostics. In *International Conference on Prognostics and Health Management, 2008. PHM 2008.*, pages 1–10, October 2008.
- [172] G Vachtsevanos. Performance metrics for fault prognosis of complex systems. In *Proceedings AUTOTESTCON 2003. IEEE Systems Readiness Technology Conference.*, pages 341–345. IEEE, 2003.
- [173] Adriaan Van Horenbeek and Liliane Pintelon. A dynamic predictive maintenance policy for complex multi-component systems. *Reliability Engineering and System Safety*, 120:39–50, December 2013.
- [174] Jan M van Noortwijk. A survey of the application of gamma processes in maintenance. *Reliability Engineering & System Safety*, 94(1):2–21, 2009.
- [175] Christophe Varnier and Nouredine Zerhouni. Scheduling predictive maintenance in flow-shop. In *IEEE 2012 Prognostics and System Health Management Conference (PHM-2012 Beijing)*, pages 1–6. IEEE, May 2012.
- [176] K. Verbert, B. De Schutter, and R. Babuška. Timely condition-based maintenance planning for multi-component systems. *Reliability Engineering and System Safety*, 159:310–321, March 2017.
- [177] Wlamir Olivares Loesch Vianna and Takashi Yoneyama. Predictive Maintenance Optimization for Aircraft Redundant Systems Subjected to Multiple Wear Profiles. *IEEE Systems Journal*, 12(2):1170–1181, June 2018.

- [178] João Paulo Vieira, Roberto Kawakami Harrop Galvão, and Takashi Yoneyama. Predictive Control for Systems with Loss of Actuator Effectiveness Resulting from Degradation Effects. *Journal of Control, Automation and Electrical Systems*, 26(6):589–598, December 2015.
- [179] Marius Vileiniskis and Rasa Remenyte-Prescott. Quantitative risk prognostics framework based on petri net and bow-tie models. *Reliability Engineering and System Safety*, 165:62–73, 2017.
- [180] Roberto Villarejo, Carl-Anders Johansson, Diego Galar, Peter Sandborn, and Uday Kumar. Context-driven decisions for railway maintenance. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 230(5):1469–1483, July 2016.
- [181] Elizabeth Walter. *Cambridge advanced learner's dictionary*. Cambridge university press, 2008.
- [182] Cheng Wang, Jianxin Xu, Hongjun Wang, and Zhenming Zhang. A criticality importance-based spare ordering policy for multi-component degraded systems. *Eksploatacja i Niezawodność - Maintenance and Reliability*, 20(4):662–670, September 2018.
- [183] Kai Wang, Jing Tian, Michael Pecht, and Aidong Xu. A Prognostics and Health Management Based Method for Refurbishment Decision Making for Electromechanical Systems. In *15th IFAC Symposium on Information Control Problems in Manufacturing*, volume 48, pages 454–459, 2015.
- [184] Ling Wang, Jian Chu, and Jun Wu. Selection of optimum maintenance strategies based on a fuzzy analytic hierarchy process. *International journal of production economics*, 107(1):151–163, 2007.
- [185] Pingfeng Wang, Prasanna Tamilselvan, Janet Twomey, and Byeng Dong Youn. Prognosis-informed wind farm operation and maintenance for concurrent economic and environmental benefits. *International Journal of Precision Engineering and Manufacturing*, 14(6):1049–1056, June 2013.
- [186] Wenbin Wang. A scheduling model for systems with task and health dependent remaining useful life prognostics. *International Journal of Production Research*, 52(19):5764–5779, October 2014.
- [187] Yiwei Wang, Christian Gogu, Nicolas Binaud, Christian Bes, Raphael T. Haftka, and Nam H. Kim. A cost driven predictive maintenance policy for structural airframe maintenance. *Chinese Journal of Aeronautics*, 30(3):1242–1257, June 2017.
- [188] Yiwei Wang, Christian Gogu, Nicolas Binaud, Christian Bes, Raphael T Haftka, and Nam-Ho Kim. Predictive airframe maintenance strategies using model-based prognostics. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 232(6):690–709, December 2018.
- [189] Zhao-Qiang Wang, Wenbin Wang, Chang-Hua Hu, Xiao-Sheng Si, and Wei Zhang. A Prognostic-Information-Based Order-Replacement Policy for a Non-Repairable Critical System in Service. *IEEE Transactions on Reliability*, 64(2):721–735, June 2015.

- [190] Zhaoqiang Wang, Changhua Hu, Wenbin Wang, Xiangyu Kong, and Wei Zhang. A prognostics-based spare part ordering and system replacement policy for a deteriorating system subjected to a random lead time. *International Journal of Production Research*, 53(15):4511–4527, August 2015.
- [191] Zachary Welz, Jamie Coble, Belle Upadhyaya, and Wes Hines. Maintenance-based prognostics of nuclear plant equipment for long-term operation. *Nuclear Engineering and Technology*, 49(5):914–919, August 2017.
- [192] George M. Whitesides and Rustem F. Ismagilov. Complexity in chemistry. *Science*, 284(5411):89–92, 1999.
- [193] Zimin (Max) Yang, Dragan Djurdjanovic, and Jun Ni. Maintenance scheduling in manufacturing systems based on predicted machine degradation. *Journal of Intelligent Manufacturing*, 19(1):87–98, February 2008.
- [194] Aibo Zhang, Yiliu Liu, Anne Barros, and Yukun Wang. Prognostics and health management for safety barriers in infrastructures: Opportunities and challenges. In *Safety and Reliability—Safe Societies in a Changing World. Proceedings of ESREL 2018*, june 2018.
- [195] Bin Zhang, Liang Tang, Jonathan Decastro, and Kai Goebel. Prognostics-enhanced receding horizon mission planning for field unmanned vehicles. In *AIAA Guidance, Navigation, and Control Conference*, page 6294, 2011.
- [196] Bin Zhang, Liang Tang, Jonathan DeCastro, Michael Roemer, and Kai Goebel. Autonomous vehicle battery state-of-charge prognostics enhanced mission planning. *International Journal of of Prognostics and Health Management*, 5:1–11, 2014.
- [197] Youmin Zhang and Jin Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual reviews in control*, 32(2):229–252, 2008.
- [198] Zhengxin Zhang, Xiaosheng Si, Changhua Hu, and Yaguo Lei. Degradation data analysis and remaining useful life estimation: A review on wiener-process-based methods. *European Journal of Operational Research*, 271(3):775–796, 2018.
- [199] Xuejing Zhao, Mitra Fouladirad, Christophe Bérenguer, and Laurent Bordes. *Maintenance policy for deteriorating system with explanatory variables*. R&RATA, 2009.
- [200] Qingwei Zhong, Richard M Lusby, Jesper Larsen, Yongxiang Zhang, and Qiyuan Peng. Rolling stock scheduling with maintenance requirements at the chinese high-speed railway. *Transportation Research Part B: Methodological*, 126:24–44, 2019.



# **Appendixes**



## Appendix A

# Post-Prognostic Decisions Summary Tables

### A.1/ Maintenance Decisions

#### A.1.1/ Single Machine

Tables A.1, A.2, A.3, and A.4 describe the papers that studied the optimization of maintenance scheduling in the case of single machine. The description is made by specifying if they considered multi-components, the degradation model, the objective of the optimization and the method, how the RUL of the components/systems is used and the corresponding application domain. One can also see that stochastic processes, especially the gamma process, are extensively used as a degradation model for single machine maintenance decision-making. Almost all these works aim at minimizing maintenance cost. In most of the works, RUL is used in the decision-making process to determine the dates of maintenance interventions. These works actually do not present an optimization method other than a simple algorithm that compares the RUL to a safety threshold to schedule the maintenance intervention, or in the case of multiple components, the authors usually use a grouping algorithm to group the maintenance activities on the different component, assuming that maintaining two or more components at the same time costs less than maintaining each component alone.

#### A.1.2/ Multiple Machines

Tables A.5, A.6 and A.7 summarize the papers that studied the optimization of the maintenance plannings in the case of multi-machines systems. The tables compare the works by specifying if the papers considered multi-components, the used degradation model, the objective of the optimization and the used method, how the RUL of the components/systems are used and the corresponding application domain. The first point that one can notice in these tables, is that most of the works do not specify the used degradation model. Actually a common assumption in these works is that the RUL of the component and/or the system is supposed to be obtained from a prognostic process. One can deduce that the objective behind integrating prognostics and health management technologies in the case of multi-machines maintenance scheduling can be either cost-oriented (by

Table A.1: Existing papers for single machine maintenance scheduling. (Part 1)

Reference	Multi-Component	Degradation Model	Objective	Used Method	Integration of RUL	Application Domain
[199]		Continuous-State Stochastic Process	Minimize Maintenance Cost	Periodic Inspection CBM	Not Included	Manufacturing
[27]	✓	Weibull Distribution	Minimize Maintenance Cost	Genetic Algorithm	In Cost Function as Risk Penalty	Manufacturing
[170]	✓	Proportional Hazards Model	Minimize Maintenance Cost	Developed Algorithm	In Cost Function	Manufacturing
[173]	✓	Gamma Process	Minimize Long-Term Mean Cost per Unit Time,	Maintenance Grouping Algorithm	Maintenance Dates	Manufacturing
[151]	✓	Gamma Process	Minimize Maintenance Cost	Maintenance Grouping Algorithm	Maintenance Dates	Manufacturing
[95]	✓	Gamma Process	Minimize Maintenance Cost	None	Selection of Maintenance Opportunity	Manufacturing
[85]		Gamma Process	Minimize Long-Run Expected Maintenance Cost	None	Inspection Duration + Maintenance Date + Cost Function	Manufacturing

Table A.2: Existing papers for single machine maintenance scheduling. (Part 2)

Reference	Multi-Component	Degradation Model	Objective	Used Method	Integration of RUL	Application Domain
[86]		Gamma Process	Minimize Long-Run Expected Maintenance Cost	Generalized Pattern Search Algorithms	Inspection Duration + Maintenance Date + Cost Function	Manufacturing
[166]		Autoregressive Model	Minimize the Long-Run Expected Average Cost.	Semi-Markov Decision Process	Not Included	Manufacturing
[45]		Gamma Process	Minimize Maintenance Cost	Not Mentioned	Inspection Dates	Manufacturing
[103]		Gamma Process	Minimize the Long-Run Expected Average Cost	Not Mentioned	Maintenance Dates	Manufacturing
[158]	✓	Not Mentioned	Minimize the Long-Run Expected Average Cost	Particle Swarm Optimization	In Cost Function + Maintenance Dates	Manufacturing
[121]		Not Mentioned	Minimize the Long-Run Expected Cost	Dynamic Programming	In Cost Function	Manufacturing
[183]	✓	Not Mentioned	Minimize Refurbishment Cost	None	Selection of Maintenance Policy + Maintenance Dates	Manufacturing

Table A.3: Existing papers for single machine maintenance scheduling. (Part 3)

Reference	Multi-Component	Degradation Model	Objective	Used Method	Integration of RUL	Application Domain
[110]	✓	Not Mentioned	Minimize Maintenance Cost	Simulation-Based Real Options Analysis	Selecting Maintenance Opportunity + In Cost Function	Wind Turbines
[127]		Data-Driven	Minimize Maintenance Cost	Proportional Hazards Model + Neural Network	Maintenance Dates + In Cost Function	Wind Turbines
[118]		Gauss-Poisson Process	Minimize Maintenance Cost	Partially Observable Markov Decision Process + Dynamic Programming	In Cost Function as Transition Probability	Railways
[55]	✓	Data-Driven	Guarantee a Level of Reliability	Not Mentioned	Selection of Maintenance Strategy	Railways
[176]	✓	Continuous-Time Stochastic Process	Maximize Component lifetime + Maximize Cost Benefits	Markov Decision Process + Maintenance Grouping Algorithm	Maintenance Dates + In Cost Function	Railways
[159]	✓	Wiener Process	Minimize Maintenance Cost	Markov Decision Process	Not Included	Aerospace
[177]	✓	Multiple Model	Minimize Maintenance Cost	Monte Carlo Simulations	In Cost Function	Aerospace
[187]	✓	Physic-Based	Minimize Maintenance Cost	Not Mentioned	Maintenance Dates	Aerospace

Table A.4: Existing papers for single machine maintenance scheduling. (Part 4)

Reference	Multi-Component	Degradation Model	Objective	Used Method	Integration of RUL	Application Domain
[188]	✓	Physic-Based	Minimize Maintenance Cost	Not Mentioned	Maintenance Dates	Aerospace
[143]	✓	Not Mentioned	Minimize Maintenance Cost	Odds Algorithm + Maintenance Grouping Algorithm	Selection of Maintenance Opportunity + In Cost Function	HVAC
[144]	✓	Not Mentioned	Minimize Maintenance Cost	Odds Algorithm + Maintenance Grouping Algorithm	Selection of Maintenance Opportunity + In Cost Function	HVAC
[84]	✓	Gamma Process	Minimize Long-Run Maintenance Cost	Monte Carlo Simulations	Maintenance Dates	Not Mentioned
[140]	✓	Gamma Process	Minimize Long-Run Maintenance Cost per Time Unit	Monte Carlo Simulation	Selection of Maintenance Opportunity	Not Mentioned

minimizing maintenance costs, or maximizing profits) or reliability-oriented (by minimizing failure probabilities and maximizing availability and reliability). As these tables show, most works that focused on multi-machine systems, use the remaining useful life as either a penalty on the waste of RUL in the cost function or as a criteria to define the maintenance dates by comparing it to a safety threshold.

## **A.2/ Operational Decisions**

### **A.2.1/ Production and Mission Scheduling**

Tables A.8 and A.9 contain the papers that treated production or mission planning in the context of PHM. Most of these works are built on the assumption of having a prognostic module that provides the RUL of the machines and/or their components. This assumed prognostic module also provides the health state requirement of certain production jobs or mission and how much they affect the systems' RUL. For these reasons, most of these works do not specify the degradation model or the method used to estimate the RUL. Instead, they focus on the decision process and its optimization procedure. The objectives of these papers can be classified into three categories: (i) time-oriented objective (in which the focus is on extending the planning horizon or the life time of the system. or minimize the mission duration); (ii) cost-oriented objective by maximizing the benefits; or (iii) reliability-oriented by minimizing failure risk. Various optimization algorithms are used in the decision-making process: local search algorithms, linear programming, path-finding methods and task assignment methods. The integration of the RUL into the decision-making process for production and mission planning is more dynamic than for maintenance decisions. This can be observed through the use of RUL not only as a penalty in the cost function or as a constraint but also as a decision variable for selecting the suitable path or by making a reject list in the case of mission assignment. Another important point is that most of these decision-making processes are based on one RUL estimation for the whole decision-making process. Thus, the dynamics of RUL estimation are not considered in Table ??.

### **A.2.2/ Logistics Planning**

The works that treated only logistics planning are summarized in Table A.10. One can note that stochastic processes are commonly used for the logistics decision-making, to implement the uncertainties in estimations. Although all these works have a common cost-oriented objective to minimize the inventory and logistics costs, various methods are used to achieve that. As for the RUL, it is estimated with different dynamics from only one time to a periodic estimation, and it is used mainly to define the ordering dates for the spare parts, taking into consideration the lead-time to deliver the orders. Most of the works that treat logistic decisions, are in aerospace or heavy manufacturing applications, in which the spare parts are large and thus require a large storage facilities. Thus the cost of inventory is expensive, and the storage of multiple spare parts can be difficult.



Table A.5: Existing papers for multi-machine maintenance scheduling. (Part 1)

Reference	Multi-Component	Degradation Model	Objective	Used Method	Integration of RUL	Application Domain
[90]		Data-Driven	Minimize Maintenance Cost	Genetic Algorithm	Used For Clustering	Geographically Distributed
[129]		Data-Driven	Minimize the Traveled Distance	Genetic Algorithm	Maintenance Dates	Geographically Distributed
[2]	✓	Constant Degradation	None	New Grouping Maintenance Algorithm	Maintenance Dates	Geographically Distributed
[28]		Weibull Distribution	Minimize the Total Cost	Genetic Algorithm and Particle Swarm Optimisation	In Cost Function	Railways
[29]		Weibull Distribution	Minimize the Total Cost	Genetic Algorithm	In Cost Function	Railways
[50]	✓	Data-Driven	Minimize the Overall Cost	job sequencing heuristic and genetic algorithms	In Cost Function	Railways
[180]	✓	Data-Driven	Minimize Slowdowns and Shutdowns	Minimum Bayes Risk	Maintenance Dates	Railways
[193]	✓	Not Mentioned	Maximize the Profit	Genetic Algorithm	In Cost Function	Manufacturing
[5]		Continuous Time Markov Chain	Maximize the Long Run Average Profit	Choosing the Solution that Maximize the Objective	In Cost Function	Manufacturing

Table A.6: Existing papers for multi-machine maintenance scheduling. (Part 2)

Reference	Multi-Component	Degradation Model	Objective	Used Method	Integration of RUL	Application Domain
[126]	✓	Data-Based	Minimize Maintenance Cost and Maximize Availability	Rule Based System	Rule Definition	Manufacturing
[128]		Not Mentioned	Minimize Traveling Cost	Genetic Algorithm	Maintenance Dates + Cost	Manufacturing
[169]	✓	Data-Driven	Minimize Maintenance Cost	Simulation-Based CBM	Maintenance Dates	Wind Turbines
[75]		Not Mentioned	Minimize Maintenance Cost	Real Options Analysis	In Cost Function + Maintenance Date	Wind Turbines
[111]	✓	Data-Driven	Maximize Cumulative Revenue	Monte Carlo simulation-based European real options analysis	Selection of Maintenance Date	Wind Turbines
[185]		Not Mentioned	Minimize Maintenance Cost	Simulation-Based CBM	Maintenance Dates	Wind Turbines
[87]		Not Mentioned	Minimize Production Cost and Reduce Cycle Time and Maximize Reliability and Quality.	Genetic Algorithm	In Cost Function + Selecting Aircraft to Maintain	Aerospace

Table A.7: Existing papers for multi-machine maintenance scheduling. (Part 3)

Reference	Multi-Component	Degradation Model	Objective	Used Method	Integration of RUL	Application Domain
[153]	✓	Not Mentioned	Minimize the Component Replacement Cost	Ant Colony Optimization	Maintenance Dates	Aerospace
[24]	✓	Data-Driven	Maximize Aircraft Availability	Not Mentioned	Maintenance Dates	Aerospace
[115]		Not Mentioned	Minimize Maintenance Cost	Linear Programming	As Constraint + In Cost Function	Aerospace
[117]	✓	Physic Based	Minimize Maintenance Cost	Genetic Algorithm	In Cost Function	Aerospace
[56]	✓	Not Mentioned	Minimize Maintenance Cost	Game Theory	Cost Function + Maintenance Dates	Aerospace
[124]	✓	Not Mentioned	Minimize Maintenance Cost	Support Vector Regression	In Cost Function	Aerospace

Table A.8: Existing papers for production and mission planning. (Part 1)

Reference	Degradation Model	Objective	Used Method	Dynamics of RUL Estimation	Integration of RUL	Application
[167]	Data-Driven	Minimize Mission Failure Risk	$D^*$ Search Algorithm	Periodic	Path Selection + In Cost Function	Autonomous Vehicle
[34]	Hidden Markov Model	Maximize Benefits	Tabu Search Algorithm	One Time Before Launching The Decision-Making Process	In Cost Function	Manufacturing
[79]	Not Mentioned	Maximize the Production Horizon	Integer Linear Program	One Time Before Launching The Decision-Making Process	Selecting The Production Profile + As Constraint	Not Mentioned
[196]	Physic-Based	Easiest Path, Minimize the Wear of Components + Minimize Mission Failure Risk	$D^*$ Search Algorithm	At Each Step of the Decision Algorithm	In Selecting the Path + In Cost Function	Autonomous Vehicle
[40]	Not Mentioned	Minimize the Total Mission Time	Receding Horizon Task Assignment	One Time Before Launching The Decision-Making Process	In UAV-Mission Matching + In Cost Function	Unmanned Aerial Vehicle
[154]	Weibull Distribution	Minimize Mission Time	Receding Horizon Task Assignment	One Time Before Launching The Decision-Making Process	Creating the Reject List + In Cost Function	Unmanned Aerial Vehicle

Table A.9: Existing papers for production and mission planning. (Part 2)

Reference	Degradation Model	Objective	Used Method	Dynamics of RUL Estimation	Integration of RUL	Application
[35]	Not Mentioned	Minimize the Shortage in Energy Production	Mirror Prox Algorithm + Adaptive Lasso Algorithm	One Time Before Launching The Decision-Making Process	Selecting the Production Profile + As Constraint	Fuel Cells
[80]	Not Mentioned	Maximize the Production Horizon	Mixed Integer Linear Programming	One Time Before Launching The Decision-Making Process	Selecting the Production Profile + As Constraint	Fuel Cells
[162]	Not Mentioned	Maximize the Lifetime of the Surface	Dijkstra's Algorithm	One Time Before Launching The Decision-Making Process	In Cost Function + Selecting The Path	Conveying System for MEMS

Table A.10: Existing papers for logistics.

Reference	Degradation Model	Objective	Used Method	Dynamics of RUL Estimation	Integration of RUL	Application
[114]	Wiener Process	Minimize the Expected Discounted Cost-to-Go	Dynamic Program	Periodic	Ordering Spare Parts	Manufacturing
[91]	Net Mentioned	Minimize Inventory and Logistics Cost	Discrete Event Logistics Systems	One Time Before Launching The Decision-Making Process	Ordering Spare Parts	Aerospace
[36]	Gaussian Stochastic Process	Minimize Inventory and Logistics Cost	Discrete Event Logistics Systems	One Time Before Decision-Making and One Time in the middle of the Decision Horizon	Ordering Spare Parts	Aerospace
[119]	Markov Decision Process	Minimize Inventory and Logistics Cost	Heuristics	Periodic	Ordering Spare Parts	Aerospace

### **A.2.3/ Automatic Control**

Tables A.11 and A.12 summarize the works on the automatic control optimization while taking the health status into consideration. Most aim to compute new control set points or controllers' parameters taking into consideration the health state of the actuator. Since the application can be considered a low-level control process, the degradation models are most likely based on the physics of the degradation of the actuator. In some cases, when the physics of degradation are complex the authors tend to use stochastic processes to model the deterioration of the actuator. As Table ?? shows, almost all the works aim to extend the lifetime of the actuator by balancing the performance and the control power. Automatic control loops are characterized with high speed feedback dynamic. Thus, the estimation of RUL needs to follow the same high speed that is guaranteed by the chosen degradation model and the estimation techniques. Consequently, the decision-making process needs to match the speed of the feedback dynamic and the speed of changes in the actuators' health state. Therefore, one can note that works that investigate automatic control usually avoid classic optimization algorithms, which can take a long time to compute and find solutions, and to use analytical methods instead. As for the RUL, it is most commonly used as a constraint in the computation of new control settings or included in the function that defines the new settings or in the cost function itself.

## **A.3/ Mixed Decisions**

### **A.3.1/ Mission or Production Planning Jointly with Maintenance**

Tables A.13 and A.14 summarize the papers that optimized the mission or production scheduling jointly with maintenance. One can note the big variety in the degradation models used for these works. Some works did not focus on the degradation model of the system or the components and just assumed that each job has a failure probability related to the system health. Most likely these failure probabilities are not explained and the authors did not show how they obtained such information.

### **A.3.2/ Logistics Jointly with Maintenance**

Table A.15 summarizes the works that optimized the logistics movements (orders of spare parts) and maintenance planning. First, almost all the works that dealt with spare part ordering and maintenance planning used stochastic process for the degradation model. Works in the aerospace domain used the Wiener process. The objectives of these works are always cost-oriented, in which either the cost are to be minimized or revenues and benefits to be maximized. As for the method used to solve the optimization problem, mostly each of these works developed its own heuristics with the exception of two that used genetic algorithms with Monte Carlo simulations. All these works used the prognostic information in the same way, i.e., to define the maintenance and the spare part ordering dates. Actually, the works of each domain are quite similar, and the only difference is the proposed heuristics. Sadly, neither of these works proposes a comparison between the results they obtained and the results obtained by other authors.

Table A.11: Existing papers for automatic control. (Part 1)

Reference	Controller Type	Degradation Model	Objective	Used Method	Integration of RUL	Application
[15]	Linear-Quadratic Regulator	Not Mentioned	Maximize Performance	Power Analysis	As A Constraint	Servo Motor
[20]	Model Predictive Control	Physic Based	Maximize Reliability and Survivability under Fault/Failure Conditions	Analytical Process (LaGrangian)	As A Constraint	Electro-Mechanical Actuators Motor
[21]	Model Predictive Control	Physic Based	Maximize Component Lifetime	Analytical Process	As An Objective + Change The Control Parameters	Electro-Mechanical Actuators Motor
[16]	Fault-Adaptive Control	Physic Based	Minimize the Derived Risk	Analytical Process	As A Constraint + In Cost Function	Unmanned Ground Vehicle Motor
[148]	Model Predictive Control	Not Mentioned	Distribute the Control Effort to Extend the Systems Lifetime	Analytical Process	As A Constraint + As A Control Variable	Pumps
[104]	Linear-Quadratic Regulator	Gamma Process	Maximize the Horizon with the best Performance	RUL-based LQR Algorithm	To Change the LQR Parameters	DC-Motor



Table A.12: Existing papers for automatic control. (Part 2)

Reference	Controller Type	Degradation Model	Objective	Used Method	Integration of RUL	Application
[139]	Proportional-Integral-Derivative	Stochastic Process	Maximize Component Lifetime	Generic Particle Filter	As A Constraint + Defines The Control Set-point	Pump
[178]	Model Predictive Control	Physic Based	Maximize Component Lifetime	Mixed Integer Quadratic Program	As A Constraint + Define The Control Set-point	Double-Integrator Plant Model
[102]	Linear-Quadratic Regulator	Stochastic Process	Balance the Energies of the Tracking Error and of the Control Set-point	Analytical Process	As A Constraint + Define the Regulators Parameters	DC Motor with Ball Bearing
[74]	Model Predictive Control	Stochastic Process	Maximize the Economic Performance	Chance-Constraint Programming	As A Constraint + Defines the Controller Parameters	Water Distribution Network

Table A.13: Existing papers for jointly production/mission and maintenance scheduling. (Part 1)

Reference	Multi-Machines	Degradation Model	Objective	Used Method	Integration of RUL	Application Domain
[39]	✓	Not Mentioned	Minimize Mission Time + Minimize Maintenance Cost	Receding Horizon Task Assignment + Maintenance Grouping Algorithm	Classification of UAVs	Aerospace
[81]	✓	Deterministic Model	Minimize the Waste of Useful Life	Linear Program	Task Severity + In Cost Function	Railways
[142]		Physic Based	Maximize the Service Life + Maximize the Revenue	Not Mentioned	In Cost Function	Wind Turbines
[145]		Not Mentioned	Minimize the Total Tardiness	Developed Algorithm	As A Constraint	Manufacturing
[186]		Gamma Process	Maximizing the Long-Term Expected Profit per Unit Time	Developed Algorithm	Maintenance Dates	Manufacturing
[59]	✓	Deterministic Model	Minimize Makespan + Minimize Maintenance Cost	Developed Heuristic	Task Severity + Maintenance Dates	Manufacturing
[100]	✓	Deterministic Model	Minimize the Total Maintenance Cost	Mixed-Integer Linear Programming and Pro-Genetic Algorithms	In Cost Function + As A Constraint	Manufacturing

Table A.14: Existing papers for jointly production/mission and maintenance scheduling. (Part 2)

Reference	Multi-Machines	Degradation Model	Objective	Used Method	Integration of RUL	Application Domain
[99]	✓	Fuzzy Logic	Minimize Makespan + Minimize Maintenance Cost	Variable Neighborhood Search Heuristics	Task Severity + Maintenance Dates	Manufacturing
[42]	✓	Not Mentioned	Not An Optimization Problem	-	Asses Systems Capacity to Achieve a Task	Manufacturing
[32]		Gamma Process	Minimize Total Cost per Unit Time	Periodic CBM	Maintenance Dates	Manufacturing
[13]	✓	Not Mentioned	Minimize Total Cost	Multi-Agent	Selection of the Suitable Task + Assess Systems Capacity to Achieve a Task + Maintenance Dates	Manufacturing
[120]		Weibull Distribution	Minimize the Total Cost	Genetic algorithm	In Cost Function + As A Constraint	Manufacturing

Table A.15: Existing papers for jointly spare parts ordering and maintenance scheduling.

Reference	Multi-Machines	Degradation Model	Objective	Used Method	Integration of RUL	Application Domain
[189]	✓	Wiener Process	Minimize Expected Cost	Developed Heuristic	Maintenance and Ordering Dates	Aerospace
[190]		Wiener Process	Minimize Expected Cost	Developed Heuristic	Maintenance and Ordering Dates	Aerospace
[182]		Monotonically Increasing Stochastic Process	Minimize the Expected Replacement Cost	Developed Heuristic	Selection of Components to Maintain + Maintenance and Ordering Dates	Aerospace
[26]	✓	Wiener Process	Minimize the Total Cost	Genetic Algorithm Combined with Monte Carlo Simulations	Maintenance and Ordering Dates	Aerospace
[25]	✓	Wiener Process	Minimize the Total Cost	Genetic Algorithm Combined with Monte Carlo Simulations	Inspection, Maintenance and Ordering Dates	Aerospace
[31]	-	Not Mentioned	Minimize the Expected Long-Run Total Cost	Developed Heuristic	Maintenance and Ordering Dates	Manufacturing
[17]	✓	Not Mentioned	Minimize Long Term Maintenance and Inventory Costs	Periodic CBM	Maintenance Dates + Ordering Dates	Manufacturing
[120]		Poisson Process	Maximize Net Revenue	Markov Decision Process Based Heuristics	Maintenance Dates + Ordering Dates	Manufacturing
[133]		Stochastic Process	Minimize Long Run Expected Cost per Time Unit	Developed Heuristic	Warning Generation + Maintenance Dates	Wind Turbines

## Appendix B

# Manufacturing Study Case Results Tables

### B.1/ Single Speed

Table B.1: Results of the single-speed tests

		Machine	GA-CBM	PHM+	Difference
Small Tasks	Single Product	M1	27.1	35.9	8.8
		M2	27.7	37.2	9.5
		M3	12.1	28.3	16.2
		M4	31.8	34.8	3
	Multiple Products	M1	18.6	31.1	12.5
		M2	24.5	32.6	8.1
		M3	10	25.2	15.2
		M4	28	30.8	2.8
Medium Tasks	Single Product	M1	27.8	32.4	4.6
		M2	28.2	30.6	2.4
		M3	14.6	25.5	10.9
		M4	31.5	28.9	-2.6
	Multiple Products	M1	20.6	27.4	6.8
		M2	26	28.4	2.4
		M3	9.8	22.8	13
		M4	28.7	25	-3.7
Large Tasks	Single Product	M1	27.3	26	-1.3
		M2	27.1	26.1	-1
		M3	14.8	20.3	5.5
		M4	30.9	23.7	-7.2
	Multiple Products	M1	23.2	21.5	-1.7
		M2	26.8	22.5	-4.3
		M3	9.7	17.5	7.8
		M4	30.3	19.3	-11

## B.2/ Multiple Speeds

Table B.2: Results of tests on multiple profiles

		Machine	CBM-GA	PHM+	Difference
Small Tasks	Single Product	M1	27.1	40.5	13.4
		M2	27.7	40.9	13.2
		M3	12.1	26.1	14
		M4	31.8	45.4	13.6
	Multiple Products	M1	18.6	33.5	14.9
		M2	24.5	36.2	11.7
		M3	10	26.9	16.9
		M4	28	40.5	12.5
Medium Tasks	Single Product	M1	27.8	38.6	10.8
		M2	28.2	41	12.8
		M3	14.6	30.5	15.9
		M4	31.5	44.9	13.4
	Multiple Products	M1	20.6	33.1	12.5
		M2	26	35.7	9.7
		M3	9.8	27.2	17.4
		M4	28.7	40	11.3
Large Tasks	Single Product	M1	27.3	35.5	8.2
		M2	27.1	38.2	11.1
		M3	14.8	28.2	13.4
		M4	30.9	43.1	12.2
	Multiple Products	M1	23.2	31.2	8
		M2	26.8	33.7	6.9
		M3	9.7	25.3	15.6
		M4	30.3	38.7	8.4



