A Stochastic Event-Generation Simulator for Dynamic Risk Assessment and Maintenance Decision Support in Aircraft Systems

Adjalla Marcel Morel, Magassa Oumar, Cantez Ali, Farez Yazid, Silva de Souza Aguiar Fredson, Kabkoub Keltoum, Diallo Amadou
Institut Galilée, Université Sorbonne Paris Nord, Villetaneuse, 93430, France

Jérôme Lacaille
Safran Aircraft Engines, Moissy-Cramayel, 77550, France
Jerome.lacaille@safrangroup.com

Abstract

This paper presents a stochastic event-generation simulator developed as a foundational component of a project, which aims to dynamically update risk assessments using operational data. The simulator models the degradation, inspection, and failure dynamics of a system equipped with a single component and a single failure mode, providing a controlled environment for studying risk evolution and maintenance decision-making.

Component ageing follows a Weibull lifetime distribution, inspections occur according to a stochastic calendar, and measurement uncertainty is introduced through a Beta-distributed inspection signal. A threshold-based detector triggers preventive or corrective maintenance actions, generating a detailed chronological log of inspections, failures, replacements, false alarms, missed detections, and associated costs. Monte-Carlo experimentation enables the analysis of how inspection intervals, detector accuracy, and ageing parameters jointly influence operational risk and maintenance efficiency.

This simulator is a first step and already provides a ground-truth environment for validating future risk-estimation algorithms, since the true underlying failure rates are known. Second, once calibrated on real operational data, it can support fleet-level event analysis, including residual-risk estimation required when new failure modes emerge. Finally, the detector model contributes to the dynamic update of occurrence ratings, improving the reliability for future component design.

The contribution of this paper is threefold:

1. We introduce an analytically-controlled stochastic simulator specifically designed for validating dynamic risk-estimation algorithms.
2. We propose a simple but realistic fatigue–inspection–detection pipeline capturing ageing, imperfect measurements and maintenance actions.
3. We provide a fully open, reproducible implementation tailored for education and early-stage research.

1. Introduction

Risk assessment in the aerospace industry traditionally relies on structured methodologies such as Failure Mode, Effects & Criticality Analysis (FMECA). These analyses are essential for certification, safety justification, and maintenance planning. However, once established, they tend to remain static, even though operational data continuously accumulate throughout the life of an aircraft engine or landing gear system. As a result, risk evaluations may become outdated, and engineering teams must rely on time-consuming manual analyses to update occurrence ratings or justify maintenance decisions.
The project "Algorithm for Collection and Control of Operational Risks for Decision-making" (ACCORD) was initiated to address this gap. Its objective is to develop a dynamic risk-estimation framework capable of updating FMECA occurrence ratings using operational evidence.

The project combines two major components:

1. **A risk-estimation algorithm** capable of integrating historical observations, prediction errors, and contextual information.
2. **A stochastic event-generation simulator** capable of producing realistic operational scenarios for validation, calibration, and decision support.

This paper focuses on the second component: a simplified but technically rigorous simulator developed initially as a pedagogical project for Master's students at Institut Galilée. Despite its simplicity—one system, one component, one failure mode—the simulator captures essential mechanisms of degradation, inspection, detection, and maintenance. It generates detailed event logs suitable for Monte-Carlo analysis, maintenance-policy evaluation, and validation of risk-estimation algorithms.

Beyond its initial pedagogical purpose, the simulator constitutes a strategic asset for ACCORD. It provides a controlled environment where the true failure behaviour is known, enabling quantitative validation of dynamic risk-estimation methods. Once calibrated on real operational data, it can also support engineering teams and aviation authorities by estimating residual risk when new failure modes emerge.

2. Background and Motivation

Traditional FMECA processes rely on expert judgement and historical knowledge. Their occurrence ratings are rarely updated dynamically, even though operational data could provide evidence of changing risk levels. Recent research has explored automated FMECA updates, similarity-based event classification, and PHM-based risk indicators. However, validating such approaches requires datasets where the true failure rates and degradation mechanisms are known.

Digital twins and fleet simulators have been proposed in the literature, but they are often complex, proprietary, or tailored to specific systems. The objective here is different: to build a transparent, modular, and analytically controllable simulator that can serve as a testbed for risk-estimation algorithms and maintenance-policy evaluation.

The simulator described in this paper is intentionally simple, yet it incorporates realistic elements: stochastic ageing, imperfect inspections, preventive and corrective maintenance, and cost modelling. This balance between simplicity and realism makes it suitable for both academic training and early-stage research within ACCORD. Beyond simply needing data whose characteristics are fully understood, it is also essential, within the framework of collaborative work between a company like Safran and a university, to have non-confidential data that can be exchanged between partners. This type of data—operational events—is indeed highly sensitive. Such a simulator enables the establishment of robust, publication-oriented collaborations between industrial and academic partners.

3. Simulator Architecture

3.1 Implementation and experimental setup

To support experimentation and reproducibility, the simulator is deployed as an online application developed in Python using the Streamlit framework. It enables the configuration of simulation parameters, the execution of Monte Carlo experiments, and the generation of event logs.
For the illustrative scenario used throughout this paper, the following configuration is selected:

- **Component ageing**
  Mean time to failure $\eta = 720$ h (30 days)
  Shape parameter $\beta = 3$ (wear-out behaviour)
- **Inspection schedule**
  Nominal interval $\mu = 168$ h (7 days)
  Standard deviation $\sigma = 25.2$ h (± 1 day variability)
- **Detection model**
  Threshold $s = 0.5$
  Maximum standard deviation (low-quality detector) corresponding to κ producing precision ≈ 0.8 and recall ≈ 0.3
  Noise decreasing as wear increases (Beta-distributed observation)
- **Maintenance policy**
  Systematic removal at age limit 792 h (33 days)
  Replacement delay: 12 h
  Cost model applied to new component (100€), inspections (100€), preventive replacements (1000€), corrective replacements (1200€).

These parameters define the baseline configuration used for all subsequent experiments and analyses.

The full source code and documentation are available in an open GitHub repository:
https://github.com/alc4ml/ProjetM2D2526/
The simulator can be accessed online without installation via Streamlit at:
https://projetm2d2526-m2d.streamlit.app/

This platform enables researchers, students, and industrial partners to quickly explore different ageing behaviours, inspection strategies, detector qualities, and fleet-level dynamics, while ensuring that the exact scenarios used in this paper can be reproduced.

3.2 System and Component Model

We consider a single system composed of one component subject to progressive degradation. The component exhibits a single failure mode whose time-to-failure follows a Weibull distribution, a common choice for modelling wear-out mechanisms in mechanical systems. At any time t, the component is characterised by:

- its chronological age,
- its underlying degradation level,
- its instantaneous probability of failure before the next scheduled inspection.

The cumulative failure probability is given by the Weibull distribution:

$$F(t) = \mathbb{P}(T \leq t) = 1 - \exp\left(-\left(\frac{t}{\eta}\right)^{\beta}\right)$$

With η the scale parameter and β the shape parameter.

For the illustrative scenario used in this paper, the parameters are set to $\eta = 720$h and $\beta = 3$, representing a classical increasing-hazard wear-out behaviour ($\beta > 1$). These values define the reference configuration used in this study.

The Weibull model allows the generation of right-censored observations (preventive replacements) and true failures, both of which are later exploited in Section 6 for parameter re-identification.

3.3 Inspection Scheduling

Inspections are not strictly periodic. Instead, inspection dates are generated by sampling stochastic intervals around a nominal value. This models operational constraints such as maintenance availability, workload, or logistical variability.

$$I_k = \sum_{j=1}^{k} \Delta I_j \quad \text{avec} \quad \Delta I \sim \mathcal{N}(\mu, \sigma^2)$$

With μ=168h and σ=25.2h in the illustrative scenario (corresponding to a typical weekly inspection ± one day). After each inspection event, the next inspection date is scheduled by adding a newly sampled interval.
At each inspection, the simulator records the timestamp, the component age, the noisy degradation estimate, and the resulting maintenance action.

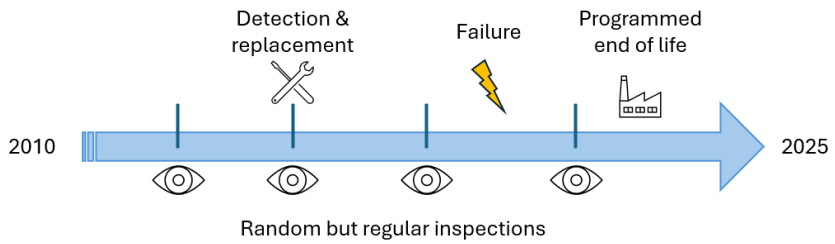

Figure 1. Schematic view of regular inspections and events.

3.4 Imperfect Detection

Real-world inspections are imperfect. To represent this realistically, the simulator generates a noisy observation of the true degradation level.

Let $p=F(t)$ denote the latent wear state, equal to the true cumulative probability of failure at age t. The inspection produces an observed degradation score:

$$W \mid p \sim \mathrm{Beta}(a, b)$$

where the parameters are defined such that $\mathrm{E}[W] = a/(a+b) = p$, $\kappa = a+b$, thus

$$a = \kappa p, \qquad b = \kappa(1-p)$$

The variance is then expressed as:

$$\mathrm{Var}(W) = \frac{p(1-p)}{\kappa + 1}$$

The parameter κ controls the quality of the sensor:

- If $\kappa \gg 1$: the variance is low, the inspection is very accurate.
- If κ is small: the variance is high, the inspection is very noisy.

The resulting distribution allows the modelling of both false alarms (false positives) and missed detections (false negatives). The same formulation also allows inferring detector quality from data using moment matching.
A graphical representation of the Beta density for different wear levels is provided in Figure 2.

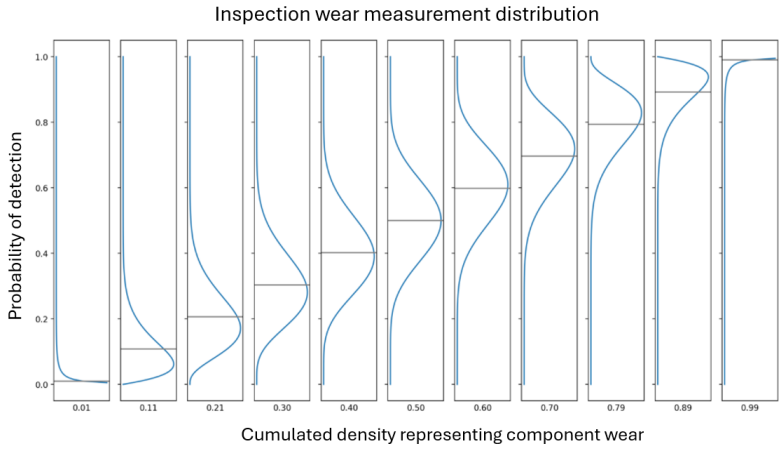

Figure 2. This graph presents the density of the detection score *W* according to the cumulative density *p* of the component wear.

In the forward direction of the simulator, the wear level p is assumed known and fix the quality κ, calculate (a, b), and then analytically generate the measurement data W.

However, it is also possible to infer the detector's quality from the observations. We estimate the mean $\bar{W}$ and the variance σ^2, then the confidence parameter is obtained using the method of moments:

$$\hat{\kappa} = \frac{\bar{W}(1-\bar{W})}{s^2} - 1$$

Then we deduce the parameters of the distribution: $\hat{a} = \hat{\kappa}\,\bar{W}$ and $\hat{b} = \hat{\kappa}\,(1-\bar{W})$.

3.5 Maintenance Decision Logic and cost

A simple threshold-based rule is used:

- if the observed (generated) degradation score W exceeds a predefined threshold s, a preventive replacement is performed,
- otherwise, the component remains in service.

If a failure occurs before the next inspection, a corrective replacement is triggered immediately (see Figure 1).

The simulator also include:

- age-limit replacements,
- delays between failure and intervention,
- and a cost accumulation for each event type.

3.6 Event Log Generation

The simulator outputs a detailed chronological logbook including system and component identifiers, event type (inspection, failure, replacement), event date and time, component age, usage since last event, maintenance decision correctness (true/false), event cost and cumulative cost.

	Unnamed: 0	system_id	component_id	event_date	event_time	event_type	event_report	system_age	component_age	usage_since_last_event_h	FF	cost_event	cost_cumulated
0	0	1	1	2010-01-07	16:32:56	inspection	None	160.55	160.55	160.55		100	100
1	1	1	1	2010-01-13	18:41:37	inspection	None	306.69	306.69	146.14		100	200
2	2	1	1	2010-01-21	22:01:04	inspection	None	502.02	502.02	195.32		100	300
3	3	1	1	2010-01-28	23:18:45	inspection	wornout	671.31	671.31	169.29		100	400
4	4	1	1	2010-01-29	00:01:00	replacement	unused	672.02	672.02	0.7		1100	1500
5	5	1	2	2010-02-04	07:07:31	inspection	None	823.13	151.11	151.11		100	1600
6	6	1	2	2010-02-11	17:07:17	inspection	None	1001.12	329.1	178		100	1700
7	7	1	2	2010-02-18	04:25:09	failure	None	1156.42	484.4	155.3		1200	2900
8	8	1	2	2010-02-18	04:25:09	replacement	unused	1156.42	484.4	0		1100	4000
9	9	1	3	2010-02-25	20:06:27	inspection	None	1340.11	183.69	183.69		100	4100

Figure 3. Extract from the logbook of the illustrative scenario.

This logbook is the core output used for analysis, algorithm validation, and scenario exploration. The fleet of 250 systems each contains a single component of identical type and characteristics. During inspections, some components are replaced; however, the removed component may still be in acceptable condition. After verification, maintenance personnel may confirm the absence of wear, in which case the component is returned to stock for future reuse. Consequently, two types of replacements coexist: the installation of a brand-new component (unused) or the reinstallation of an old but still serviceable (reused) component.

4. Simulation Scenarios

The simulator produces a timeline of events for each system, considering component aging, scheduled inspections, and real failures. The main scenarios are the following.

- **Inspection without Replacement:** An inspection is carried out at the planned date. If the estimated failure risk remains below the defined threshold, no

replacement is performed. The component continues to operate normally, and only the inspection cost is recorded.

- **Inspection Leading to Preventive Replacement:** During an inspection, the failure risk exceeds the allowed threshold. A preventive replacement of the component is then triggered. After this replacement, a deep inspection is carried out to evaluate whether the decision was justified or not (the simulation simply evaluates the survival function of the component). This analysis makes it possible to identify correct replacements as well as decision errors.
- **Component Reuse and Maintenance:** An inspection may imply a later replacement, and after replacement, a deeper inspection is carried out. If the removed component was identified as still in good condition, it is saved for later reuse. In the same process, the component is restored to a like-new condition. The cost of reusing a component is lower than the cost of acquiring a new one.
- **Real Failure:** A failure occurs before the next scheduled replacement. The system suffers a breakdown, and immediate replacement of the component is required. The failure cost is recorded, and a new component is installed to restore the system operation.
- **Inspection Scheduling:** After each inspection, a new inspection date is automatically scheduled. This date depends on the chosen maintenance strategy (for example, frequent or spaced inspections). The time interval between two inspections has a direct impact on the failure risk: the more spaced the inspections are, the higher the probability that a failure occurs before the next one.
- **Automatic Replacement at Age Limit:** A component is automatically replaced when it reaches a predefined maximum age, independently of inspections.
- **Delay Before Replacement:** In some situations, a component cannot be replaced immediately after a failure or a maintenance decision. A random delay is then considered before the intervention, potentially leading to additional costs.
- **Fleet Evolution Over Time:** The number of systems also evolves over time, following a random law that includes the creation of new systems, but also the removal of systems.

These scenarios allow the analysis of the interactions between inspection frequency, detection quality, ageing behaviour, and maintenance cost.

In practice, we model fleet evolution using an approximately stochastic logistic growth process, an assumption inspired by real-world fleet data such as those from Emirates and Air France. However, since the fleet size evolves in discrete units (a count of systems), a more appropriate formalism is a continuous-time Birth–Death process. In this framework, time does not advance in fixed steps; instead, each transition occurs at a randomly sampled instant, following the dynamics of a Continuous-Time Markov Chain. For our model, we consider the following birth rate to follow a logistic model, given the number n of systems:

$$\lambda_n = r\, n\, (1 - n/K)$$

where r is a growing factor, and K is the support capacity; the death rate $\nu_n = n * \nu$ is assumed to be constant per system and describes the average rate at which a system is

retired. We also impose that the population does not die, so that at least one system is always maintained.
In practice, the Markov chain calculates the dates of the next event following the exponential law Exp(λ_n+ν_n) and decides whether to add a system with the probability $\lambda_n/(\lambda_n+\nu_n)$ or to remove one with the opposite probability $\nu_n/(\lambda_n+\nu_n)$.

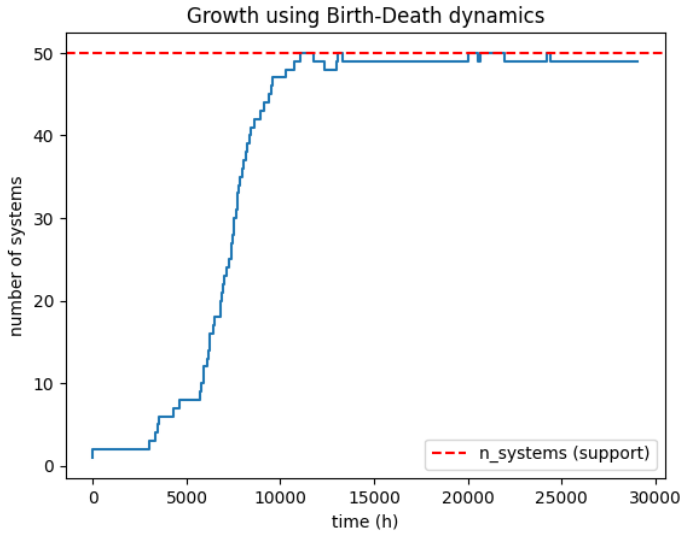

Figure 4. Example of initial fleet size evolution, limited to *K*=50 systems.

5. Analysis Capabilities

A Key Performance Indicator (KPI) is a quantitative measure used to evaluate the effectiveness and success of an organization, a system, or a strategy in relation to predefined objectives. In the context of industrial maintenance, KPIs help to manage equipment reliability, optimize operational costs, and measure the impact of preventive strategies. KPIs are quantifiable measures of performance over time for a specific objective. KPIs provide measurable targets for teams, allow progress to be tracked over time, and generate insights that support better decision-making across the organization.

In this analysis, three main categories of KPIs are considered: the number of failures and replacements, the quality of the detector and the costs.

5.1 Number of failures and replacements

Several indicators were calculated to evaluate system reliability across the entire fleet. At the fleet level, the number of failures and replacements was calculated over three different time periods: monthly, quarterly and yearly. The objective is to determine how frequently systems fail and how often they require repair or replacement.
First, we aim to measure the average number of failures across the entire fleet. To do this, the output data is grouped by period. The total number of failures is then calculated for each period, and the average of these values is subsequently determined. For example, to obtain the average number of failures per month, the average of the number of failures observed across all months is calculated. The same principle is applied to replacements.
The standard deviation is also calculated for these indicators. It makes it possible to measure the stability of the fleet's maintenance activity. For example, in the case of

failures: low standard deviation indicates that the number of failures is relatively constant from one period to another, on the contrary high standard deviation indicates a high variability in the number of failures.
Another objective is to determine how many times a system fails on average, either over the entire lifetime of the fleet or per period. The average of these values over the entire fleet is then calculated in order to obtain the average number of failures per system and per period.
Finally, we compute the **PER**, Preventive Effectiveness Ratio, which is used to evaluate the balance of the maintenance strategy:

- $PER > 1$ – More preventive replacements are performed than actual failures occur. If the ratio is very high, components may be replaced too early (over-maintenance).
- $PER < 1$ – There are more failures than replacements. This indicates that preventive maintenance is not frequent enough or not effective enough to prevent failures (under-maintenance).
- $PER \approx 1$ – There is a numerical balance between planned interventions and observed failures.

5.2 Quality of the detector

The quality of the detection system is evaluated by analysing its ability to correctly identify failures. A detection is counted when an inspection leads to a valid replacement, meaning that the alert corresponds to a real failure. Two types of events are considered:

- Detection: corresponds to the number of times the system correctly identified an event (true positives).
- False alerts: corresponds to the number of times the system triggered an alert when no real failure was present (false positives).

Usual precision and recall indicators are also computed.

- Precision answers the question: *"When the system triggers an alert, what is the probability that there is actually a failure?"*. It is equal to 1 minus the false alarm rate.
  A precision of 1 (100%) means that there are no false alerts. Every triggered intervention is useful.
- Recall answers the question: "Out of all the failures that occurred, what percentage was successfully detected by the system?", also usually called the detection rate.
  For example, if Recall = 0.5, it means that only one failure out of two is detected by the system.

5.3 Costs

The last category of KPIs concerns the analysis of the average fleet's maintenance costs. As for failure and replacement rates we also calculate the average fleet cost for different periods.

6. Statistical validation

To validate the mathematical consistency of the simulator, we performed parameter estimation on the generated output logs. The goal is to reverse-engineer the logs to find the original input parameters using Maximum Likelihood Estimation.

6.1 Inspection Intervals (Normal Distribution)

The time between inspections is modelled using a normal distribution. By filtering the logs for consecutive inspections and calculating the mean and standard deviation of the time deltas, we successfully recovered:

- Calculated μ: 166.54 hours and 166.13 hours (True input: 168)
- Calculated σ: 26.70 hours and 27.24 hours (True input: 25.2)

6.2 Failure Ages (Weibull Distribution and Right-Censoring)

The component's natural failure age follows a Weibull distribution characterized by a shape parameter β and a scale parameter η.
If we only calculate the parameters based on the components that actually failed, the estimation yields incorrect results (e.g., η=*575* instead of 720). This happens because of right-censoring: the inspections successfully remove worn-out components *before* they fail. By ignoring these "surviving" components, the naive model assumes components die much younger than they would. To correctly estimate the parameters, the likelihood function L must account for both failures (using the Probability Density Function, $f(t)$) and preventive replacements (using the Survival Function, $S(t)$):

$$L(\eta, \beta) = \prod_{\text{failures}} f(t_i; \eta, \beta) \times \prod_{\text{replaced}} S(t_j; \eta, \beta)$$

Using a custom Python script optimizing this negative log-likelihood, we successfully recovered the exact parameters:

- Calculated η: 720.16 hours and 721.06 (True input: 720)
- Calculated β**:** 3.00 and 2.96 (True input: 3.0)

6.3 Influence of parameters on the simulation

Here are some results of the computation on one simulation. We run multiple simulation iterations to obtain statistical results. Our goal was to analyse the influence of some parameters

As shown on the next graph (Figure 5) PER appears to be well calibrated for a detection threshold value of 0.5. It balances the inspection interval according to the maintenance policy. The detection threshold represents the sensitivity of the diagnostic system used during inspections. When this threshold increases, several trends emerge:

- the preventive maintenance efficiency ratio (PMR) decreases;
- detector accuracy drops sharply;
- the number of failures tends to increase;
- recall decreases.

These results are explained by the fact that a high threshold delays the triggering of preventive maintenance actions. Components are then left in a degraded state for longer before intervention, which increases the risk of failure.

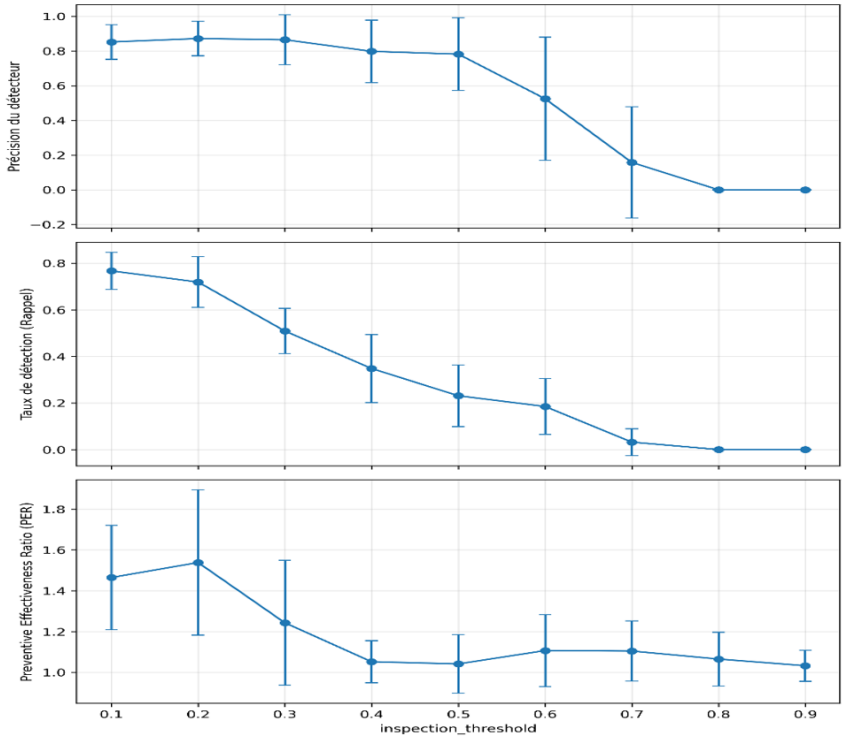

Figure 5. Evolution of Precision, Recall and PER versus detection threshold *s*.

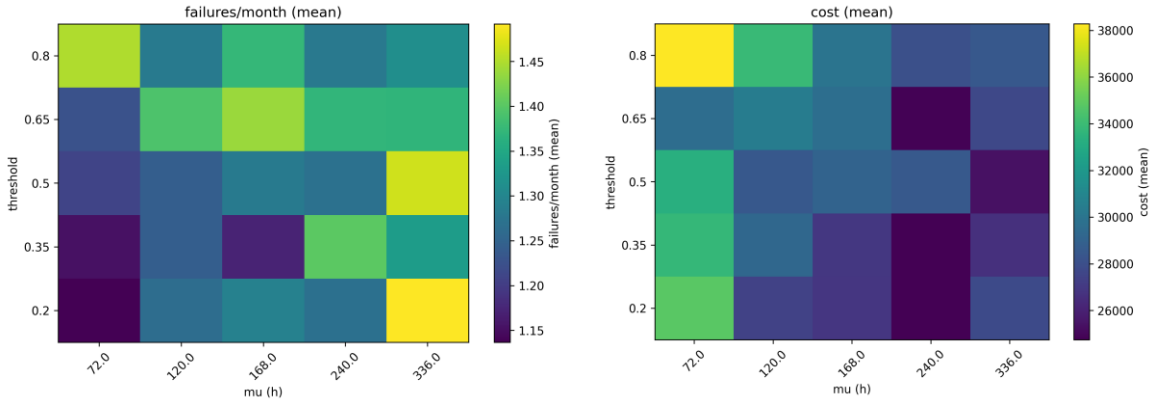

Figure 6. Average number of failures per month (left) and cost per system (right) according to inspection frequency (mu) and detection threshold.

The lowest costs are observed for intermediate threshold values and relatively high inspection intervals.

7. Conclusion and Future Work

The event-generation simulator presented in this paper provides a controlled and analytically traceable environment for studying degradation, inspection, and maintenance dynamics. Despite its intentionally simple scope—one system, one component, one failure mode—it reproduces key mechanisms required to analyse inspection efficiency, detection accuracy, and the trade-off between preventive and corrective actions. The statistical validation confirms that the simulator behaves

consistently with its mathematical foundations, enabling reliable use for algorithm testing and methodological exploration.

Beyond its pedagogical origins, the simulator constitutes a practical asset for the ACCORD project. It enables the generation of fully understood and non-confidential operational-event datasets, thereby supporting collaboration between academic teams and industrial partners. The simulator also provides a baseline environment for validating dynamic risk-estimation methods and for assessing the impact of maintenance policies or detector characteristics.

Future developments will extend the framework toward multi-component systems, interacting failure modes, and richer fleet-level behaviours. Integration with ACCORD's risk-estimation algorithms and deployment as a demonstrator for engineering teams will further enhance its value as both a research platform and a decision-support tool.

Acknowledgements

We thank the University of Sorbonne Paris Nord (USPN) and Institut Galilée for making this work possible. We would also like to express our gratitude to Polytechnique Montréal, and especially to Professor Samuel Jean Bassetto, who leads the ACCORD program. This collaboration between Safran and Polytechnique Montréal played a key role in the development and support of this research.

References and footnotes

1. Department of Defense. (1980, 24). *Procedures for performing a Failure Mode, Effects, and Criticality Analysis, MILSTD 1629-A*. MILSTD 1629A.
2. Hakim Elbadiry, Samuel Bassetto, & Mohammed Salah Ouali. (2016). Comparative study of similarity analysis methods for Aviation system failure. *IEEE Aerospace and Electronics systems magazine*. https://doi.org/DOI. No. 10.1109/TAES.2016.150109
3. Mili, A., Bassetto, S., Siadat, A., & Tollenaere, M. (2009). Dynamic risk management unveil productivity improvements. Journal of Loss Prevention in the Process Industries, 22(1), 25-34. https://doi.org/10.1016/j.jlp.2008.07.011
4. Dersin, Pierre (2023). Modeling Remaining Useful Life Dynamics in Reliability Engineering, CRC-Press, ISBN 1032168595.
5. Taha, H. A., Sakr, A. H., & Yacout, S. (2019). Aircraft engine remaining useful life prediction framework for industry 4.0. Proceedings of the International Conference on Industrial Engineering and Operations Management, 1093–1103.
6. Elsheikh, A., Yacout, S., Ouali, M. S., & Shaban, Y. (2020). Failure time prediction using adaptive logical analysis of survival curves and multiple machining signals. Journal of Intelligent Manufacturing, 31(2), 403–415. https://doi.org/10.1007/s10845-018-1453-4