

ABSTRACT

Title of Thesis: **A Genetic Algorithm and Monte Carlo
Simulation-Based Framework for Maintenance
Policy Optimization of Multi-Component Aircraft
Systems**

Name of Candidate: Matthew Collins, Master of Science, 2026

Thesis directed by: Professor Yunfei Zhao, Department of Mechanical
Engineering

This thesis presents a maintenance policy optimization framework for multi-component systems that are subject to stochastic degradation and imperfect maintenance. The objective of the thesis is to determine the optimal inspection interval as well as the repair and replacement maintenance thresholds that minimize the expected long-run cost of operation. The optimization framework considers component degradation, maintenance actions, and system reliability requirements.

The multi-component system is modeled as a series-parallel architecture that consists of multiple subsystems with k-out-of-n redundancy structures. Component degradation is represented using a few different degradation models to reflect different physical failure mechanisms. Examples include Paris' Law for fatigue crack growth and Wiener process models for wear accumulation. The system components are inspected periodically, and maintenance decisions are determined using condition-based thresholds that define repair and replacement actions. Repairs are assumed to be imperfect, restoring components to a condition that is better than old but worse than new.

A Monte Carlo simulation framework was developed to evaluate maintenance policies

under uncertainty. The simulation tracks component degradation, maintenance events, corrective actions, and system failures over a specified operating horizon. A genetic algorithm was also developed to optimize the inspection interval and component-level maintenance thresholds. The overall objective function aims to minimize the expected cost per unit time, including inspection, repair, replacement, corrective maintenance, and system failure costs.

The proposed framework is ultimately demonstrated through two aircraft-inspired case studies consisting of a turbofan engine, hydraulic pump, electrical generator, and actuator subsystem. In the simplified single-component-per-subsystem case study, the optimized policy resulted in an inspection interval of 211.9 hours and an expected operating cost of \$303.79 per hour. For the complex k -of- n system case study, the optimized inspection interval decreased to 146.5 hours with an expected operating cost of \$494.76 per hour. In addition, a sensitivity analysis was conducted to demonstrate how the optimized policy changes when the cost parameters are individually increased and decreased by $\pm 5\%$. The final results illustrate the ability of the optimization framework to identify cost-effective maintenance policies while accounting for stochastic degradation behavior and system-level reliability constraints. The methodology provides a flexible approach for condition-based maintenance optimization that can be extended to a wide range of engineering systems.

**A Genetic Algorithm and Monte Carlo Simulation-Based Framework for
Maintenance Policy Optimization of Multi-Component Aircraft Systems**

by

Matthew Collins

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science

2026

Advisory Committee:

Professor Yunfei Zhao, Chair

Professor Katrina Groth

Professor Mohammad Modarres

© Copyright by
Matthew Collins
2026

Contents

Nomenclature	vii
1 Introduction	1
1.1 Background	1
1.2 Maintenance Challenges	2
1.3 Research Objectives	2
1.4 Thesis Organization	3
2 Literature Review	4
2.1 Single Component	5
2.2 Multi-Component Systems	7
2.3 Discrete Degradation State	8
2.4 Continuous Degradation States	10
2.5 Threshold Based Maintenance	12
2.6 Genetic Algorithms for Maintenance Optimization	14
3 Problem Statement	17
3.1 System Description and Assumptions	17
3.2 Component Degradation Models	18
3.2.1 Paris' Law	19
3.2.2 Wiener Process Degradation Models	20

3.2.3	Rotating Part Degradation	21
3.3	Maintenance Policy and Decision Variables	23
3.4	Cost Structure	25
3.5	Objective Function	26
4	Methodology	28
4.1	Genetic Algorithm Framework	28
4.1.1	Selection	30
4.1.2	Crossover	31
4.1.3	Mutation	33
4.1.4	Termination Criterion	34
4.2	Monte Carlo Simulation Procedure	34
4.2.1	Simulation Initialization	35
4.2.2	Time-Stepping Loop	35
5	Case Studies	39
5.1	Case Study 1	39
5.1.1	Component-Specific Degradation Models and Parameters	41
5.1.2	Results and discussion	45
5.2	Case Study 2	51
5.2.1	Results and Discussion	52
5.3	Discussion of Research Objective and Model Challenges	56
5.3.1	Corrective-only Baseline Comparison	56
5.3.2	Evaluation of Research Objectives	57
6	Conclusion and Future Work	60

List of Tables

1	Abbreviations and technical terminology used in this thesis.	vii
2	Mathematical variables used in the maintenance optimization framework. . .	xi
5.1	Paris' Law crack growth model parameters for the engine.	41
5.2	Hydraulic component Wiener process degradation parameters.	42
5.3	Lag-based actuator degradation model parameters.	43
5.4	Rotating part wear accumulation parameters for the generator UOA model. .	43
5.5	Cost and imperfect repair parameters used in the simulation.	44
5.6	Genetic algorithm parameters used for policy optimization.	44
5.7	Optimal maintenance policy obtained from the genetic algorithm.	45
5.8	Optimal maintenance policy obtained from the genetic algorithm for the ex- panded (k_i -out-of- n_i) case study.	52
5.9	Comparison of the optimized maintenance policy with the corrective-only baseline for Case Study 1.	57
5.10	Evaluation of research objectives and model capabilities.	58

List of Figures

3.1	Multi-component system RBD diagram.	18
4.1	Genetic Algorithm Framework	29
4.2	Selection	30
4.3	Crossover	32
4.4	Mutation	33
4.5	Monte Carlo simulation framework used to evaluate candidate maintenance policies.	36
5.1	Example aircraft-inspired system consisting of four single-component subsystems connected in series. Each subsystem follows a 1-out-of-1 structure, meaning failure of any component causes failure of the overall system.	40
5.2	Average Fitness convergence curve showing the evolution of the average fitness values across generations.	46
5.3	Best Policy Fitness convergence curve showing the evolution of the best fitness values across generations.	46
5.4	Cost distribution of total cost across 1000 Monte Carlo simulation runs. . . .	47
5.5	Sensitivity analysis results for Case Study 1.	49
5.6	Aircraft-inspired complex (k_i -out-of- n_i) system	52
5.7	Average Fitness convergence curve showing the evolution of the average fitness values across generations.	54

5.8	Best Policy Fitness convergence curve showing the evolution of the best fitness values across generations.	54
5.9	Cost distribution of total cost across 1000 Monte Carlo simulation runs. . . .	55

Nomenclature

This section defines the abbreviations, genetic algorithm terminology, and mathematical variables used throughout this thesis.

Abbreviations and Technical Terminology

Table 1: Abbreviations and technical terminology used in this thesis.

Abbreviation or Term	Definition
CBM	Condition-based maintenance; a maintenance strategy in which maintenance decisions are based on the observed condition or degradation state of a component.
CM	Corrective maintenance; an unplanned maintenance action performed after a component has reached its critical failure threshold.
DEAP	Distributed Evolutionary Algorithms in Python; the Python library used to implement the genetic algorithm.

Continued on next page

Table 1 – Continued from previous page

Abbreviation or Term	Definition
GA	Genetic algorithm; a population-based optimization method inspired by evolutionary selection.
MC	Monte Carlo simulation; repeated stochastic simulation used to estimate the expected performance of a maintenance policy.
PM	Preventive maintenance; a planned maintenance action performed before component failure.
RBD	Reliability block diagram; a graphical representation of the functional relationships and redundancy structure of a system.
RUL	Remaining useful life; the estimated operating time remaining before a component reaches its failure condition.
UOA	Used oil analysis; a condition-monitoring method that uses wear particles or contaminants in oil as indicators of component degradation.
Action	The maintenance decision applied to a component. The actions considered in this thesis are do nothing, repair, and replace.
Candidate policy	One possible combination of an inspection interval and subsystem repair and replacement thresholds.

Continued on next page

Table 1 – Continued from previous page

Abbreviation or Term	Definition
Chromosome	The encoded representation of one complete candidate maintenance policy in the genetic algorithm.
Convergence	Stabilization of the best observed fitness value according to the specified stopping criterion. Convergence does not guarantee that the global optimum has been identified.
Crossover	A genetic operator that combines genes from two parent policies to produce offspring policies.
Elitism	The preservation of the best-performing candidate policy from one generation to the next.
Feasible policy	A candidate policy that satisfies all decision-variable bounds and threshold-ordering constraints.
Fitness	The numerical measure used to evaluate a candidate policy. In this thesis, fitness is the estimated expected maintenance cost per operating hour, with lower values preferred.
Gene	One decision variable within a chromosome, such as the inspection interval or one maintenance threshold.
Generation	One complete iteration of policy evaluation, crossover, mutation, selection, and population replacement.

Continued on next page

Table 1 – Continued from previous page

Abbreviation or Term	Definition
Individual	The DEAP representation of one candidate maintenance policy. Individual and chromosome are used interchangeably in the genetic algorithm discussion.
Mutation	A genetic operator that randomly perturbs one or more policy variables to maintain population diversity and explore new regions of the search space.
Population	The collection of candidate maintenance policies evaluated during one generation of the genetic algorithm.
Selection	The process used to choose candidate policies for reproduction based on their fitness values.
Tournament selection	A selection method that randomly samples a group of candidate policies and selects the policy with the best fitness from that group.
Uniform crossover	A crossover method in which each gene is independently exchanged between two parent policies with a specified probability.

Mathematical Variables

Table 2: Mathematical variables used in the maintenance optimization framework.

Symbol	Definition	Units
i	Subsystem index, where $i = 1, \dots, s$.	—
j	Component index within subsystem i , where $j = 1, \dots, n_i$.	—
s	Total number of subsystems in the multi-component system.	—
n_i	Number of components contained in subsystem i .	—
k_i	Minimum number of functioning components required for subsystem i to remain operational.	—
t	Operating time.	h
Δt	Discrete simulation time-step duration.	h
H	Total operating horizon of one Monte Carlo replication.	h
N	Number of Monte Carlo replications used to evaluate a maintenance policy.	—
$x_{i,j}(t)$	Degradation state of component j in subsystem i at time t . Its physical meaning depends on the component degradation model.	Model-dependent

Continued on next page

Table 2 – Continued from previous page

Symbol	Definition	Units
$x_{i,j}(0)$	Initial degradation state of component j in subsystem i .	Model-dependent
$x_{i,\text{crit}}$	Critical degradation threshold at which a component in subsystem i is considered failed.	Model-dependent
$A_{i,j}(t)$	Maintenance action applied to component j in subsystem i at time t .	–
T_{sys}	Common scheduled inspection interval for the multi-component system.	h
$\tau_{i,1}$	Repair threshold for components in subsystem i .	Model-dependent
$\tau_{i,2}$	Replacement threshold for components in subsystem i .	Model-dependent
p_i	Imperfect-repair factor for subsystem i . A smaller value represents a greater reduction in degradation following repair.	–
θ	Complete maintenance-policy vector containing the inspection interval and subsystem repair and replacement thresholds.	–
$J(\theta)$	Expected long-run cost per operating hour associated with policy θ .	\$/h

Continued on next page

Table 2 – Continued from previous page

Symbol	Definition	Units
$G(\theta)$	Total cost accumulated under policy θ during one simulated operating horizon.	\$
$G_n(\theta)$	Total cost accumulated during Monte Carlo replication n under policy θ .	\$
N_{insp}	Number of scheduled system inspections during the operating horizon.	—
$N_{\text{rpr},i,j}$	Number of preventive repair actions performed on component j in subsystem i .	—
$N_{\text{repl},i,j}$	Number of preventive replacement actions performed on component j in subsystem i .	—
$N_{\text{corr},i,j}$	Number of corrective-maintenance actions performed on component j in subsystem i .	—
N_{sysfail}	Number of system-level failures during the operating horizon.	—
C_{insp}	Cost of one scheduled system inspection.	\$
$C_{\text{rpr},i}$	Cost of one preventive repair for a component in subsystem i .	\$
$C_{\text{repl},i}$	Cost of one preventive replacement for a component in subsystem i .	\$

Continued on next page

Table 2 – Continued from previous page

Symbol	Definition	Units
$C_{\text{corr},i}$	Cost of one corrective-maintenance action for a component in subsystem i .	\$
C_{sysfail}	Penalty cost assigned to a system-level failure.	\$
C	Paris' Law crack-growth coefficient.	Model-dependent
m	Paris' Law crack-growth exponent.	–
ΔK	Stress-intensity-factor range used in the Paris' Law degradation model.	MPa $\sqrt{\text{m}}$
ω	Normally distributed random variable representing uncertainty in the Paris' Law crack-growth rate.	–
σ_ω	Standard deviation associated with the random Paris' Law crack-growth term.	–
a_0	Initial crack length of the modeled engine component.	mm
a_{crit}	Critical crack length at which the modeled engine component is considered failed.	mm
λ	Mean degradation drift rate in a Wiener-process degradation model.	Model-dependent
σ_λ	Standard deviation of the hydraulic-component degradation drift-rate distribution.	h^{-1}

Continued on next page

Table 2 – Continued from previous page

Symbol	Definition	Units
σ	Diffusion coefficient representing stochastic variability in a Wiener-process degradation model.	Model-dependent
$Z(t)$	Independent standard Gaussian random variable used in a stochastic degradation update, where $Z(t) \sim \mathcal{N}(0, 1)$.	–
$W(t)$	Wear-metal concentration used as the degradation state of the generator.	ppm
a	Initial wear-metal concentration in the generator degradation model.	ppm
b	Linear wear-accumulation coefficient in the generator degradation model.	ppm/h
c	Quadratic wear-accumulation coefficient in the generator degradation model.	ppm/h ²
σ_x	Diffusion coefficient representing uncertainty in generator wear-metal accumulation.	Model-dependent
W_{crit}	Critical generator wear-metal concentration.	ppm
P	Genetic algorithm population size.	–
k	Tournament size used during tournament selection. This use of k is distinct from the subsystem reliability parameter k_i .	–

Continued on next page

Table 2 – Continued from previous page

Symbol	Definition	Units
p_c	Probability that crossover is applied to a selected pair of parent policies.	–
p_m	Probability that mutation is applied to an offspring policy.	–
p_g	Independent probability that an individual gene is modified during mutation.	–
N_{gen}	Maximum permitted number of genetic algorithm generations.	–
N_{patience}	Number of consecutive generations without sufficient improvement allowed before early termination.	–
δ_{min}	Minimum reduction in fitness required to count as an improvement for the early-stopping criterion.	\$/h

Chapter 1

Introduction

1.1 Background

Multi-component systems are essential to many industries such as aviation, nuclear power, and manufacturing because they form the machinery required to perform important tasks. These systems are often made up of a variety of components that may fail under various failure mechanisms depending on the component's material or the type of stress it undergoes on a regular basis. Since these systems have a variety of parts that fail under various forms of degradation, maintenance must be performed in order to renew the system and allow it to keep running at a sufficient rate. In terms of economic outcome, maintenance is key for minimizing downtime while maximizing reliability. For some time, many logistics teams have been tasked with answering the question of how should maintenance be performed in order to minimize the cost. As a result, developing maintenance policies that effectively balance maintenance cost and system reliability remains an important challenge for multi-component systems.

1.2 Maintenance Challenges

A common challenge in maintaining a multi-component system is that components often undergo stochastic degradation meaning that over time as a component deteriorates it is inherently random and unpredictable. Maintenance actions performed on one component may not be required for another component, making it difficult to determine inspection schedules and maintenance thresholds that minimize overall system cost. In addition, the components that make up a multi-component system are being used simultaneously in k of n redundancy systems. These components have to work together to make a system properly run and the failure of one component can affect overall system performance and may contribute to system-level failures depending on the system architecture. Modern industries can often use sensors to determine the age of a given component, however these sensors can cost wide sums of money and not every industry can afford to pay for equipment such as those. By not paying for the equipment, inspections must take place in order to get a proper reading of the health of each component to determine if maintenance is needed. In addition, it is important to differentiate when a component should be repaired and when a component should be replaced. For each component, there must exist degradation thresholds in which it is more economically beneficial to repair or replace a component. Also, for a multi-component system there must be an inspection interval that is economically beneficial in catching degraded components before they fail, while also not occurring too frequently to the point where they accumulate too much of an economic cost. Understanding the k of n reliability structures of a multi-component system is also important for determining the duration in which a system can run with failed components before the system itself fails.

1.3 Research Objectives

This thesis breaks down a multi-component system to determine what sort of possible maintenance actions should be taken, that is, a maintenance policy. To explore variations of

different policies, evaluate them against each other, and identify the optimal policy, a genetic algorithm will be used. It is an evolutionary based optimization tool that has the ability to generate policies and utilize a fitness function to compare the policies and determine which policy has a minimized cost. Monte Carlo simulation will also be used to model the degradation of the components while also accumulating the cost of a given policy. This thesis develops a simulation-based optimization framework that combines degradation modeling, maintenance decision thresholds, Monte Carlo simulation, and Genetic Algorithm optimization to identify near-optimal maintenance policies for multi-component systems.

1.4 Thesis Organization

This thesis will cover an extensive literature review that will show strategies other researchers have used to optimize maintenance policies. This includes policies for single component and multi-component systems as well as studies that focus on discrete degradation states and studies that focus on continuous degradation states. The literature review also shows how other studies have handled degradation thresholds in regards to decision making and maintenance actions. Lastly, there will be a literature review on how other researchers have used genetic algorithms to optimize their own maintenance policy. Next, this thesis will present the problem statement in which the various elements of the problem will be broken down from definitions, to degradation models, and to cost models. After that, a systematic methodology will be discussed for how the genetic algorithm performs its search for a near-optimal solution. The next chapter will present a case study followed by the results and a discussion. Lastly, the conclusions of the thesis as well as possible future work will be discussed.

Chapter 2

Literature Review

Optimized maintenance planning has been a central challenge in modern aviation systems. Increasing operational complexity, stringent safety requirements, and rising economic pressures require decision frameworks that are cost effective while maintaining system reliability. Aircraft components are subject to diverse failure mechanisms which include gradual degradation, random failures, and externally induced shocks. In addition, maintenance actions interact with operational schedules, resource constraints, and long-term system performance. Addressing these challenges requires an integrated understanding of how degradation is modeled, how reliability information gets generated and updated, and how maintenance decisions are optimized under events of uncertainty. In addition, a broad body of literature has emerged, spanning operational maintenance routing, reliability estimation, health management architectures, cost modeling, finite-horizon optimization, learning-based prognostics, and component-specific degradation modeling. This chapter reviews these contributions, organizing the literature into similar categories that progressively add together from operational planning and reliability modeling toward health-aware, sequential maintenance decision frameworks. By critically examining the assumptions, methodologies, and limitations of existing approaches, this literature review will be able to establish the foundation and research gap that motivate the development of the stochastic, system degradation-driven

maintenance optimization framework proposed in this thesis.

2.1 Single Component

Initial research began with maintenance optimization for single components where the core logic of the studies was to establish a threshold for maintenance. Single components were initially researched because they are mathematically simpler. They are good for establishing inspection interval optimization as well as repair and replacement thresholds. Overall, the primary objective is typically to minimize expected maintenance cost while maintaining an acceptable level of reliability and availability.

A big factor in maintenance is to decide when a part needs to be fixed and how many spare parts are needed for a component. [19] aimed to optimize the threshold for repair maintenance on a part and how many spares are needed. In order to do this, they use the gamma process to model wear and account for imperfect maintenance. The gamma process is used because it is commonly used for wear, corrosion, erosion, and cumulative degradation. It then uses an inventory policy in which when a part's wear reaches a certain level, the system checks a warehouse. If there are no spare parts, it calculates if it is cheaper to order a fast shipping part or allow the part to degrade until failure. In addition, they calculate maintenance as imperfect, meaning that repair does not make the part in good-as-new condition which means that the component will wear faster each time the component is repaired.

[17] aimed to minimize the long term operational costs by finding a degradation threshold to conduct preventative maintenance. The methodology used in their research was to use a stochastic process in which the equipment's health was tracked by a single numerical diagnostic variable. In order to do this, they calculate the moment when the health value crosses over the safety limit and use probability density functions to predict when the limit crossing occurs. They aim to find a preventive maintenance threshold that is set at a slightly

lower value than the failure threshold so that they can conduct maintenance before the failure occurs.

[16] aimed to determine the best time intervals for maintenance tasks in order to reduce the cost of aircraft maintenance. They did this by using Erlang and exponential distributions in order to map out how a system moves through different stages of aging before failure occurs. In order to do this, they treat the aircraft as a system that moves through a set of states. The Erlang distribution examines a component such as the engine and calculates the amount of time that it takes to go from the first stage of new condition, to the second stage of being worn, to the third stage of being in critical condition. The model uses hazard rates and reliability functions rather than focusing on the physical wear of the components. The methodology used in this thesis replaces the generic probability curves with Monte Carlo simulations of physics-based models.

[3] evaluated maintenance performance between cost and system availability for systems that degrade over time and face two types of failure models at the same time. Their methodology was to model the two different failures simultaneously. First was the internal degradation using a gamma stochastic process which represents monotonic wear. In addition, the observed component underwent random external shocks that the component became increasingly vulnerable as it degraded over time. They modeled the second failure by using the Poisson process. The maintenance policy used periodic inspections where the health was compared to two specific levels which were a preventative threshold and a corrective threshold.

Overall, these studies focused on single components rather than multi-component systems. In many industries, maintenance teams have to review multi-component systems by studying their redundancy structures, competing maintenance priorities, and system level dependencies.

2.2 Multi-Component Systems

Real-world systems are often multi-component systems. These are important to study because maintenance decisions should ideally be made at the system level since there is dependency between components. A major challenge that arises when logistically planning maintenance for multi-component systems is that maintenance actions and inspection scheduling should be coordinated for all components while balancing system reliability with operational cost. This also typically increases the computational complexity. As a result, researchers have explored heuristic optimization, simulation-based optimization, selective maintenance strategies, and degradation aware maintenance scheduling.

[21] aimed to evaluate the reliability of the entire aircraft electrical grid while also accounting for the unknowns in how different parts interact with each other. Their study was very useful with providing the physics-based methods to calculate the reliability and parameters of the electrical generators. The methodology treated component failure rates as uncertain variables represented through probability distributions rather than fixed constant values and propagated this uncertainty through the system structure to evaluate how uncertainty in component failures affected the overall aircraft power supply reliability. To do this, they used the exponential function with various hazard rates in order to model the various types of electrical equipment on the aircraft. After, they displayed the relationship of the generators in a k of n redundancy system to calculate the system reliability.

Part of the maintenance cost optimization process has to deal with the order in which components receive maintenance. [13] proposed an opportunistic grouping strategy in their research. Their objective was to reduce the total maintenance cost and unnecessary maintenance interventions in multi-component systems by optimizing the timing and grouping of maintenance activities. To do this, they used remaining useful life (RUL) predictions to identify windows of opportunity. For example, if a component's RUL was low enough, it would be serviced together with neighboring components in order to save setup costs and labor instead of waiting longer for the component to reach a maintenance threshold. Their end goal

was to minimize the total number of maintenance interventions by grouping tasks together. This inspired the methodology used in this thesis because rather than waiting to maintain a failed component, this article aims to optimize maintenance timing using predicted RUL information in order to reduce overall maintenance cost. Similarly, the methodology in this thesis promotes maintenance policies that use optimal maintenance thresholds to reduce the overall operational cost.

Sometimes, the maintenance actions can undergo task duration uncertainty that could impact the maintenance cost or the maintenance schedule. [18] researched how to ensure that enough planes are available for missions even when repairs take longer than expected or if technicians are busy. The methodology used in their research was to use mixed-integer linear programming combined with robust optimization. Since the airlines know that maintenance takes a random amount of time, they pad the schedule with extra time based on a conservative parameter which controls how much variability in maintenance duration the schedule is protected against. The authors then used Monte Carlo simulation to sample uncertain maintenance durations in order to test if the workforce schedule still satisfied the fleet availability constraints.

2.3 Discrete Degradation State

Existing research has used both discrete states and continuous states to model component degradation. This section focuses on studies that used discrete degradation models, and the next section will focus on studies that used continuous degradation models.

[16] used discrete variables to describe the aging states as components transition from new, to worn, to critical, and to failed. In his methodology, Okoro used the Erlang distribution to estimate the probability of transitioning from one discrete condition state to another and maintenance actions were triggered when the system entered a specific degradation state.

[13] evaluated components using remaining useful life categories in order to determine maintenance timing. In this methodology, opportunistic maintenance was triggered when components entered critical maintenance windows. The actions were to maintain immediately, schedule maintenance with another task, or delay until further notice.

Outside of the aviation industry, [9] researched the maintenance decision making problem for machines that are constantly deteriorating. Their methodology was to use Deep Q-Learning in which they framed the machine's life as a series of states and every hour, the agent would read the health state and choose an action. A reward function was used to support keeping the machine running while giving penalties when there was a maintenance cost, failure cost, or downtime which loses money. The Deep Q-Network worked to simulate the machine thousands of times and learn to value preventative maintenance until it eventually learned maintenance policies that balanced maintenance cost, production loss, and machine reliability. Their research holds a shared goal with the methodology used in this thesis to minimize a complex cost function that includes repairs, failures, and downtime.

[22] developed a condition-based maintenance optimization framework for a K -out-of- N :G multi-component system under periodic inspection. Their model considers economic dependency, stochastic dependency, and imperfect maintenance. The objective of the study was to minimize the expected long-run discounted cost. Component degradation was modeled using heterogeneous stochastic processes, including Wiener, Gamma, and inverse Gaussian processes, while stochastic dependence among components was represented using copula functions. Imperfect maintenance was modeled as a reduction in the degradation level rather than a full restoration to a good-as-new condition. To solve the optimization problem, the continuous degradation state space was discretized and a Markov decision process was solved using value iteration with Monte Carlo simulation to obtain the optimal inspection interval and condition-based maintenance policy.

2.4 Continuous Degradation States

For maintenance optimization problems, continuous degradation states can be used for degradation models to represent component wear as a continuous process over time. Certain methodologies mostly use continuous degradation states when degradation accumulates gradually. Some examples of continuous state frameworks that are useful for modeling are crack growth, fatigue, and stochastic wear accumulation. Continuous degradation can be useful when determining when a component approaches a critical failure threshold. The methodology used in this thesis combines continuous degradation with discrete maintenance decisions in order to optimize maintenance cost and system reliability.

As discussed in the previous sections, [19] used a gamma stochastic process to represent wear accumulation over time. The component health was represented as a continuous degradation variable and maintenance thresholds were triggered when degradation exceeded a defined health level. Overall, their research demonstrated how continuous degradation trajectories could be integrated into decision frameworks.

[3] also used continuous degradation models through a gamma stochastic process to represent internal degradation. The methodology used here combined continuous degradation with stochastic external shock events which were modeled using the Poisson process and became increasingly likely as the degradation accumulated. Preventive and corrective maintenance action were triggered when the degradation state crossed specific maintenance thresholds as well. Overall, this method supported the use of continuous degradation in maintenance optimization to model how stochastic wear accumulation can be integrated into cost and availability analysis.

[17] aimed to minimize the long term operational costs by calculating an optimal degradation threshold for preventative maintenance. The methodology used in this article was to use a stochastic process in which the equipment's health was tracked by a single numerical diagnostic variable. In order to do this, they calculate the moment when the health value crosses the safety limit. They aim to find a preventive maintenance threshold that is set at a

slightly lower value than the failure threshold so that they can conduct maintenance before the failure occurs.

[7] discuss the realistic physics based modeling of a turbofan engine in their article in which their objective was to create a 3D simulation of how fatigue cracks initiate and grow in turbofan compressor disks. The methodology used was a finite element method for global stress and a dual boundary element method for local crack simulation. They used the effective stress intensity factor that's in Paris' law in order to track the growth rate of the crack. In the end, they were able to use their research to develop key parameters for the ATA 23 turbofan engine which became the foundation of the physics based simulation used in this thesis for the turbofan engine.

The main document used to extract simulation data for the hydraulic pumps in the genetic algorithm (GA) for this thesis was written by [12]. In this article, they aim to model the reliability of hydraulic pumps by using a term known as soft failure for when the performance drops too low rather than hard failure for when it explodes. The methodology used here was the Wiener process with drift which models the pump's volumetric efficiency as a stochastic process that shows consistent degradation, but also has some stochastic variability. Overall, this article backed the claim that the Wiener process with drift should be used to model the degradation of the hydraulic pump and the parameters from this article were utilized in the GA framework.

For the turbofan engine, Nascimento and Viana aimed to predict the degradation of an entire fleet of aircraft by combining physical laws with neural networks [15]. They did this by building a recurrent neural network (RNN) where the hidden state represented physical damage. They then embedded Paris' law by using the formula to govern how cracks grow over time and ensured that the model never predicted that the crack would heal itself. While the crack growth followed physics, they used the RNN to calculate stress intensity between short and long flights. Their work greatly supported the GA by showing how to properly model the degradation of a turbofan engine while also providing important parameters for

the turbofan engine model being researched. While this paper focused on prediction, the paper did not provide an optimization framework to decide when to schedule maintenance and minimize cost which is the focus of the GA in this thesis.

2.5 Threshold Based Maintenance

In maintenance optimization, it is important to determine the degradation thresholds in which maintenance actions should be conducted. In order to find these thresholds, the degradation of components must also be properly modeled. This section reviews the literature on degradation modeling and threshold based maintenance decision making.

[17] used a stochastic degradation variable to continuously track equipment health. Historical data was used to define critical thresholds. When degradation got near the critical threshold, maintenance decisions would be triggered. This methodology used a probability density function to estimate when degradation would reach the critical threshold and the preventive maintenance threshold was set at a lower degradation value in order to reduce the probability of failure. This methodology was important in showing how degradation monitoring could be integrated into condition-based maintenance decision making.

[19] addressed degradation modeling by using a gamma stochastic process to model continuous degradation over time. The component health deteriorated until the maintenance thresholds were reached. This model triggered maintenance actions to be conducted when an observed state was past the maintenance threshold rather than conducting maintenance at a set time interval. This paper also incorporated imperfect maintenance by allowing repaired components to degrade after maintenance. This paper evaluated different possible thresholds and how they impacted the long term maintenance cost.

[3] also used a gamma degradation process, but they introduced random shocks to represent unexpected failure by combining a Poisson shock process with continuous degradation. In this scenario, preventive and corrective maintenance decisions were triggered when the

degradation exceeded a predefined threshold. Periodic inspections were also used in order to monitor the degradation state of each component. Overall, the methodology evaluated the relationship between degradation accumulation, maintenance timing, and system availability. It also showed how stochastic degradation modeling could support maintenance optimization under uncertainty.

[13] used RUL calculations in order to determine when a component neared a critical maintenance threshold. In this problem, maintenance actions were triggered when the component degradation level reached a defined critical threshold level. The paper evaluated how degradation progression influenced maintenance scheduling decisions. Maintenance decisions were based on the degradation state of a component as well as the health condition of neighboring components. It showed how grouping maintenance for components could reduce downtime and overall maintenance cost.

[11] aimed to move away from point estimates and instead develop a probability distribution of RUL. Instead of generating a single predicted failure time, Lee calculated the cumulative probability of failure within future flight cycles. Monte Carlo dropout was then used to generate multiple stochastic RUL predictions that were combined to estimate the RUL probability distribution. In order to perform maintenance optimization, the probabilistic RUL distributions were input into a Deep Reinforcement Learning agent. The agent used the probabilistic RUL information to make maintenance decisions at fixed decision intervals over future flight cycles while balancing maintenance cost, fleet availability, and failure risk. The methodology also simulated aircraft operational cycles in order to evaluate how maintenance timing impacted long-term fleet availability. This article helped inspire this thesis to incorporate uncertainty into maintenance policy optimization.

2.6 Genetic Algorithms for Maintenance Optimization

Genetic algorithm relies on an evolutionary optimization process to generate many potential policies and evaluate them according to a fitness function. It can select the best policies and use crossover and mutation to further explore policies until a near-optimal solution is found. In multi-component optimization, the policy typically revolves around scheduling timing, maintenance actions, and degradation thresholds to perform the actions. This section discusses works that use genetic algorithms to minimize the maintenance cost for a system.

[14] aimed to determine the optimal degradation thresholds for condition based preventive maintenance in a multi-component deteriorating system. To do this, the methodology had to balance maintenance cost, system availability, and operational profit. To begin, the paper modeled a continuously degrading multi-component system that underwent stochastic degradation. A Markov degradation framework was implemented to represent the degradation levels between health states and maintenance actions were triggered when the degradation level reached a defined threshold. Monte Carlo simulation was also used to accumulate degradation as well as the maintenance outcomes. A genetic algorithm was then used to search for the optimal maintenance thresholds and policies to represent preventive maintenance and replacement maintenance. Their study used abstract degradation states to represent the health of the component meaning that the component would exist in a discrete degradation state and there would be probabilities of transitioning between states over time. In contrast, the methodology in this thesis used physics based models to accumulate degradation over time.

[8] aimed to minimize the long term maintenance and operational costs while also improving system reliability and availability through optimized maintenance thresholds. To do this, they modeled system degradation using discrete Markov health states and transitioned the components between degradation conditions over time. Every time a degradation threshold was passed, a maintenance action would be triggered. Monte Carlo simulation was used to evaluate stochastic degradation and maintenance outcomes which included pro-

duction interruptions. The optimization problem ultimately evaluated different maintenance threshold combinations in order to identify the most cost-effective policy. In this problem, a genetic algorithm was used to search for the optimal maintenance thresholds. Each chromosome represented a maintenance policy and the GA evaluated various policies using the simulated system degradation. The fitness function was based on cost, downtime, and system performance. In the end, the GA selected the degradation thresholds that resulted in a minimized maintenance cost. Their study represented degradation through abstract Markov health states rather than a physics-based degradation model.

[20] aimed to minimize maintenance and operational costs while maximizing system useful life and reliability. They modeled degradation using reliability aging equations and age-reduction factors in which maintenance partially restored the effective age of a component rather than returning it to a new condition. Various combinations of maintenance actions and time intervals were evaluated and the maintenance interactions of the multiple components were considered. In this scenario, the GA was used to optimize the maintenance schedule and actions. Each chromosome represented a preventive maintenance strategy consisting of maintenance timings and maintenance actions for individual system components, including no action, preventive maintenance, and replacement maintenance.

[4] aimed to develop a condition-based maintenance optimization framework for a degrading gas turbine nozzle system. Their research addresses the uncertainty in degradation parameters and maintenance outcomes while improving decision making. To do this, they used a multi-state degradation model where components transitioned between states probabilistically. Uncertainty was separated into aleatory uncertainty where there was natural stochastic degradation variability and epistemic uncertainty which was in the model parameters due to limited data. There were two main Monte Carlo simulation loops that were used. The outer loop sampled uncertain model parameters, and then the inner loop simulated stochastic degradation and maintenance outcomes. This problem used a genetic algorithm to optimize the policy. Each chromosome represented a different policy and the decision vari-

ables were the inspection settings, maintenance thresholds, and the condition-based policy parameters. Some limitations of this paper were that it only focused on a single component and that it focused on abstract multi-state degradation rather than a detailed physics-based degradation mechanism.

[23] aimed to develop a framework to minimize the airline maintenance cost by determining the optimal timing and location for maintenance activities. Their goal was to ensure that the flight schedule could be kept while minimizing operational disruptions. To do this, they built an optimization model that considered flight schedules, maintenance capacity, flight costs, opportunity costs, and maintenance costs. The maintenance timing had to be constrained with the allowable time windows. A GA was used to optimize the schedule and maintenance plans where each chromosome represented maintenance timing, the location of maintenance, and the type of maintenance action. A key limitation of their study was that it focused on the scheduling rather than the degradation of the aircrafts. No degradation modeling was used at all and the maintenance decisions were based on the schedule's convenience rather than degradation thresholds or the component health states.

[10] aimed to optimize aircraft assignment, scheduling, facility selection, and crew allocation while minimizing maintenance costs. To do this, the problem was formulated into four phases. The first phase was the assignment to airports, then the maintenance scheduling, the maintenance facilities, and last was the maintenance crew. Binary and integer decision variables for aircraft assignment and scheduling were used. In this scenario, each chromosome of the GA represented the flight assignment, maintenance schedule, facility, and crew assignment. Once again, their research focused primarily on the convenience of scheduling rather than degradation-based maintenance optimization. In addition, their research does not consider uncertainty while the problem in this thesis will.

Chapter 3

Problem Statement

3.1 System Description and Assumptions

A typical multi-component system could be defined by a system consisting of s subsystems connected in series. Each component degrades stochastically. The degradation state of component j in subsystem i at time t can be defined as $x_{i,j}(t)$ where $i = 1, \dots, s$ denotes the subsystem index and $j = 1, \dots, n_i$ denotes the component index within subsystem i . In any given scenario, if $x_{i,j} = 0$, then the component is considered to be in a good-as-new state. The component undergoes stochastic degradation until a critical degradation threshold is reached. It is assumed that at $t = 0$, $x_{i,j}(0) = 0$ for all components because all components will begin in a good-as-new condition.

Degradation models may be represented using either discrete or continuous state variables. In this thesis, the degradation state of each component is represented as a continuous variable. Time is discrete with equal time intervals, so maintenance actions, inspections, and degradation updates are evaluated at predetermined time intervals.

A component, j , in subsystem i is considered failed if the degradation state, $x_{i,j}$, reaches a critical degradation threshold which can be defined as $x_{i,\text{crit}}$. As a result, the component failure can be defined as: $x_{i,j}(t) \geq x_{i,\text{crit}}$ where component j in subsystem i fails at the time

t when the degradation state crosses the critical degradation threshold.

A multi-component system can typically have a k_i -of- n_i redundancy structure, where k_i is the minimum number of functioning components required for subsystem i to remain operational. In this scenario, these components work in parallel and the system may not reach critical failure if a singular component fails. Instead, a subsystem fails when fewer than k_i of its n_i components remain operational in order for that section in the series to fail and therefore lead to critical failure of the system. Figure 3.1 illustrates the generalized series-parallel system architecture considered in this thesis.

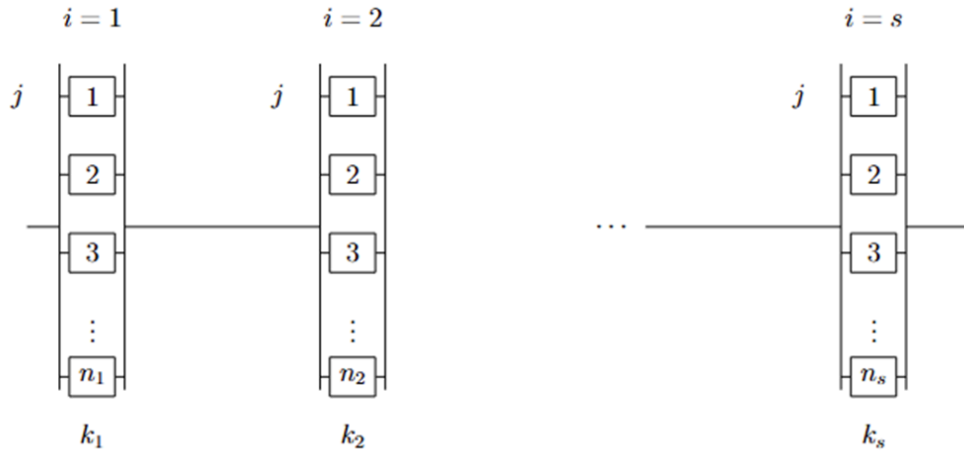


Figure 3.1: Multi-component system RBD diagram.

3.2 Component Degradation Models

Multi-component systems have components with different mechanical properties that can degrade as a result of various failure mechanisms. As a result, a single degradation model may not be able to model the degradation of all components in a system. In this thesis, component degradation is represented by stochastic models that capture the uncertainty associated with each degradation process. For this thesis, four different failure mechanisms are considered

and used to demonstrate the flexibility of the maintenance optimization framework.

3.2.1 Paris' Law

The first failure mechanism that this thesis will utilize is crack propagation caused by cyclic fatigue loading. When a crack propagates to a critical level, the component will fail. The crack growth is based on the stochastic Paris' Law formulation presented by [24]. The crack growth length is given by $x_{i,j}(t)$

$$\frac{dx_{i,j}(t)}{dt} = e^{\omega} C (\Delta K)^m, \quad (3.1)$$

where C and m are material-dependent Paris' Law constants and $\omega \sim \mathcal{N}(0, \sigma_{\omega}^2)$ represents uncertainty in the crack growth process.

Failure occurs when

$$x_{i,j}(t) \geq x_{i,\text{crit}}. \quad (3.2)$$

The variable $x_{i,j}(t)$ represents the crack length of component j in subsystem i at operating time t . The stress-intensity factor range, ΔK , characterizes the loading experienced at the crack tip and is a function of the crack geometry, applied stress range, and crack length. The parameter σ_{ω} represents the uncertainty associated with the crack growth process. The critical crack length, $x_{i,\text{crit}}$, represents the crack size at which the component is considered failed.

For the discrete-time simulation, the crack length is updated over each time step as:

$$x_{i,j}(t + \Delta t) = x_{i,j}(t) + e^{\omega} C (\Delta K)^m \Delta t \quad (3.3)$$

In this formulation, t represents operating time in the simulation, and the crack growth is expressed with respect to operating time so that the model is consistent with the other

degradation models used in this thesis.

3.2.2 Wiener Process Degradation Models

Many engineering systems experience gradual degradation that evolves continuously through time while being subject to uncertainty. Such degradation can be represented using a Wiener process with drift. A Wiener process consists of a deterministic drift term that captures the average degradation trend and a stochastic diffusion term that accounts for random variability. The general form of a Wiener process degradation model is [12]:

$$x_{i,j}(t + \Delta t) = x_{i,j}(t) + \lambda \Delta t + \sigma \sqrt{\Delta t} Z(t) \quad (3.4)$$

where λ is the drift parameter, σ is the diffusion coefficient, and $Z(t) \sim \mathcal{N}(0, 1)$ represents standard Gaussian noise. The Wiener process framework is used in this thesis to model hydraulic component degradation and actuator friction-growth degradation.

Hydraulic Component Degradation

Hydraulic components commonly experience degradation due to friction, contamination, internal leakage, and fatigue. These failure mechanisms often cause the component to undergo a gradual loss of performance over time. The degradation state $x_{i,j}(t)$ represents the accumulated degradation of component j in subsystem i . The Wiener process with drift from equation Equation (3.4) can model the degradation of hydraulic components. A component is considered failed when

$$x_{i,j}(t) \geq x_{i,\text{crit}}, \quad (3.5)$$

where $x_{i,\text{crit}}$ is the critical degradation threshold for component j in subsystem i .

Lag-Based Degradation Model

Some types of components such as actuators degrade through increasing friction, damping, or mechanical resistance. The degradation from these failure mechanisms causes a lagged response from the component. As a result, a model that measures the degradation should analyze the component's lag to determine the wear [2]. These degradation mechanisms cause an increasing actuator lag or tracking error that can be directly monitored during operation. Therefore, the actuator lag itself is selected as the degradation state in this thesis. To model the evolution of the degradation state of $x_{i,j}$, the Wiener process from Equation (3.4) can once again be used, where $x_{i,j}(t)$ represents the actuator lag or tracking error for actuator j in subsystem i at time t . The drift parameter λ represents the mean growth rate of actuator lag, the diffusion coefficient, σ represents uncertainty in actuator degradation, Δt represents the time step, and $Z(t) \sim \mathcal{N}(0, 1)$ represents standard Gaussian noise. The component can be considered failed when:

$$x_{i,j}(t) \geq x_{i,\text{crit}} \quad (3.6)$$

where $x_{i,\text{crit}}$ represents the critical actuator lag threshold at which the actuator is considered failed.

3.2.3 Rotating Part Degradation

Some components experience multiple forms of degradation in which moving parts cause material to break down. This can cause parts such as bearings and gears to not perform as well due to rotating machinery wear. In addition to this, some components such as generators require oil to run. As metallic parts break down, they can collect in the oil which could eventually cause the component to fail. As a result, a used oil analysis (UOA) is a good indicator for the health of oil lubricated rotating machinery. A UOA monitors the degradation by measuring the wear metal concentration within the oil. When the accumulation of metal parts reaches a given threshold, the oil is no longer good for the component and the

component can be considered as failed. The wear accumulation trend can be approximated by a quadratic function of operating time:

$$x_{i,j}(t) = a + bt + ct^2 \quad (3.7)$$

where $x_{i,j}(t)$ represents the wear metal concentration measured through UOA at time t , a represents the initial wear level, b represents the linear wear coefficient, c represents the nonlinear wear coefficient. The quadratic wear model is adapted from the UOA methodology presented by [1]. The linear term models the gradual accumulation of wear particles during operation, while the quadratic term captures accelerated wear as the component ages.

The simulation framework used in this thesis requires a degradation rate so that the wear state can be updated at each time step. Therefore, the wear accumulation rate is obtained by differentiating the quadratic wear model with respect to time. The resulting wear rate is given by

$$\frac{dx_{i,j}(t)}{dt} = b + 2ct \quad (3.8)$$

where b represents the constant wear accumulation rate and $2ct$ captures the increase in wear rate as the component ages.

To model uncertainty in the wear accumulation process, the wear accumulation rate from Equation (3.8) is introduced as the drift term in a Wiener process, similar to the hydraulic and lag-based degradation models. The degradation state is then updated according to

$$x_{i,j}(t + \Delta t) = x_{i,j}(t) + (b + 2ct) \Delta t + \sigma_x \sqrt{\Delta t} Z(t) \quad (3.9)$$

where σ_x represents the uncertainty associated with the wear accumulation process and $Z(t) \sim \mathcal{N}(0, 1)$ is standard Gaussian noise. The term $(b+2ct)\Delta t$ represents the expected wear accumulated during the current time step, while the stochastic term $\sigma_x \sqrt{\Delta t} Z(t)$ captures random variations in wear generation caused by differences in operating conditions, loading,

lubrication quality, and manufacturing variability. In this scenario, failure occurs when

$$x_{i,j}(t) \geq x_{i,\text{crit}} \quad (3.10)$$

where $x_{i,\text{crit}}$ is the critical wear concentration threshold. A component is considered failed when the wear metal concentration exceeds this critical value.

3.3 Maintenance Policy and Decision Variables

An ideal maintenance policy must take several factors into account. In order to have an understanding of the degradation state $x_{i,j}(t)$ of each component at a given time, an inspection must be conducted in order to collect a degradation reading. It is important to consider how often a system should receive an inspection as performing an inspection too often is costly while waiting too long could cause components to degrade beyond its critical threshold before receiving maintenance [17][14][4]. As a result, the first decision variable in the maintenance policy considered in this thesis is the inspection interval, T_{sys} . Each time an inspection is performed, the degradation levels of all components in a system are examined.

When the degradation readings for the components are read, there are three possible actions that can be conducted. The first action is to simply do nothing when the degradation of a component is $x_{i,j}(t) < \tau_{i,1}$. This occurs when the components are in good condition and their degradation levels are not approaching their critical thresholds. As a result, no maintenance is required and the system may continue to operate as is until the next inspection. In this scenario, the degradation state does not change and continues to degrade stochastically. The next possible maintenance action is repair maintenance when the degradation of a component is within the interval of $\tau_{i,1} \leq x_{i,j}(t) < \tau_{i,2}$. Repair maintenance is conducted before a component reaches its critical degradation threshold so that their degradation level can be reset towards 0 and the life of the component can be extended. The third maintenance action is replacement maintenance when the degradation of a component $x_{i,j}(t) \geq \tau_{i,2}$. Re-

placement maintenance occurs when a component is near a critical degradation level to the point where repairing the component would be more expensive with replacing the component with a new one. Finding a proper threshold value between the repair and replacement actions is critical because it often costs less to repair a component than it does to replace a component. The maintenance actions can be defined below:

$$A_{i,j}(t) = \begin{cases} \text{Do Nothing,} & x_{i,j}(t) < \tau_{i,1} \\ \text{Repair,} & \tau_{i,1} \leq x_{i,j}(t) < \tau_{i,2} \\ \text{Replace,} & x_{i,j}(t) \geq \tau_{i,2} \end{cases} \quad (3.11)$$

Repair is considered to be imperfect. In practice, it is difficult to take a degraded component and repair it so that its degradation state returns to $x_{i,j} = 0$ for a good-as-new condition. Typically, a substantial amount of repair is done at a reasonable cost that brings the degradation state down towards 0, but not completely in good-as-new condition. This degradation state can be defined as better than old, but worse than new [22] [19]. Repairs are assumed to be constant for all components within a subsystem. This assumption reflects the fact that components in a given subsystem typically undergo similar repair procedures and have comparable maintenance techniques. As a result, the repair effectiveness for a subsystem can be defined by the variable, p_i . In this thesis, a constant value for p_i for subsystem i is considered. When repair maintenance is conducted, the updated degradation level can be defined as

$$x_{i,j,\text{new}} = p_i x_{i,j,\text{old}} \quad (3.12)$$

The degradation level can often reach a critical failure threshold in between inspection intervals. During this time, the failed component can do additional damage to the system making it more difficult to repair. This can be defined as corrective maintenance which requires the replacement of the component, but is typically more expensive than replacement

maintenance.

In this thesis, all components within the same subsystem are assumed to share the same repair and replacement thresholds. Therefore, subsystem i is assigned a repair threshold, $\tau_{i,1}$, and a replacement threshold, $\tau_{i,2}$. This reduces the number of decision variables while still allowing each subsystem to have maintenance thresholds that reflect their own degradation behavior and maintenance requirements. The maintenance policy considered in this thesis can be represented as follows [4] [14]

$$\theta = [T_{sys}, \tau_{1,1}, \tau_{1,2}, \tau_{2,1}, \tau_{2,2}, \dots, \tau_{s,1}, \tau_{s,2}] \quad (3.13)$$

where $\tau_{i,1}$ is the repair threshold for subsystem i , and $\tau_{i,2}$ is the replacement threshold for subsystem i . Both thresholds are applied to every component within the subsystem.

3.4 Cost Structure

As stated in Section 3.3, the actions performed during a maintenance inspection come at a cost. The following breakdown of the cost structure is discussed in this section [6]. Every time an inspection occurs, which is determined by the inspection time interval T_{sys} , there is a flat fee of C_{insp} . C_{insp} can be defined as the cost of inspection which involves collecting readings of the degradation states of the system components. If no maintenance action (i.e., repair or replacement) is taken, then no other cost is incurred. However, if a repair is performed, then it comes at the cost of C_{rpr} and if a replacement is performed, then it costs C_{repl} . Corrective maintenance occurs after a component has already failed and therefore represents an unplanned maintenance action. Corrective maintenance is generally more expensive than preventive maintenance because it may involve emergency repairs, unscheduled downtime, and additional operational disruptions. When a component failure occurs, the system redundancy condition is evaluated using the k_i -of- n_i subsystem structure. If the failed component does not cause the subsystem to fall below the required number of

functioning components, then the failed component remains out of service until the next scheduled inspection, at which point corrective maintenance is performed and the cost C_{corr} is incurred. However, if the failure causes the subsystem to fall below the required number of functioning components, then there is a system failure and the immediate action is to perform corrective maintenance for all failed components in the subsystem, where the cost C_{corr} per component is incurred, and an additional system failure penalty of $C_{sysfail}$ is applied [19].

As previously mentioned, a critical challenge in creating a near-optimal policy for a multi-component system is to find an ideal balance between maintenance decisions so that the cost is reduced. These values are affected by the inspection interval, T_{sys} , because it determines how often inspections occur. They are also greatly impacted by the repair and replacement maintenance thresholds because they determine how often the repair/replacement actions occur.

3.5 Objective Function

The objective function to be optimized in this thesis is presented below,

$$J(\theta) = \mathbb{E} \left[\frac{G(\theta)}{H} \right] \quad (3.14)$$

where $J(\theta)$ represents the expected cost per unit time for policy θ , $G(\theta)$ represents the total cost accumulated over the time horizon, and H represents the time horizon. The expectation $\mathbb{E}[\cdot]$ in the objective function is used to consider the uncertainty in the degradation process of each component.

The cumulative cost $G(\theta)$ can be further broken down into the cost of inspection, C_{insp} , the cost of repair, C_{rpr} , the cost of replacement, C_{repl} , the cost of corrective maintenance,

C_{corr} , and the cost of system failure, $C_{sysfail}$. Specifically, $G(\theta)$ can be represented as follows,

$$G(\theta) = N_{insp}C_{insp} + \sum_{i=1}^s \sum_{j=1}^{n_i} \left(N_{rpr,i,j}C_{rpr} + N_{repl,i,j}C_{repl} + N_{corr,i,j}C_{corr} \right) + N_sC_{sysfail}. \quad (3.15)$$

N_{insp} is the total number of inspections and is determined by the inspection interval T_{sys} and the time horizon H . The number of repair actions for component j in subsystem i , $N_{rpr,i,j}$, is determined by the number of inspection times where the degradation state satisfies $\tau_{i,1} \leq x_{i,j}(t) < \tau_{i,2}$. Similarly, the number of replacement actions for component j in subsystem i , $N_{repl,i,j}$, is determined by the number of inspection times where $x_{i,j}(t) \geq \tau_{i,j,2}$. The number of corrective maintenance actions, $N_{corr,i,j}$, is determined by the number of times component j in subsystem i fails between scheduled inspections and requires unscheduled maintenance. The number of system failures, N_s , represents the number of times the overall system fails due to a subsystem violating its k_i -of- n_i redundancy requirement.

As shown, the total cost $G(\theta)$ is determined by the number of inspections, repairs, replacements, corrective maintenance actions, and system failures that occur over the time horizon H . Since the objective is to minimize the expected cost per unit time [21], the optimal policy is the policy vector θ that minimizes $J(\theta)$.

Chapter 4

Methodology

The overall objective of this thesis is to minimize the cost of maintaining a multi-component system through the optimization of maintenance policies. The policy variables include the system inspection interval and the repair and replacement thresholds associated with each component. Due to the stochastic nature of component degradation and the large policy space, a simulation-based optimization framework is utilized. This framework combines Monte Carlo simulation for policy evaluation with a Genetic Algorithm for policy optimization[14].

4.1 Genetic Algorithm Framework

A Genetic Algorithm (GA) is used in this research to solve the maintenance optimization problem. The figure shown below provides an overview of the GA methodology:

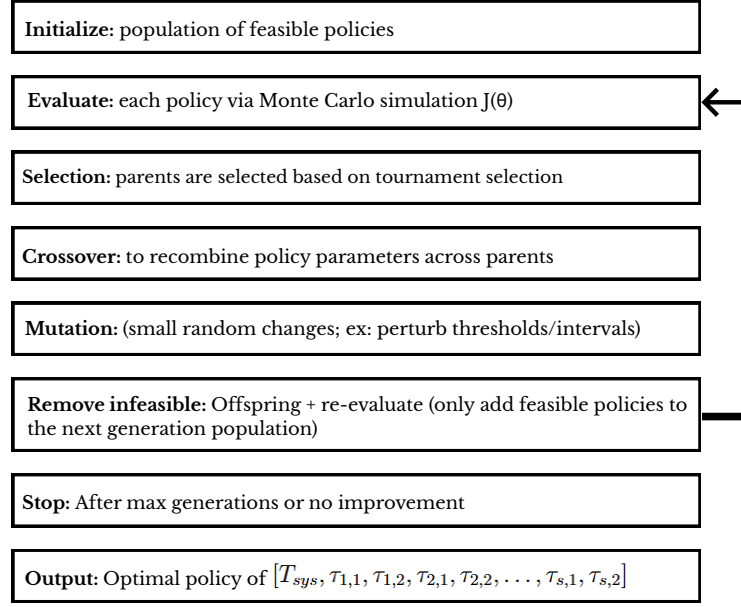


Figure 4.1: Genetic Algorithm Framework

The GA maintains a population of candidate policies in which each policy is represented as a chromosome that is made up of the decision variables which are represented as genes [5] [10]. Initially in any problem, the population will be randomly generated while also respecting any defined constraints to ensure diverse policies in the feasible range. Each of these individual policies are created by using the following method. First, inspection intervals are sampled uniformly from their valid ranges. Then, repair thresholds are sampled uniformly from a restricted range. This is important because it ensures the repairs are ultimately triggered before the component reaches critical degradation. Next, replacement thresholds are sampled from a higher range. This is important because it means that the replacements occur near the critical level while also leaving a margin for safety. After that, the threshold constraint is checked. If at any point, repair threshold is greater than the replacement threshold, then both thresholds must be resampled until the constraint becomes satisfied. In order to avoid an infinite loop, a maximum quantity of re-sampling attempts are allowed. Lastly, once it is verified that the constraints are satisfied, the individual can be added into the population. This initialization procedure ensures that the initial population is both diverse and entirely

feasible while also providing a strong starting point for evolutionary search.

The rest of this section will break down the various sections of the GA framework and provide further detailed explanations.

4.1.1 Selection

The population evolves over successive generations through three genetic operations which are selection, crossover, and mutation. First, selection identifies the high quality individuals that act as parents for the next generation. Parents for reproduction are selected from the current population. Tournament selection is used because it preferentially selects individuals with better fitness values for reproduction without requiring fitness scaling [5]. In addition, it is robust to outliers and poor performing individuals, computationally efficient to implement, and allows the strength of the selection process to be controlled through the tournament size parameter. The selection process is depicted in the following figure:

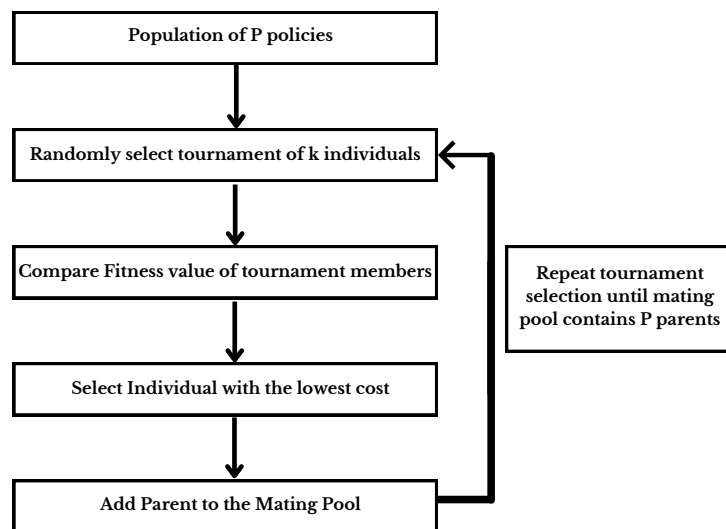


Figure 4.2: Selection

The selection procedure operates as follows:

First a population of P policies are generated. Next, a tournament size k is set to control the likelihood that individuals with higher fitness are selected for reproduction while maintaining diversity within the population. To select a parent, k individuals are first sampled randomly from the population without replacement. The fitness values of the sampled individuals are then compared, and the individual with the lowest fitness value is selected as the parent. The selected parent is added to the mating pool. This process is repeated until the mating pool contains the required number of parents. Overall, this process favors individuals with better fitness values, increasing their probability of selection, while still maintaining diversity by occasionally selecting less-fit individuals.

In addition, an elitism strategy is used to preserve a small number of the best individuals from the current generation [5]. These individuals are copied directly into the next generation without modification, ensuring that high-quality solutions are not lost due to crossover or mutation.

4.1.2 Crossover

The next algorithm operator is crossover, which combines the genetic material from two parent policies to create two offspring. Uniform crossover is an effective method for maintenance optimization because this method allows parameter values to be mixed component-wise across parents [5] [20]. For example, an offspring could possibly inherit the inspection interval from a gene of one parent and the repair threshold of a different gene from another parent. The crossover operator used in this thesis is uniform crossover because each gene represents an independent decision variable. Therefore, exchanging individual genes between parent policies does not disrupt any required ordering of the chromosome. The crossover process is depicted using the following figure:

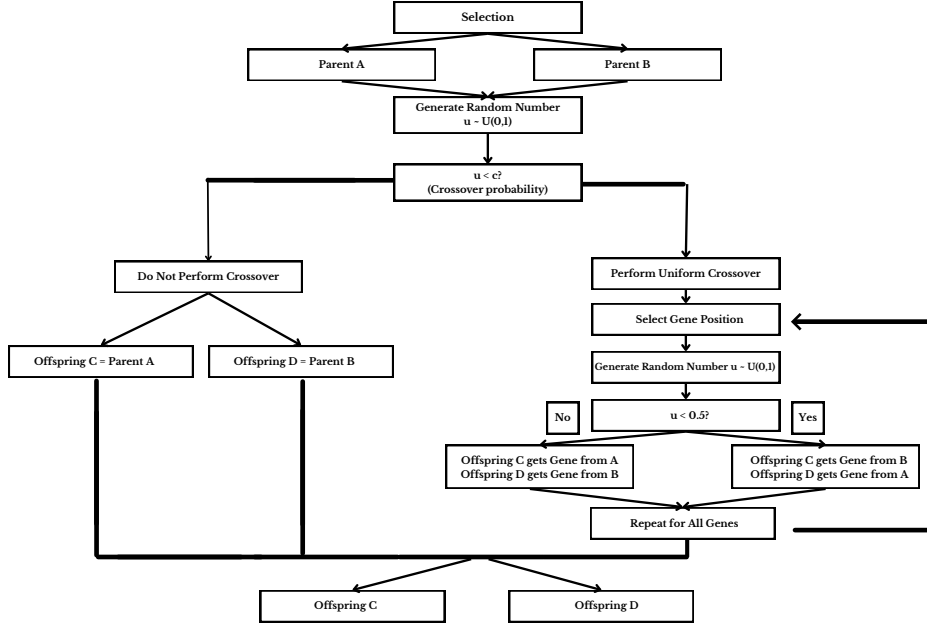


Figure 4.3: Crossover

The uniform crossover procedure will operate as follows. Two parent policies denoted as parent A and parent B, are first selected from the mating pool using the tournament selection procedure described in the previous section. After that, a probability c is specified. A random number, u , is generated from a uniform distribution on the interval $[0, 1]$. If $u < c$ then uniform crossover is performed. Otherwise, the offspring are identical copies of the parent policies.

When crossover is performed, two offspring, C and D are generated by examining each gene position in the policy vector θ . For each gene, a random number, u is generated from a uniform distribution on the interval $[0, 1]$. If $u < 0.5$, offspring C inherits the gene from parent A and offspring D inherits the corresponding gene from parent B. Otherwise, offspring C inherits the gene from parent B and offspring D inherits the corresponding gene from parent A.

Overall, uniform crossover enables different maintenance policies to exchange individual decision variables. In addition, the genetic algorithm is able to combine desirable inspection intervals and maintenance thresholds from different parent policies, allowing ideal charac-

teristics to propagate throughout the population.

4.1.3 Mutation

Mutation introduces random variation by perturbing individual genes with small random changes, typically drawn from a Gaussian distribution [5]. The mutation process is depicted in the following figure:

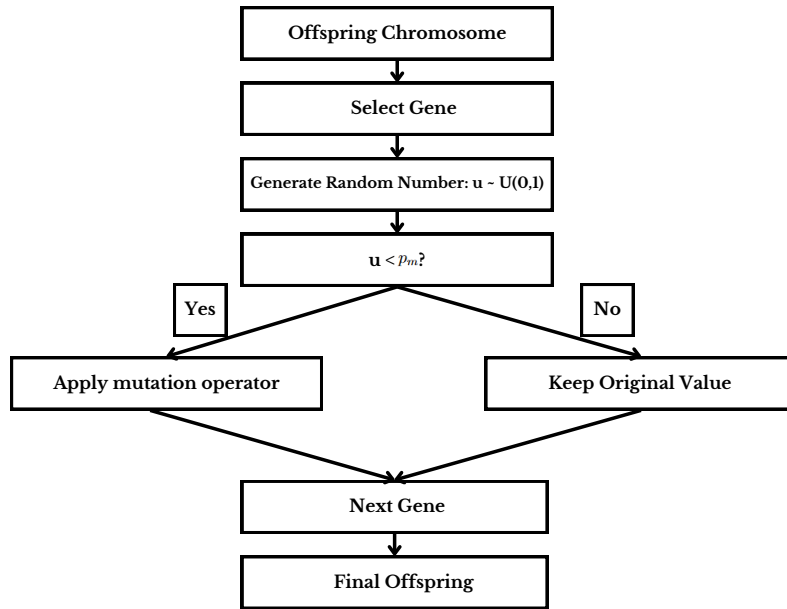


Figure 4.4: Mutation

This ultimately maintains population diversity and prevents premature convergence to local optima [4]. In the mutation process, each gene in each offspring chromosome is subject to mutation with probability p_m . The mutation operator will differ depending on the specific types of genes.

Gaussian mutation perturbs selected decision variables by adding a random value sampled from a gaussian distribution. The mutation process differs depending on the type of decision variable being modified. Inspection interval genes are constrained to remain within the allowable inspection interval bounds. Repair threshold genes are constrained to remain below

the corresponding replacement threshold, while replacement threshold genes are constrained to remain above the associated repair threshold and below the component failure threshold. After mutation is performed, the offspring are checked to ensure that all decision variables satisfy the prescribed parameter bounds and maintenance policy constraints. If an offspring is not feasible, it is removed before the offspring are added to the next generation.

The mutation probability of p_m is kept low to preserve high quality solutions while still introducing diversity into the population. Since the chromosome consists of one inspection interval and two maintenance thresholds for each of the subsystems, it contains $1 + 2s$ genes. As a result, each offspring experiences an average of $(1 + 2s) \times p_m$ mutations per generation. This mutation rate provides sufficient diversity to reduce premature convergence while preserving the overall structure of well performing maintenance policies.

4.1.4 Termination Criterion

The GA terminates when the following condition is met. The criterion for termination occurs when the convergence shows no improvement for a determined quantity of consecutive generations, then the algorithm is considered converged and terminates. This criterion indicates that the GA has exhausted its ability to find better solutions.

At the end of the analysis, the GA will output the best policy found across all generations, the estimated fitness of the best policy, and the generation number at which the policy was discovered.

4.2 Monte Carlo Simulation Procedure

Each candidate policy generated by the Genetic Algorithm is evaluated using Monte Carlo simulation [14]. The simulation estimates the expected cost per unit of time by repeatedly modeling component degradation, maintenance actions, and system failures under uncertainty. The resulting expected cost is assigned as the fitness value of the policy.

4.2.1 Simulation Initialization

Monte Carlo simulation is used to evaluate the performance of candidate maintenance policies. For a given policy θ , repeated simulation runs are performed to estimate the expected cost per unit of time under stochastic degradation. The simulation framework consists of initialization, time-stepping, maintenance decision making, and cost accumulation. The detailed simulation procedure is described in the following sections.

Each Monte Carlo run simulates the operation of a system from time $t = 0$ to a mission horizon, H under a given maintenance policy, θ . The simulation proceeds in discrete time steps. At the start of each run, the simulation state initializes where all components begin in a perfect state with no degradation and all counters used to record the quantity of inspections, repairs, replacements, corrective maintenance actions, and system failures are initialized to zero. The hydraulic pump, generator, and actuator begin with no accumulated degradation, whereas the engine begins with an initial crack length of 0.2 mm. The initial degradation states are selected to represent newly serviced components at the beginning of the operating horizon.

4.2.2 Time-Stepping Loop

The overall Monte Carlo simulation procedure used to evaluate each candidate maintenance policy is illustrated in Figure 4.5 which is displayed below. The framework models component degradation, maintenance actions, and system failures over the mission horizon before estimating the expected long-run cost of the policy.

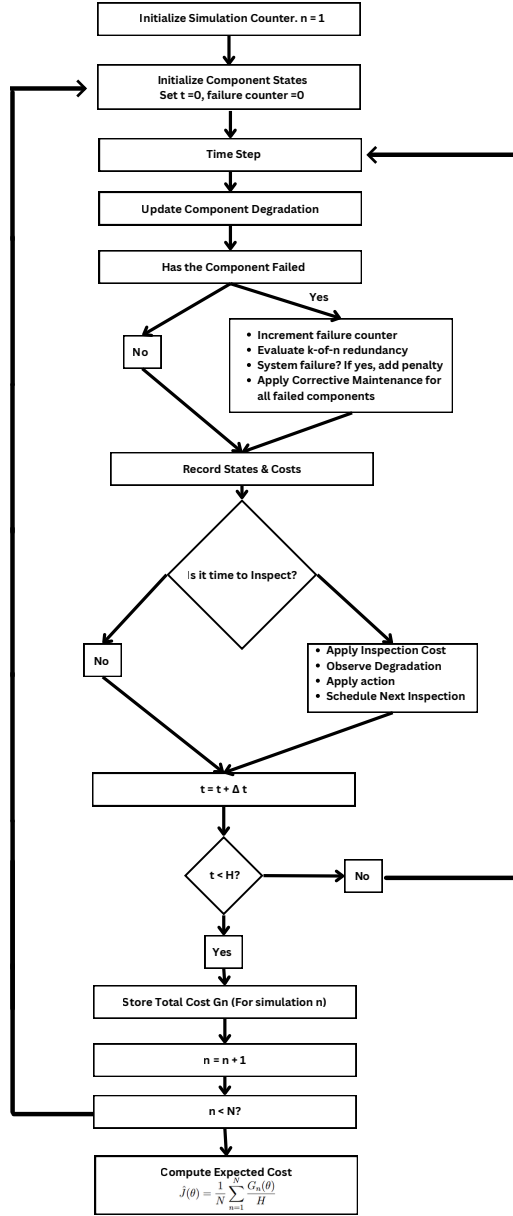


Figure 4.5: Monte Carlo simulation framework used to evaluate candidate maintenance policies.

Following initialization, the Monte Carlo simulation evaluates a candidate maintenance policy by simulating the stochastic degradation of every component over a mission horizon of H . Each simulation advances in discrete time increments of Δt until the mission horizon is reached. To estimate the expected long-run cost associated with a candidate policy, the

complete simulation is repeated for N independent Monte Carlo replications. The average cost across the N simulations is then used as the objective function value supplied to the genetic algorithm.

During each simulation, the degradation state of every operational component is updated at each time step according to its assigned stochastic degradation model. If a component has previously undergone repair or replacement, its degradation state is first restored according to the corresponding maintenance model before degradation continues to evolve.

Following the degradation update, each component is evaluated to determine whether its degradation state exceeds the corresponding failure threshold. When component (i, j) satisfies $x_{i,j}(t) \geq x_{i,\text{crit}}$, the component is declared failed. The system reliability structure is then evaluated to determine whether the component failure results in a system-level failure. If the subsystem no longer satisfies its required k_i -out-of- n_i redundancy, the system failure counter, N_s , is incremented and the corresponding system failure penalty is applied before corrective maintenance is performed.

At each scheduled inspection, the degradation state of every component is observed and maintenance decisions are determined according to the candidate maintenance policy. Components whose degradation states remain below the repair threshold continue operating without maintenance. Components whose degradation states exceed the repair threshold undergo imperfect repair, while components exceeding the replacement threshold are replaced. For any component that remains in the failed state, corrective maintenance is performed and the corrective maintenance counter, $N_{\text{corr},i,j}$, is incremented. Inspection costs and maintenance costs are accumulated and the next inspection is scheduled according to the optimized inspection interval.

After all maintenance actions have been completed, the simulation records the updated degradation states, maintenance events, and accumulated costs before advancing to the next time step. This procedure is repeated until the mission horizon H is reached. Upon completion of the simulation, the total accumulated cost for the current Monte Carlo replication,

G_n , is recorded. Since the expectation in Equation (3.14) cannot be evaluated analytically, it is approximated using N independent Monte Carlo simulations. After all N Monte Carlo replications have been completed, the expected long-run cost is computed as:

$$\hat{J}(\theta) = \frac{1}{N} \sum_{n=1}^N \frac{G_n(\theta)}{H}, \quad (4.1)$$

This estimated cost is then returned to the genetic algorithm as the fitness value for the candidate maintenance policy.

Chapter 5

Case Studies

Commercial aircraft rely on multiple flight-critical subsystems whose degradation over time directly affects safety, reliability, and operational availability. These subsystems—primarily the engines (ATA 72), electrical generators (ATA 24), hydraulic systems (ATA 29), and primary flight control actuators (ATA 27) each experience wear due to mechanical loading, vibration, thermal cycling, and random shock events. Because each component follows its own stochastic degradation process, their reliability characteristics differ substantially across the platform. These four subsystems were selected because they represent major flight-critical systems that exhibit distinct degradation mechanisms, allowing the optimization framework to be evaluated across various degradation models.

5.1 Case Study 1

For the first case study, the aircraft is modeled as a series system consisting of multiple flight-critical subsystems. Each subsystem is represented by a k_i -out-of- n_i reliability structure. For this case study, all subsystems follow a 1-out-of-1 configuration. A single representative component is selected from each major flight-critical subsystem to demonstrate the proposed optimization framework while keeping the case study computationally tractable. A high-level reliability block diagram (RBD) is shown in Figure 5.1, illustrating the system architecture

used in the case study:

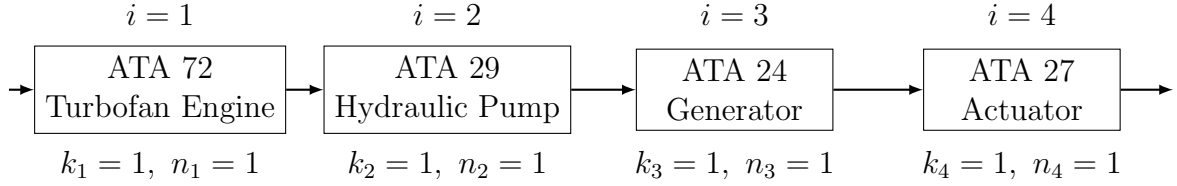


Figure 5.1: Example aircraft-inspired system consisting of four single-component subsystems connected in series. Each subsystem follows a 1-out-of-1 structure, meaning failure of any component causes failure of the overall system.

In this illustrative case study, each subsystem consists of a single critical component ($k_i = n_i = 1$). Therefore, failure of any component results in failure of its corresponding subsystem. This architecture links individual component failures directly to system-level reliability, meaning that poorly timed maintenance on a single subsystem can trigger mission delays, unscheduled outages, or hazardous failures. Maintenance actions such as inspection, repair, and replacement are essential for avoiding these events; however, determining the correct timing is difficult. As previously mentioned, performing maintenance on the aircraft system too early leads to unnecessary cost, while delaying maintenance increases failure risk. Because the degradation of each component is stochastic and interacts with the system-level redundancy structure, identifying an optimal maintenance policy is inherently uncertain and computationally complex. The degradation model parameters and corresponding failure thresholds for each subsystem are summarized later in Tables 5.1–5.4. These values are used throughout the simulation to determine component failures and the resulting maintenance actions. Parameter values were selected from representative degradation studies in the literature and are intended to reflect realistic degradation behavior for each subsystem rather than any specific aircraft platform.

5.1.1 Component-Specific Degradation Models and Parameters

Turbofan Engine: Paris' Law

The turbofan engine degradation is modeled using the stochastic Paris' Law formulation which was presented in Section 3. The parameter values used for the engine degradation model are summarized in Table 5.1 [7]. These values are representative of a turbofan engine and are held constant throughout all simulations.

Table 5.1: Paris' Law crack growth model parameters for the engine.

Parameter	Value	Description
C	3.0×10^{-14}	Paris' Law crack growth coefficient
m	3.6	Paris' Law crack growth exponent
ΔK	20.0	Stress-intensity factor range
σ_ω	0.1	Standard deviation of stochastic crack-growth term
$x_{i,j}(0)$	0.2 mm	Initial crack length
$x_{i,\text{crit}}$	10.0 mm	Critical crack length for failure

In this case study, an initial crack length is assumed to be 0.2 mm to represent small manufacturing defects or naturally occurring microscopic flaws that exist within component [7].

Wiener Process Degradation - Hydraulics

Hydraulic component degradation can be modeled using the Wiener process. The parameter values adopted in the case study are shown below in Table 5.2 [12]. The drift and diffusion coefficients determine the expected degradation rate and stochastic variability.

Table 5.2: Hydraulic component Wiener process degradation parameters.

Parameter	Value	Description
λ	$4.5955 \times 10^{-4} \text{ hr}^{-1}$	Mean degradation drift parameter
σ	$\sqrt{2.7833 \times 10^{-4}} \text{ hr}^{-1/2}$	Diffusion coefficient
$x_{i,j}(0)$	0.0	Initial degradation state
$x_{i,\text{crit}}$	0.085	Critical degradation threshold
$x_{i,\text{crit}}$	1.0	Critical normalized degradation threshold
Δt	24 hr	Simulation time step

The measured degradation value for the hydraulic pump is based on a dimensionless degradation index that characterizes the overall health of the component used in the Wiener process model rather than a physical quantity [12]. The degradation index represents the cumulative effect of degradation mechanisms such as wear, internal leakage, friction, contamination, and other age-related deterioration. The original hydraulic critical degradation threshold is 0.085. In this thesis, the degradation state is normalized by this critical value so that failure occurs when $x_{i,j}(t) = 1.0$.

Lag-Based Wiener Process Degradation - Actuator

Actuator degradation is represented by the observable lag model which is discussed in Section 3. The parameters that define the lag evolution and failure threshold are shown below in Table 5.3.

Table 5.3: Lag-based actuator degradation model parameters.

Parameter	Value	Description
$x_{i,j}(0)$	0.0 s	Initial actuator lag state
$x_{i,\text{crit}}$	2.5 s	Critical actuator lag threshold
λ	8.33×10^{-3} s/hr	Mean actuator lag growth rate
σ	1.25×10^{-3} s/ $\sqrt{\text{hr}}$	Diffusion coefficient for actuator lag uncertainty
Δt	24 hr	Simulation time step

Rotating Part Degradation - Generator

The rotating part degradation model discussed in Section 3 can be used to model the degradation of the rotating parts within the generator. The model measures wear accumulation by considering the ppm of impurities in the oil. Table 5.4 [1] displays the parameters of the wear accumulation process below.

Table 5.4: Rotating part wear accumulation parameters for the generator UOA model.

Parameter	Value	Description
a	0.0 ppm	Initial wear metal concentration
b	0.05 ppm/hr	Linear wear accumulation coefficient
c	1.0×10^{-4} ppm/hr ²	Nonlinear wear accumulation coefficient
σ_x	0.05 ppm/hr ^{1/2}	Diffusion coefficient
$x_{i,\text{crit}}$	150.0 ppm	Critical wear concentration threshold
Δt	24 hr	Simulation time step

Cost Parameters

Maintenance costs are required in order to evaluate candidate maintenance policies. Actions such as inspection, repair, replacement, corrective maintenance and system failure come at a cost. These parameters are shown below in Table 5.5 and are fixed throughout the

optimization process.

Table 5.5: Cost and imperfect repair parameters used in the simulation.

Component	C_{rpr}	C_{repl}	C_{corr}	p_i
Engine	\$25,000	\$120,000	\$250,000	0.30
Generator	\$2,000	\$12,000	\$25,000	0.20
Hydraulic	\$1,500	\$8,000	\$18,000	0.35
Actuator	\$3,000	\$18,000	\$40,000	0.25
$C_{\text{insp}} = \$50,000$ per inspection				
$C_{\text{sysfail}} = \$1,000,000$ per system-level failure				

Genetic Algorithm Parameters

The optimization framework is implemented using a genetic algorithm. Each Monte Carlo simulation is performed over an operating horizon of $H = 100,000$ hours using a simulation time step of $\Delta t = 24$ hours. Table 5.6 below displays the parameters for the optimization procedure. This includes the population size, crossover probability, mutation probability, stopping criteria, and Monte Carlo evaluation settings.

Table 5.6: Genetic algorithm parameters used for policy optimization.

Parameter	Value	Description
Population size	50	Number of candidate policies in each generation
Crossover probability	0.8	Probability that crossover is applied
Mutation probability	0.1	Probability that mutation is applied
Gene mutation probability	0.1	Probability that an individual gene is mutated
Tournament size	3	Number of individuals sampled during tournament selection
Monte Carlo runs	250	Number of simulation replications per policy evaluation
Time horizon, H	100,000	Simulation time horizon
Time step, Δt	24.0	Simulation time step
Patience	10	Early stopping limit for generations without improvement
Minimum improvement	0.001	Minimum cost improvement required to reset early stopping

The parameter values presented in this chapter define the aircraft-inspired case study used throughout the thesis. Below are the results of the study.

5.1.2 Results and discussion

The code was run and converged after 47 generations. The overall results are shown in the table below.

Table 5.7: Optimal maintenance policy obtained from the genetic algorithm.

Subsystem	Repair Threshold	Replacement Threshold
Engine	5.138 mm	8.579 mm
Hydraulic	0.1024	0.3897
Generator	17.9 ppm	137.5 ppm
Actuator	1.112 s	1.933 s

Overall, the results of this table were found after the GA converged after 47 simulations. The near-optimal time interval for inspection was 211.9 hours along the 100,000 hour horizon which meant that a total of 471 inspections took place along the time horizon. The engine had a repair threshold of 5.138mm and a replacement threshold of 8.579mm. The Generator had a repair threshold of 17.9ppm and a replacement threshold of 137.5ppm. The Hydraulic pump had a repair threshold of 0.1024 and a replacement threshold of 0.3897. The Actuator had a repair threshold of 1.112s and a replacement threshold of 1.933s. In the end, the overall optimized cost per hour was \$303.79/*hr*.

Fitness Convergence Curve

After the analysis, the developed code generates two graphs which display the average fitness convergence and the best policy fitness convergence. A third graph is generated to display the cost distribution over 1000 Monte Carlo distributions. The three graphs are displayed below.

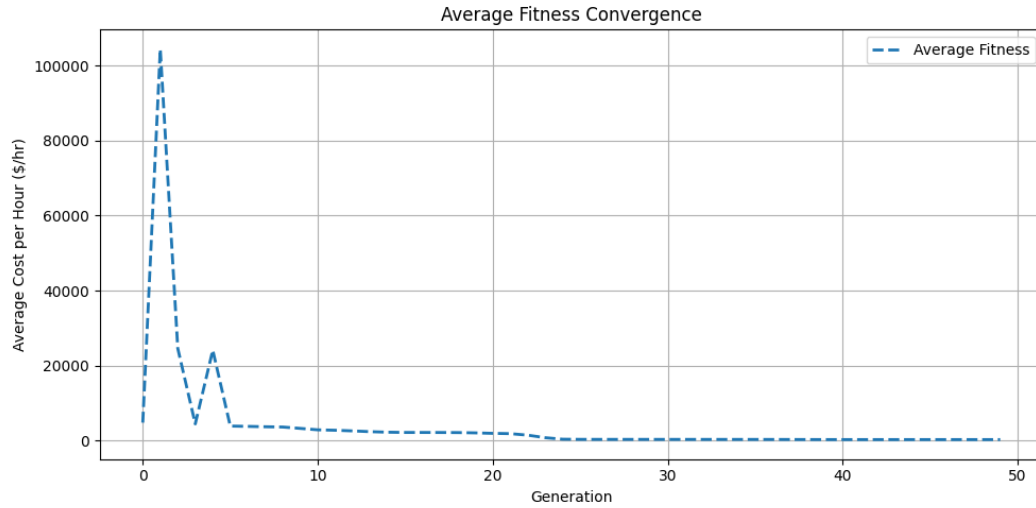


Figure 5.2: Average Fitness convergence curve showing the evolution of the average fitness values across generations.

Figure 5.2 shows that on average the policies start out in a higher range of \$20,000 – \$100,000. Around the 22nd generation, the average policy begins to converge. Following that, the rest of the generations explore the region as the algorithm learns what sorts of decision variables yield policies with the best results.

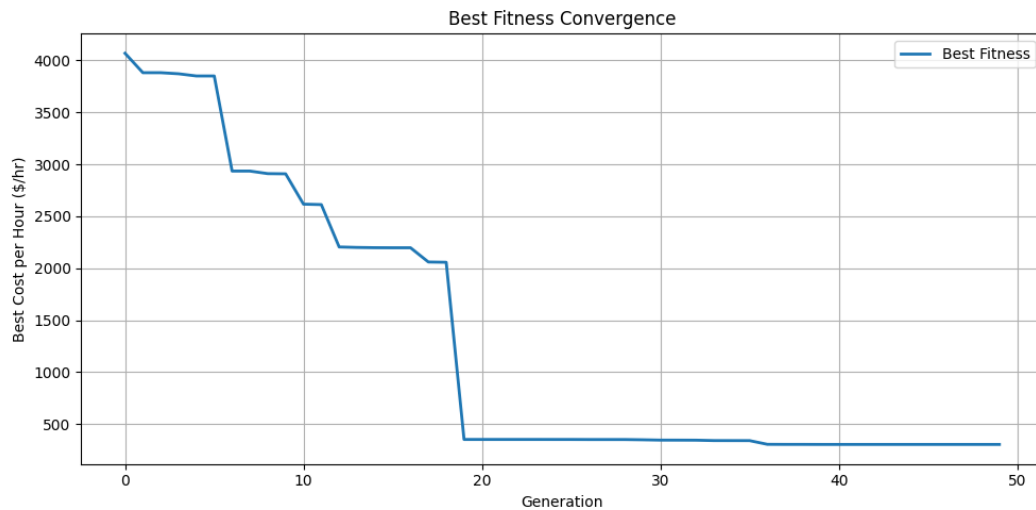


Figure 5.3: Best Policy Fitness convergence curve showing the evolution of the best fitness values across generations.

The convergence curve indicates that the genetic algorithm successfully improved the maintenance policy over successive generations. Early generations show larger variation in both the best and average fitness values, which reflects broad exploration of the search space. As the algorithm gets to the 19th generation, the best fitness reaches a lower and more stable range, indicating that the population is increasingly concentrated around better-performing solutions. While some oscillation occurs between generation 20 through 35, no major improvement is observed after roughly the mid-to-late generations, suggesting that the algorithm has approached a near-optimal solution. The policy finally converges on the 47th generation where the cost per hour of the best policy is output. The final best policy corresponds to an optimized cost of approximately \$303.79 per hour.

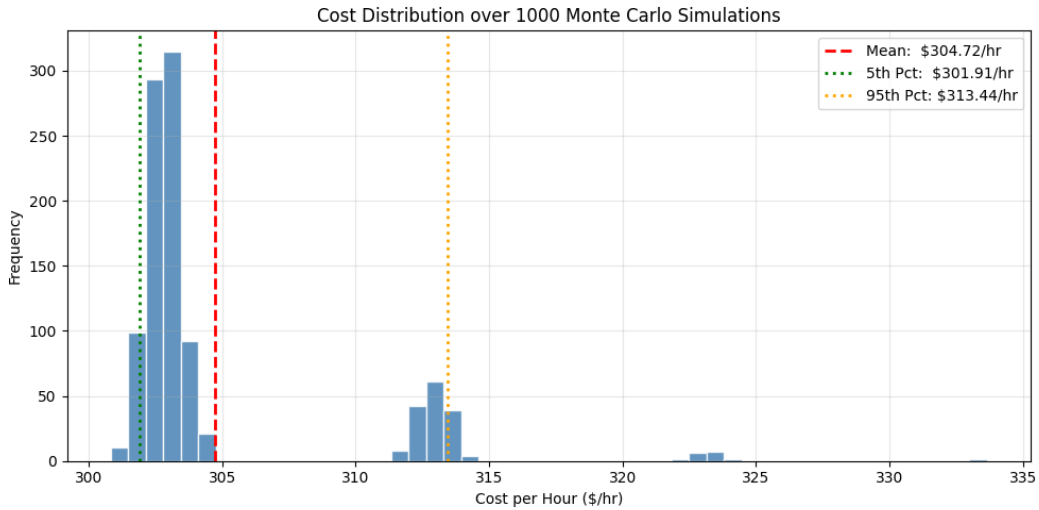


Figure 5.4: Cost distribution of total cost across 1000 Monte Carlo simulation runs.

Figure 5.4 shows the distribution of total cost obtained from 1000 Monte Carlo simulation runs under the optimized maintenance policy. The spread of the distribution reflects the stochastic nature of component degradation and failure, as well as the randomness in maintenance actions. It is important to note that there are various clusters that are spread out. This includes a large cluster around \$302 – 304/hr and a second one around \$313/hr.

Sensitivity Analysis

To evaluate how changes in the economic assumptions influence the maintenance policy selected by the genetic algorithm, a sensitivity analysis was conducted. The analysis was performed by independently increasing and decreasing the inspection, repair, replacement, corrective maintenance, and system failure costs by $\pm 5\%$. For each cost perturbation, the genetic algorithm was rerun to determine a new near-optimal maintenance policy and the corresponding expected long-term operating cost. Since the objective function is directly dependent on the maintenance cost parameters, changes in the optimized operating cost are expected. Therefore, the purpose of the sensitivity analysis is to identify which economic parameters have the greatest influence on the maintenance policy selected by the genetic algorithm and the resulting operating cost.

Percent Variation	Cost Parameter	Inspection Interval	Engine Repair	Engine Replace	Generator Repair	Generator Replace	Hydraulic Repair	Hydraulic Replace	Actuator Repair	Actuator Replace	Total Cost	Percentage Change
Baseline:		211.9	5.138	8.579	17.9	137.5	0.1024	0.3897	1.112	1.933	\$303.79	0.00%
-5%	Inspection Interval	211.9	6.92	9.951	20.9	133.7	0.1472	0.5378	1.059	1.933	\$ 289.33	-4.76%
	Engine repair	211.9	5.655	7.796	18.1	140.5	0.1663	0.4268	1.328	1.891	\$ 302.81	-0.32%
	Generator Repair	211.9	7.503	9.054	17.9	140.8	0.1555	0.5378	0.4	1.891	\$ 300.45	-1.10%
	Hydraulic Repair	211.9	2.637	7.796	40.6	139.2	0.1555	0.5315	0.954	1.933	\$ 300.04	-1.23%
	Actuator Repair	211.9	5.929	7.796	60.8	135.6	0.224	0.5046	1.06	1.933	\$ 299.15	-1.53%
	Engine Replace	211.9	2.637	9.408	63.4	137.9	0.1555	0.5474	0.311	1.934	\$ 299.35	-1.46%
	Generator Replace	211.9	0.916	8.53	23.6	137.5	0.1555	0.5351	1.385	1.891	\$ 300.75	-1.00%
	Hydraulic Replace	211.9	4.065	6.022	76.7	85.6	0.1821	0.8067	0.793	2.469	\$ 254.31	-16.29%
	Actuator Replace	211.9	4.938	8.158	45.4	116	0.1555	0.5322	0.346	2.452	\$ 261.12	-14.05%
	Corrective Engine	211.9	6.92	9.32	23.8	126.6	0.1461	0.5378	1.059	1.912	\$ 296.66	-2.35%
	Corrective Generator	211.9	5.477	8.491	17.9	106.1	0.1805	0.5378	1.059	1.933	\$ 296.52	-2.39%
	Corrective Hydraulic	279.6	5.477	8.03	42.8	120.1	0.1275	0.5753	1.036	1.896	\$ 259.30	-14.64%
	Corrective Actuator	279.6	4.589	8.025	68.3	94.5	0.1221	0.5753	0.986	1.98	\$ 255.12	-16.02%
	System Failure	211.9	2.911	8.132	41.4	126.6	0.1461	0.5378	1.078	1.98	\$ 295.69	-2.67%
5%	Inspection Interval	211.9	4.938	7.797	22.0	126.6	0.1472	0.5370	1.093	1.882	\$ 308.23	1.46%
	Engine repair	211.9	4.710	8.660	21.4	136.8	0.1095	0.4175	1.090	1.945	\$ 305.42	0.54%
	Generator Repair	211.9	5.340	8.490	22.8	139.6	0.1150	0.4250	1.185	1.912	\$ 304.98	0.39%
	Hydraulic Repair	279.6	4.260	7.940	46.7	128.4	0.0715	0.3180	0.965	1.872	\$ 312.64	2.91%
	Actuator Repair	211.9	5.010	8.370	38.5	132.0	0.1260	0.4440	1.028	1.904	\$ 305.73	0.64%
	Engine Replace	211.9	5.420	8.860	25.7	136.1	0.1100	0.4210	1.164	1.922	\$ 305.91	0.70%
	Generator Replace	211.9	5.180	8.520	20.1	141.8	0.1110	0.4280	1.134	1.941	\$ 305.20	0.46%
	Hydraulic Replace	279.6	3.820	7.310	74.6	119.5	0.0610	0.2860	0.902	1.846	\$ 319.88	5.30%
	Actuator Replace	211.9	5.290	8.440	31.6	130.7	0.1210	0.4380	1.087	2.018	\$ 304.67	0.29%
	Corrective Engine	211.9	4.960	8.740	24.2	135.0	0.1070	0.4090	1.151	1.926	\$ 306.14	0.77%
	Corrective Generator	211.9	5.260	8.510	19.7	143.2	0.1130	0.4310	1.122	1.918	\$ 305.84	0.67%
	Corrective Hydraulic	146.5	3.550	7.120	88.9	116.8	0.0580	0.2740	0.874	1.831	\$ 321.46	5.82%
	Corrective Actuator	211.9	4.740	8.180	44.3	124.6	0.1310	0.4590	0.978	1.865	\$ 307.28	1.15%
	System Failure	146.5	4.220	7.880	52.1	126.4	0.0830	0.3420	1.006	1.889	\$ 313.72	3.27%

Figure 5.5: Sensitivity analysis results for Case Study 1.

Overall, the optimized maintenance policy was most sensitive to variations in the hydraulic subsystem cost parameters. Perturbations to the hydraulic replacement and corrective maintenance costs produced the largest changes in both the optimized maintenance policy and the expected operating cost per hour. Decreasing the hydraulic replacement cost by 5% reduced the optimized operating cost by 16.29%, while decreasing the hydraulic corrective maintenance cost reduced the operating cost by 14.64%. Conversely, increasing the hydraulic replacement and corrective maintenance costs increased the operating cost by 5.30% and 5.82%, respectively. The operating cost is also sensitive to the actuator replacement and corrective maintenance costs when they are decreased by 5%, but not when they are increased by 5%. In comparison, perturbations to the engine, generator, and inspection costs generally produced changes of less than 3% relative to the baseline operating cost. These results indicate that the changes in the hydraulic subsystem has the greatest influence on the optimization results for Case Study 1.

The inspection interval experienced some variations depending on which cost parameters were perturbed. In most scenarios, the cost variations maintained the baseline inspection interval of 211.9 hours. When the hydraulic repair or replacement costs were increased by 5%, the optimized inspection interval increased to 279.6 hours. The inspection interval remained unchanged for most cost perturbations because the optimizer was able to compensate by adjusting repair and replacement thresholds. Modifying these maintenance thresholds allowed the maintenance strategy to respond to changes in the economic assumptions without requiring changes to the inspection schedule. When the hydraulic corrective maintenance cost and the system failure cost were increased by 5%, the inspection interval reduced to 146.5 hours. Overall, these changes display the trade-off between inspection frequency and maintenance costs is accounted for by the genetic algorithm. When the consequences of corrective maintenance or system failure were increased by 5%, the optimizer selected shorter inspection intervals in order to increase the probability of detecting degradation before failure occurred. In contrast, increasing the cost of hydraulic repair and replacement resulted in

longer inspection intervals. This indicates that the GA selected a different balance between inspection frequency and preventive maintenance under the modified economic assumptions.

The repair and replacement thresholds across most of the components had moderate change based on the parameter variations. The hydraulic-related cost perturbations produced the largest changes in the repair and replacement thresholds, indicating that the genetic algorithm primarily adjusted the hydraulic maintenance strategy in response to changes in the assumed economic parameters. This behavior indicates that the genetic algorithm adjusted the maintenance strategy of the hydraulic subsystem in response to changes in the assumed economic parameters, while the remaining subsystem policies required smaller modifications. In the end, these results suggest that hydraulic maintenance decisions contribute best to the economic trade-offs that were considered during optimization.

The optimized maintenance policy did not always vary monotonically with the perturbed cost parameters. This behavior is expected because the optimization problem is both non-linear and stochastic, allowing multiple maintenance policies to produce similar objective values. Consequently, small changes in the economic assumptions may cause the genetic algorithm to converge to different, but nearly equivalent, maintenance policies that achieve comparable operating costs. Therefore, the sensitivity analysis should be interpreted by identifying the cost parameters that have the greatest influence on the optimized maintenance policy rather than expecting every decision variable to change monotonically.

5.2 Case Study 2

In this next illustrative case study, the system is based on an industry example of a $(k_i\text{-out-of-}n_i)$ system. The degradation model parameters, corresponding thresholds, cost parameters, and the GA architecture parameters remained the same as they did in subsection 5.1. These values are used throughout the simulation of this industry case study to determine component failures and the resulting maintenance actions. For this scenario, the $(k_i\text{-out-of-}n_i)$ system

for each subsystem is 1-of-2 for the turbofan engine subsystem, 2-of-4 for the generator subsystem, 1-of-3 for the hydraulic pump subsystem, and 5-of-8 for the actuator subsystem. The reliability block diagram and the results for this case study are shown below.

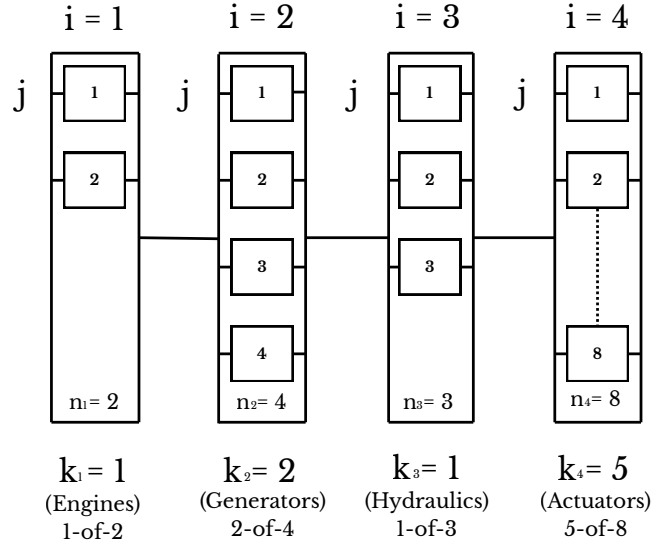


Figure 5.6: Aircraft-inspired complex (k_i -out-of- n_i) system

5.2.1 Results and Discussion

The code was run and converged after 97 generations. The overall results are shown in the table below.

Table 5.8: Optimal maintenance policy obtained from the genetic algorithm for the expanded (k_i -out-of- n_i) case study.

Subsystem	Repair Threshold	Replacement Threshold
Engine	3.914 mm	8.243 mm
Hydraulic	0.4517	1.0000
Generator	64.6 ppm	142.0 ppm
Actuator	1.294 s	1.953 s

Overall, the results of this table were found after the GA converged after 97 generations.

The near-optimal Time Interval for inspection was 146.5 hours along the 100,000 hour horizon which meant that a total of 682 inspections took place along the time horizon. The engine had a repair threshold of 3.914mm and a replacement threshold of 8.243mm. The Generator had a repair threshold of 64.6ppm and a replacement threshold of 142.0ppm. The Hydraulic pump had a repair threshold of 0.4517 and a replacement threshold of 1.0. The Actuator had a repair threshold of 1.294s and a replacement threshold of 1.953s. In the end, the overall optimized cost per hour was \$494.76/hr.

Compared to the results found from the first case study, the expanded k -of- n architecture required about 45% more frequent inspections by decreasing from an interval of 211.9 hours to 146.5 hours. The second case study also produced a higher expected operating cost of \$494.76/hr. These overall differences display the impact of increased system complexity and redundancy on the optimized maintenance policy.

One value that stands out is the optimized replacement threshold of the hydraulic pump being 1.0. This result however can be explained by further analyzing the architecture of the genetic algorithm. It is important to note that the k -of- n structure for the hydraulic subsystem is 1-of-3 and the repair cost is \$1500 compared to the replacement cost of \$8000. As a result, the genetic algorithm has little economic incentive to schedule preventive replacement before the component reaches the failure threshold. Instead, the optimized policy favors repeated repairs while relying on the subsystem redundancy to maintain operation, causing the replacement threshold to converge to the maximum allowable value of 1.0.

Fitness Convergence Curve

After the algorithm for the second case study is run, three graphs are generated which display the average fitness convergence, the best policy fitness convergence, and the cost distribution over 1000 Monte Carlo distributions for the best policy. The three graphs and an analysis are displayed below:

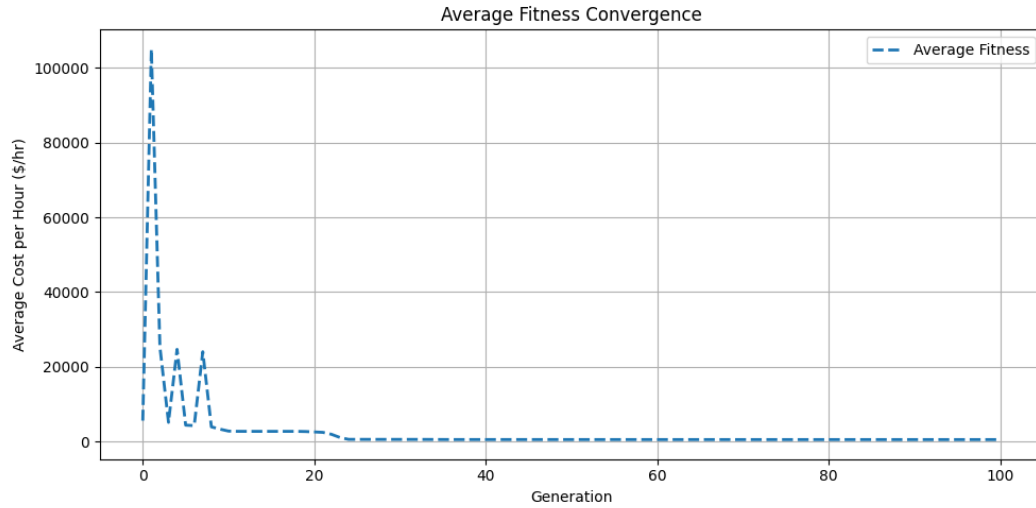


Figure 5.7: Average Fitness convergence curve showing the evolution of the average fitness values across generations.

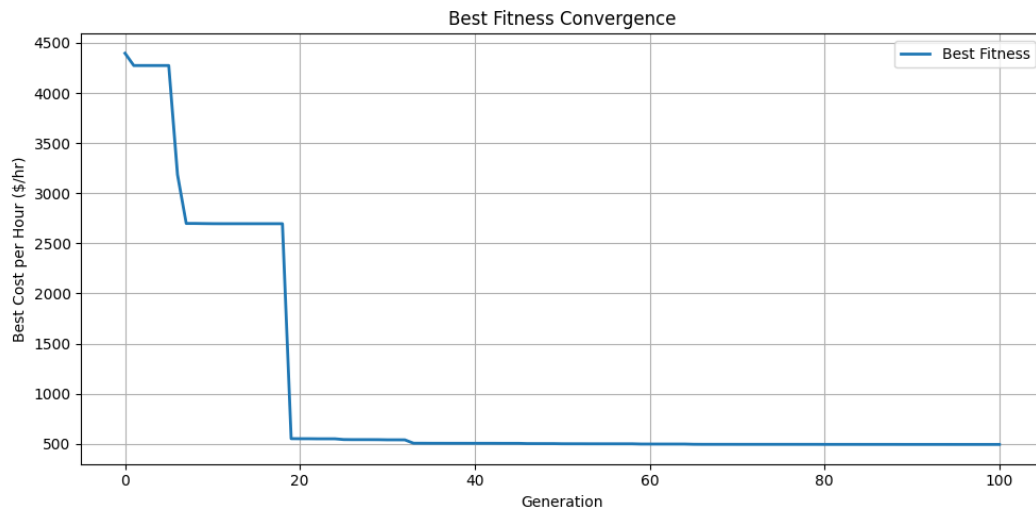


Figure 5.8: Best Policy Fitness convergence curve showing the evolution of the best fitness values across generations.

The convergence curves for both the average policy and the best policy indicate that the genetic algorithm successfully improved the maintenance policy over successive generations. Early generations up until about the 20th generation show larger variation in both the best and average fitness values, which reflects broad exploration of the search space. As the

algorithm proceeds, the best fitness reaches a lower and more stable range around the 30th generation, while the average fitness also trends downward, indicating that the population is increasingly concentrated around better-performing solutions. After the 35th generation, the best policy convergence graph has a downward trend that is barely noticeable until it finally converges on the 97th generation. This suggests that the algorithm has identified promising regions of the search space and continues refining candidate maintenance policies through selection, crossover, and mutation. The final best policy corresponds to an optimized cost of approximately \$494.76 per hour.

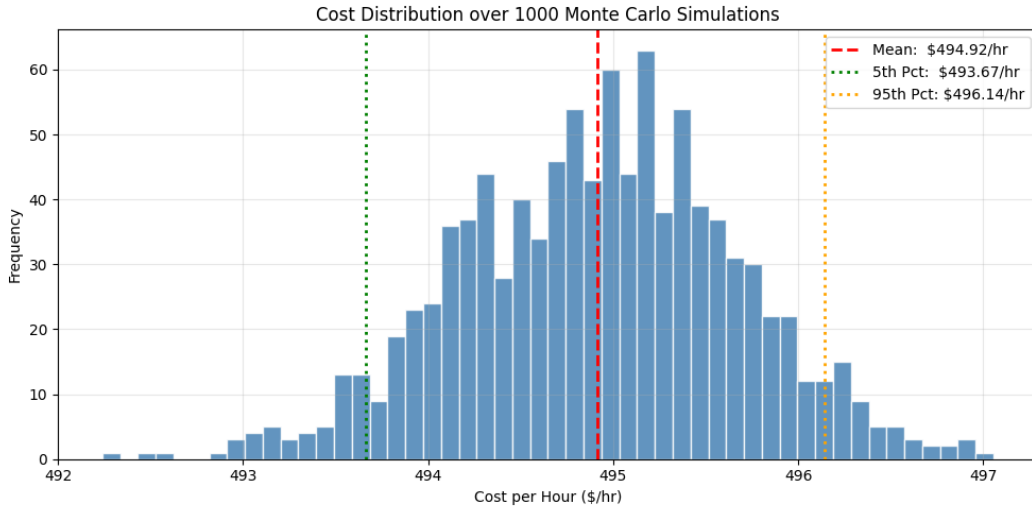


Figure 5.9: Cost distribution of total cost across 1000 Monte Carlo simulation runs.

Figure 5.9 shows the distribution of total cost obtained from 1000 Monte Carlo simulation runs under the optimized maintenance policy. The spread of the distribution reflects the stochastic nature of component degradation and failure. The distribution appears to be centered around the expected cost value of \$494.76/hr, with most simulation outcomes clustered near the mean in a bell-shaped distribution. This indicates that the estimated cost per hour is stable and not driven by a small number of extreme outcomes. In addition, the 5th and 95th percentile costs only have a difference of \$2.47/hr which indicates that the optimized policy is consistent over the long time horizon despite there being unpredictable

stochastic degradation. This small variability shows that the maintenance policy is robust and that the expected total cost value is stable across many simulation trials.

Additionally, the absence of extreme outliers suggests that the policy performs consistently across different simulated scenarios. This provides confidence that the optimized policy is robust and does not result in highly variable or unpredictable costs. Overall, the Monte Carlo results validate that the estimated cost is a reliable representation of long-run system performance under uncertainty.

Sensitivity Analysis

Due to the large computational load, a comprehensive sensitivity analysis was not performed for the complex k -of- n case study. This highlights one of the primary limitations of simulation-based genetic algorithm optimization which is that as a system complexity increases the computational effort required also increases significantly. Future work on this thesis could reduce the computational burden through more efficient optimization algorithms or high-performance computing.

5.3 Discussion of Research Objective and Model Challenges

5.3.1 Corrective-only Baseline Comparison

While the Genetic Algorithm optimized and converged on a near optimal solution, it is necessary to make a comparison to an alternative baseline method to display the capabilities of the genetic algorithm. To create a baseline policy, the scenario of case study 1 was used once again, but this time there were no inspection intervals or maintenance action thresholds. Instead, during the time horizon of the simulation, the parts would undergo stochastic degradation. In the event of a component failure, corrective maintenance would

be conducted immediately. The results of the baseline policy are shown below:

Table 5.9: Comparison of the optimized maintenance policy with the corrective-only baseline for Case Study 1.

Policy	Inspections	Preventive Actions	Expected Cost (\$/hr)	Cost Reduction (%)
Corrective-only baseline	No	No	4,762.26	–
GA-optimized policy	Yes	Yes	303.79	93.62

The results of the baseline policy in which only corrective maintenance actions are conducted resulted in an expected cost of \$4,762.26/hr while the algorithm produced a policy that cost \$303.79/hr. As a result, the optimized policy was able to offer a policy that with a 93.62% decrease in cost for a total decrease of \$4,458.47/hr. The corrective-only baseline was evaluated over 1,000 Monte Carlo replications and produced a 95% confidence interval from \$4,758.86/hr to \$4,765.66/hr. Although scheduled inspections and preventive maintenance introduced additional costs to the optimized policy, they resulted in a decrease in corrective maintenance costs as well as system failure penalties. As a result, it is demonstrated that the optimized policy was able to provide an economic benefit relative to the scenario in which the components ran until failure under the assumed parameters of case study 1.

5.3.2 Evaluation of Research Objectives

A Monte Carlo based framework was developed in order to address several challenges that are associated with maintenance policy optimization for a multi-component system. In the Introduction, these challenges were discussed, and the objective stated an intent to design a framework capable of optimizing a maintenance policy despite these challenges. The case studies were designed to demonstrate whether heterogeneous degradation models, imperfect maintenance, system redundancy, and stochastic cost evaluation could be integrated within a common simulation-optimization framework. The following checklist summarizes the capabilities of the framework which were demonstrated through two case studies. A check

mark indicates that the challenge has been overcome.

Table 5.10: Evaluation of research objectives and model capabilities.

Research Objective or Model Capability	Case Study 1	Case Study 2	Implementation or Finding
Stochastic component degradation	✓	✓	Monte Carlo simulation generated stochastic component degradation trajectories over the operating horizon.
Heterogeneous degradation models	✓	✓	Paris' Law, Wiener-process models, actuator lag, and rotating-part wear accumulation were integrated within one simulation framework.
Multi-component system architecture	✓	✓	Both case studies contained engine, generator, hydraulic, and actuator subsystems connected in series.
k_i -out-of- n_i subsystem redundancy	□	✓	Case Study 1 used a simplified 1-out-of-1 structure, while Case Study 2 included parallel components and subsystem-specific redundancy requirements.
Periodic system inspections	✓	✓	The inspection interval was included as a system-level decision variable in the optimized maintenance policy.
Repair and replacement decisions	✓	✓	Subsystem-specific repair and replacement thresholds defined the condition-based maintenance actions.
Imperfect preventive repair	✓	✓	Repair reduced the component degradation state without generally restoring the component to a good-as-new condition.

Continued on next page

Table 5.10 – Continued from previous page

Research Objective or Model Capability	Case Study 1	Case Study 2	Implementation or Finding
Failures between scheduled inspections	✓	✓	The simulation detected failures between inspections and applied corrective-maintenance costs and system-failure penalties when appropriate.
Expected long-run cost evaluation	✓	✓	Repeated Monte Carlo simulations estimated the expected cost per operating hour for each candidate policy.
Genetic algorithm policy optimization	✓	✓	The genetic algorithm optimized the inspection interval and subsystem repair and replacement thresholds.
Corrective-only baseline comparison	✓	□	The optimized Case Study 1 policy was compared with a corrective-only strategy containing no scheduled inspections or preventive maintenance.

As shown, the framework was able to accomplish the primary research objectives of integrating heterogeneous stochastic degradation models, inspection intervals, imperfect maintenance, and system reliability logic within a simulation-optimization framework. The first case study demonstrated the methodology by using a series system in which each subsystem consisted of a single component. The second case study demonstrated that the same framework could accommodate a larger series-parallel architecture with subsystem k-out-of-n redundancy requirements.

Chapter 6

Conclusion and Future Work

An optimal maintenance policy is critical for multi-component systems such as those used in the aviation industry in order to cut operational costs. Systems with many parts require constant inspections and maintenance actions that have to be coordinated by teams at a cost. Developing a systematic approach for identifying near-optimal maintenance policies is therefore critical for reducing long-run operating costs. One of the major challenges associated with maintenance optimization under stochastic degradation is the high computational cost required to evaluate candidate maintenance policies. In practice, when components fail before they can be repaired or replaced, unexpected corrective maintenance must be done and in the event of a system failure, the costs are much higher. As a result, there is a strong need for condition-based maintenance optimization that has the ability to keep track of component degradation and generate a near-optimal policy.

A key research contribution of this thesis was the development of a maintenance optimization framework for multi-component aircraft-inspired systems. In this thesis, components are assumed to undergo stochastic degradation. Four degradation models were utilized to represent different physical failure mechanisms. They were Paris' Law to model crack propagation, the Wiener process to model wear, a lag-based Wiener process to model actuator lag, and a rotating machinery wear model to represent the accumulation of wear-metal particles.

In practice, repair is considered to be imperfect as it is very difficult to have a team perfectly repair a part so that it is as good-as-new. As for the system architecture, a reliability block diagram was used which displayed components in k -of- n subsystems that were connected to each other in series. Monte Carlo simulation was used to create a framework for maintenance evaluation while a genetic algorithm was used to optimize the inspection intervals and maintenance thresholds. Once the framework was established, two aircraft-inspired case studies were used to evaluate the proposed methodology.

The two case studies were implemented to demonstrate the framework considered in this thesis. In the simplified single-component per subsystem case study, the genetic algorithm converged to a near-optimal maintenance policy with an inspection interval of 211.9 hours and an expected operating cost of \$303.79 per hour. In the more complex k -of- n system, the optimal inspection interval decreased to 146.5 hours while the expected operating cost increased to \$494.76 per hour. The reduction in inspection interval reflects the increased maintenance requirements associated with a larger and more redundant system, demonstrating that the optimization framework adapts maintenance policies according to system architecture and degradation behavior.

A sensitivity analysis was conducted to evaluate how changes in the assumed cost parameters influenced the near-optimal maintenance policy and expected operating cost. To do this, the cost parameters were individually increased and decreased by $\pm 5\%$ within the simplified case study. The sensitivity analysis showed that hydraulic replacement and corrective maintenance costs produced the largest changes in the near-optimal policy and expected operating cost.

Repeated Monte Carlo simulations demonstrated that the optimized policies produced stable long-run operating costs. The second case study exhibited a narrow, approximately bell-shaped cost distribution, while the first case study highlighted how different stochastic degradation trajectories can produce multiple operating cost regimes.

Overall, the genetic algorithm successfully converged to a near-optimal maintenance pol-

icy under the modeled assumptions. The optimized policy balanced preventive maintenance costs against corrective maintenance costs. Different degradation mechanisms resulted in different optimal maintenance thresholds. The optimization framework successfully handled multiple degradation models simultaneously. As a result, it was concluded that the methodology identifies near-optimal maintenance policies that balance preventive and corrective maintenance costs under the modeled assumptions.

While the overall results of this thesis were promising, it is important to note some key limitations of the work. An important limitation was that the case studies considered fixed maintenance costs. Typically, the cost of maintenance could vary based on how many hours were worked and how many spare parts were required to make repairs. Incorporating dynamic maintenance costs would provide a more realistic representation of industry practice. Another limitation is the assumption of consistent environmental and operating conditions. Environmental and operating conditions were assumed to remain consistent throughout the simulation. However, seasonal temperature changes or weather patterns could impact the degradation of the components. Another limitation is that maintenance resources in the case study were always available. In practice, there could be manufacturing shortages or logistical delays that result in repairs or replacements having to be delayed. This would cause an aircraft to be grounded and decrease system availability. Another limitation is that while this thesis represents structural dependence through the series-parallel k -out-of- n architecture, it does not model stochastic degradation dependence. The assumption is that there are no common-cause failures, correlated degradation, load-sharing effects, or interactions among components that were modeled. In practice, the various components in a multi-component system do interact and could have common causes of failure.

In the future, there are several ways in which the work of this thesis could be expanded. For instance, the aircraft-inspired case study could be expanded to include larger architectures, additional subsystems, and more detailed k -out-of- n redundancy structures. In addition, the maintenance policies could be improved by considering resource constraints,

fleet-level optimization, maintenance crew availability, and spare-part inventories. In addition, this thesis considers the primary failure mechanism for each component when selecting degradation models. In practice, each component could experience multiple competing or interacting failure mechanisms. Future work could consider a framework that represents multiple failure processes for each component. As discussed, this thesis did not consider stochastic dependence and failure interactions among components and subsystems. Future work should consider how these components interact and how the degradation or failure of one component could accelerate degradation or contribute to the failure of another. Lastly, it would improve the research if real airline maintenance data were collected to obtain more accurate parameters. While components can be similar, their parameters will have variations so future work could assign distributions to uncertain parameters and propagate that uncertainty through the simulation and optimization.

This research contributes a simulation-based framework that integrates heterogeneous stochastic degradation models, imperfect maintenance, system redundancy, Monte Carlo simulation, and genetic algorithm optimization. Although demonstrated using an aircraft-inspired case study, the proposed methodology is sufficiently general to be extended to other engineering systems that experience stochastic degradation and imperfect maintenance.

Bibliography

- [1] M Amata. Application of used oil analysis for predictive maintenance of diesel engines in power and energy systems. *Global Academic and Scientific Journal of Multidisciplinary Studies (GASJMS)*, 3(12):14–27, 2025.
- [2] Carl S Byington, Matthew Watson, Doug Edwards, and Paul Stoelting. A model-based approach to prognostics and health management for flight control actuators. In *2004 IEEE aerospace conference proceedings (IEEE Cat. No. 04TH8720)*, volume 6, pages 3551–3562. IEEE, 2004.
- [3] Nuria C Caballé and IT Castro. Assessment of the maintenance cost and analysis of availability measures in a finite life cycle for a system subject to competing failures. *OR Spectrum*, 41(1):255–290, 2019.
- [4] Michele Compare, Fabio Martini, and Enrico Zio. Genetic algorithms for condition-based maintenance optimization under uncertainty. *European journal of operational research*, 244(2):611–623, 2015.
- [5] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.
- [6] Marco Fioriti, Valeria Vercella, and Nicole Viola. Cost-estimating model for aircraft maintenance. *Journal of Aircraft*, 55(4):1564–1575, 2018.

- [7] Venanzio Giannella, Michele Perrella, and Valery N Shlyannikov. Fatigue crack growth in a compressor stage of a turbofan engine by fem-dbem approach. *Procedia Structural Integrity*, 12:404–415, 2018.
- [8] Michael L Hoffman. Condition-based maintenance policy optimization using genetic algorithms and gaussian markov improvement algorithm. Master’s thesis, The Pennsylvania State University, University Park, PA, November 2018.
- [9] Yu-Hsin Hung, Hong-Ying Shen, and Chia-Yen Lee. Deep reinforcement learning-based preventive maintenance for repairable machines with deterioration in a flow line system. *Annals of Operations Research*, 2024. doi: 10.1007/s10479-024-06207-x.
- [10] Mirosław Kowalski, Mariusz Izdebski, Jolanta Żak, Paweł Gołda, and Jerzy Manerowski. Planning and management of aircraft maintenance using a genetic algorithm. *Eksploatacja i Niezawodność*, 23(1):143–153, 2021.
- [11] Juseong Lee and Mihaela Mitici. Deep reinforcement learning for predictive aircraft maintenance using probabilistic remaining-useful-life prognostics. *Reliability Engineering & System Safety*, 230:108908, 2023.
- [12] Xiaoping Liu, Dejun Cui, Lihang Wang, and Lijie Zhang. *Reliability Evaluation of Hydraulic Pump Based on Performance Degradation*. Universitätsbibliothek der RWTH Aachen, 2018.
- [13] Yaqiong Lv, Pan Zheng, Jiabei Yuan, and Xiaohua Cao. A predictive maintenance strategy for multi-component systems based on components’ remaining useful life prediction. *Mathematics*, 11(18):3884, 2023.
- [14] Marzio Marseguerra, Enrico Zio, and Luca Podofillini. Condition-based maintenance optimization by means of genetic algorithms and monte carlo simulation. *Reliability Engineering & System Safety*, 77(2):151–165, 2002.

- [15] Renato Giorgiani Nascimento and Felipe A. C. Viana. Cumulative damage modeling with recurrent neural networks. *AIAA Journal*, 58(12):5459–5471, 2020. doi: 10.2514/1.J059250.
- [16] Onyedikachi Chioma Okoro, Maksym Zaliskyi, Serhii Dmytriiev, Oleksandr Solomentsev, and Oksana Sribna. Optimization of maintenance task interval of aircraft systems. *International Journal of Computer Network and Information Security*, 12(2):77, 2022.
- [17] Zarina Poberezhna and Maksym Zaliskyi. A method of cost-effective operation of equipment in the aviation enterprise with using intelligent technologies. *Optimization*, 17:18, 2024.
- [18] Hani Shahmoradi-Moghadam, Nima Safaei, and Seyed Jafar Sadjadi. Robust maintenance scheduling of aircraft fleet: A hybrid simulation-optimization approach. *IEEE Access*, 9:17854–17865, 2021.
- [19] Morteza Soltani. Joint optimization of opportunistic predictive maintenance and multi-location spare part inventories for a deteriorating system considering imperfect actions. *arXiv preprint arXiv:1810.06315*, 2018.
- [20] Yuo-Tern Tsai, Kuo-Shong Wang, and Hwei-Yuan Teng. Optimizing preventive maintenance for mechanical components using genetic algorithms. *Reliability engineering & system safety*, 74(1):89–97, 2001.
- [21] Yao Wang, Xinqin Gao, Yuanfeng Cai, Mingshun Yang, Shujuan Li, and Yan Li. Reliability evaluation for aviation electric power system in consideration of uncertainty. *Energies*, 13(5):1175, 2020.
- [22] Jun Xu, Zhenglin Liang, Yan-Fu Li, and Kaibo Wang. Generalized condition-based maintenance optimization for multi-component systems considering stochastic dependency and imperfect maintenance. *Reliability Engineering & System Safety*, 211:107592, 2021.

- [23] Zhixian Yang and Guobin Yang. Optimization of aircraft maintenance plan based on genetic algorithm. *Physics Procedia*, 33:580–586, 2012.
- [24] Enrico Zio and Giovanni Piloni. Particle filtering prognostic estimation of the remaining useful life of nonlinear components. *Reliability Engineering & System Safety*, 96(3):403–409, 2011.